Free Surface Water Flows –

Modeling, Simulation and Control

Adolf Hermann Glattfelder
Prof. emer. Dr. sc. techn.

Automatic Control Lab,
ETH Zurich[1]

August 21, 2017

[1] Report (open source), download from https://www.research-collection.ethz.ch/handle/20.500.11850/175527

Abstract

Upstream and downstream channels are the key subsystems for the design of level/flow control loops in run-of-river hydropower plants. Their dynamic response is notoriously complex, exhibiting low frequency resonance with weak damping, travelling waves, both sub- and supercritical flow, hydraulic jumps, etc.

Part one of this report provides mathematical models derived from the basic physical conservation laws of mass and momentum (Reynolds averaged Navier Stokes partial differential equations ('pde')) for water flow and depth, using the 'method of lines'/'compartment' approach (leading to models in discrete space, 'continuous' time), first for *long channels* with constant or weakly changing cross section, then for *wide channels* with lateral extensions and finally for *short channels* such as spillways etc. where change of surface level is no longer small compared to water depth.
All models are implemented in established industrial simulation packages, suitable for the engineering phases of preliminary control system design up to factory acceptance testing. Simulations are documented in detail and results for typical transients are discussed.

Part two of the report develops the design of model-based level control for the linearized pde-model, basically using a notion from discrete control systems which leads to the impedance matching concept. Then this design approach is applied to a linearized state space model with floating coefficients, and finally to the nonlinear time domain models of Part one with large but slow dynamic variation of flow regime and including 'controller gain scheduling'.
Finally the model-based design method is verified on two typical application cases.

This report is meant mainly for control engineers charged with design projects for control of hydrological and hydropower systems. The math needed is kept to the master/diploma level of university engineering graduates.

To facilitate the access to small enterprises an 'open source' system (see www.scilab.org) has been used. For further use the source code(s) for all cases are given in the connected 'zip-container' [1].

And all models are given as well in Matlab/Simulink source (using release 'R2016b') in a second connected 'zip-container'[2].

keywords:

- Partial Differential Equations for open channel flow
  (Navier Stokes, Shallow Water equations, de Saint Venant)
- Integration by Method of Lines,
- Solution by Laplace Transformation,
- Model based design of Boundary level control ,
- Model based Gain-Scheduling,
- Scilab/Xcos v.5.4.1, and Matlab/Simulink R2016b.

[1] LevelDynamics_Scilab/Xcos_v541_sources.7z (Research Data Model) at
https://www.research-collection.ethz.ch/handle/20.500.11850/175543
[2] LevelDynamics_Matlab/Simulink_R2016b_sources.7z (Research Data Model) at
https://www.research-collection.ethz.ch/handle/20.500.11850/175549

Contents

Chapter 1

Introduction

The subject of this report is the dynamic response of level and flow in typical free surface channels, as they are present in typical low-head hydropower plants, or similar natural river beds.
Note that this already implies a number of practical assumptions:

'free surface' means that disturbances will propagate at Froude's velocity $v_F = \sqrt{g.D}$,
in contrast to closed pipe flow, where they propagate with the speed of sound $v_S = \sqrt{\rho.\varkappa}$.[1]

The 'channel' cross section is to be predominantly rectangular
and not strongly trapezoidal or even triangular,

and 'typical' flow is to be turbulent, and not laminar, and without large-scale vortices.
In other words flow velocity is essentially the same over the cross section (resulting in 'piston' flow).

1.1 Basic equations

Fluid flow systems are basically described by partial differential equations (PDE), with independent variables x, y, z, for longitudinal, transversal and vertical space coordinates and time t and dependent variables level H and flow velocities u, v, w in directions x, y, z.
One approach ([1], and references therein) is to use are the 3-D Reynolds Averaged Navier-Stokes Equations (NSE):

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + ... \\
\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (vv)}{\partial y} + \frac{\partial (vw)}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + ... \\
\frac{\partial w}{\partial t} + \frac{\partial (uw)}{\partial x} + \frac{\partial (wv)}{\partial y} + \frac{\partial (ww)}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + ... - g \\
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} &= 0
\end{aligned}$$

where the first to third equations are the momentum balances, and the fourth equation is the volume balance, where p is the local pressure, fluid density is ρ and ... indicate the eddy viscosity terms.

There is an additional kinematic condition for the free fluid surface $\eta(x, y, t)$ with η measured from the undisturbed water elevation and u^s, v^s, w^s velocity components at the surface

$$\frac{\partial \eta}{\partial t} + u^s \frac{\partial \eta}{\partial x} + v^s \frac{\partial \eta}{\partial y} = w^s$$

[1] g gravity constant, D local water depth, ρ fluid density, $\varkappa$ fluid elasticity plus pipe wall elasticity

and similarly for the bottom contour, where h is the water depth from the undisturbed water level and u^b, v^b, w^b are the velocity components at the bottom

$$u^b \frac{\partial h}{\partial x} + v^b \frac{\partial h}{\partial y} + w^b = 0.$$

An alternative which is often used for such channel geometries are the equations of de Saint Venant (SVE). They can be obtained from the above equations by integration over y, whereby the channel width B comes in, and z from the bottom $z = -h$ to the free surface $z = +\eta$, that is the total local water depth D, and with A as the cross section surface $A = B \cdot D$, and U as the mean velocity for this cross section [1]

$$\begin{aligned} \frac{\partial(A \cdot U)}{\partial t} + \frac{\partial(A \cdot U \cdot U + g \cdot I_1)}{\partial x} &= g \cdot A \cdot (S_b - S_f) + g \cdot I_2 + ... \\ \frac{\partial A}{\partial t} + \frac{\partial(A \cdot U)}{\partial x} &= 0 \end{aligned}$$

where $S_b := -\partial h / \partial x$ is the bottom slope, S_f is the friction slope, I_1 is linked to the normalized pressure force over the cross section

$$I_1(x, H) = \int_{-h}^{\eta} (H - z) \cdot B(z) dz$$

and I_2 represents the integral of a reaction force from hydrostatic pressure acting on the boundary, such that

$$\frac{\partial}{\partial x}(g \cdot I_1) = g \cdot A(x) \cdot \frac{\partial H}{\partial x} + (g \cdot I_2)$$

This is a set of two first order PDE's for conservation of momentum and volume for independent variables x (longitudinal, in direction of mean flow) and time t. Dependent variables are water depth H and cross section mean longitudinal velocity U.

A third approach ('shallow water equations') uses the same independent variables, both for the general three-dimensional spacial case, and also their two- and one-dimensional reductions, but replaces h, η by water depth D and velocities u, v, w by corresponding volume flows Q_x, Q_y, Q_z. For instance in the SVE-form above, note that $Q(x) := A(x) \cdot U(x)$, and thus

$$\begin{aligned} \frac{\partial(B \cdot D)}{\partial t} + \frac{\partial(Q)}{\partial x} &= 0 \\ \frac{\partial Q}{\partial t} + \frac{\partial(Q^2/B \cdot D + g \cdot I_1)}{\partial x} &= g \cdot B \cdot D \cdot (S_b - S_f) + g \cdot I_2 + ... \end{aligned}$$

This is said to allow considering some cases better, where level and velocities change strongly over a short distance, such as in 'hydraulic jumps'. It is also closer to Q which most hydropower engineers are more familiar with than U. Also D is not directly observable / measurable in real installations. Therefore it is often more convenient to split D into H (water level above a given geodetic horizon) and S (bottom contour position above this geodetic horizon). Then $D(x) = H(x) - S(x)$. – This third approach shall be used further on in this report.

1.2 Solving the PDE's

For practical purposes, specific *transients* of dependent variables over time or steady state *profiles* of those dependent variables over space coordinates are needed. In other words the PDE's must be 'integrated'.

The usual way (see [1], [2], and references therein) to proceed from here for all three approaches is to move to both spacial and temporal finite differences with a fixed grid, suitable for numerical integration with off-the-shelf and well proven methods.

Instead a different approach shall be used here:

It consists of using finite intervals (typically of equal length L) spanning up 'compartments' along the space axes, and using the center values of these intervals as the dependent variables for level/depth and velocity/volumeflow. The time t is still continuous. This produces a set of coupled ordinary first order differential equations (ODE's), one for each compartment, if distinct compartments are used for dynamic conservation of each dependent variable separately, specifically in the context of this report for mass / volume and momentum.
In other words this produces a typical state space system description so well known in Automatic Control, where a huge amount of tools for analysis, design and simulation is available. Further this form links directly to the well known transfer function / frequency domain techniques. Finally this form of model integrates directly with the other elements of standard level/flow control loops. And it neatly fits into the experience, the knowhow, and the intuition of industrial control engineers.
These strong links to engineering knowhow are the main motivation for using this particular approach here.

This finite cell size approach is known in computational mathematics as the **'method of lines'**, MOL for short, [6], [5], and references therein. The key step is to replace the partial derivatives by finite differences

$$\left.\frac{\partial u}{\partial x}\right|_i \quad \rightarrow \quad \frac{u_i - u_{i-1}}{\Delta x_i} \qquad \text{(this is a simple 'upwind'-approximation)}$$

In electrical and mechanical engineering the same approach is known as the **'lumped parameter method'**, because PDE's are also known as system descriptions with 'distributed parameters'. A typical example is the well known "Pi" (Π) model of a High Voltage AC line, where each constitutive element consists of two adjacent sections of capacitance to ground and an inductance section connecting them.

And in pharmacokinetics, which describes the dynamic distribution of drugs in an organism, and more generally in process engineering this approach is also known as the **'compartment method'**. In each such compartment, ideal mixing of the drug to its fluid content is assumed such that the concentration at the center is representative for the whole compartment.

These are just different names for the same thing. They go back at least to the 1960'ies. Their emergence is strongly tied to the classical frequency response approach to control system design, to the advent of analog computers, and subsequent simulation packages for control analysis and design on digital computers.

The drawback of this method is (of course) the reduced spacial resolution. Note that improving this resolution along any one spacial axis will increase linearly the size of the ODE-system, and thus the programming complexity and the computational effort. So there will be a practical ('soft') limit for the number of such compartments.

Applying the transfer function or frequency response approach to a given ODE system, this corresponds to a given bandwidth of the model (see for instance [7] [8], and references therein), and higher frequency modes will be 'lost'. – However this opens a way of neatly balancing the model bandwidth (and thus the model size) to the closed loop bandwidth specifications given for a specific project. This requires selecting the compartment length L appropriately with respect to the overall channel length.

1.3 Typical channel layouts

Five typical situations shall be considered here for their dynamic characteristics

- a fairly small but deep storage pool;
- a typical fairly long but shallow channel with subcritical flow, and with supercritical flow due to a steeper bottom slope;
- a long channel with a lateral shoulder for absorbing high flow or with a lateral pool widening;
- a short channel, where water surface level gradients are no longer negligible.

1.3.1 Storage pool dynamics

Consider the following data:

$$\text{depth } D := 10m, \quad \text{length } L := 200m, \quad \text{width } B := 50m; \quad \text{and} \quad \text{flow } Q := 50m^3/s$$

From D follows for Froude's velocity $u_F = 10\ m/s$, with $u_F = \sqrt{g.D}$
and thus for the echo travelling time $T_E = 2L/u_F = 40\ s$.

The filling time $T_Q = (L.B.D)/Q$ of the storage pool is $T_Q = 2'000\ s$. Also the mean flow velocity along the pool is $u_Q = Q/(B.D) = 0.10\ m/s$, and its Froude Number value is $F = u_Q/u_F = 0.010$.

The echo traveling time is 50 times shorter than the filling time of the pool. Or the propagation effects are much faster than the overall filling effect. In other words the level may be considered as being the same all over the pool.
Note that the data used in the numerical example above indicate a comparatively small pool. Inflow basins tend to be much larger and deeper for typical storage hydro plants.
Thus the (low bandwidth) dynamic response of the overall pool level to variations of the inflow and outflow may be considered to be given by the dynamic volume balance only. And sloshing effects need not be present in such a model (as they are so much faster).

1.3.2 Long channel dynamics, subcritical flow

The length L and flow Q shall be the same, but the depth is reduced to $D := 2.5\ m$ and the width to $B := 10\ m$.
Then the Froude velocity is reduced to $u_F = 5m/s$ and the echo travelling time increases to $T_E = 80\ s$.
The Froude Number value is $F = 0.40$, which is still subcritical: water is still 'flowing', not 'shooting'.
And the overall filling time is reduced to $T_Q = (L.B.D)/Q = 100\ s$.

Now the sloshing dynamics as given by T_E are on the same time scale as the overall filling dynamics T_Q, and may no longer be neglected. They *must* be incorporated in the model. - The key element is that the ***horizontal*** acceleration of the water column within the compartments must be included in the model. This augments the first order dynamics of the storage pool into at least a third order one (see chapter 2).

Note that usually the channel length is much longer. Then evidently such basic elements need to be chained up to get a better spacial resolution and a higher bandwidth model, (see chapter 3).
This case covers most upstream / downstream reservoir dynamics in river flow hydro power plants. — A typical case is 'Birsfelden' on the Rhine near Basel, see sect.3.9 .

1.3.3 Long channel dynamics, supercritical flow

Consider now the same data, but with depth reduced to $D := 0.90m$, and thus Froude's velocity to $u_F = 3.0m/s$. The mean flow velocity is $u_Q = (Q/(B.D)) = 5.55m/s$. Such a flow will result for a channel bottom being steeper than usual. [2]

Thus the Froude Number value increases to $F = u_Q/u_F = 1.852$, which is supercritical. Water is now 'shooting', and no longer 'flowing'. As the water flows downstream faster than disturbances can travel back upstream, the echo travelling time goes to infinity, and sloshing is no longer present. Also any excessive forced outflow will not propagate upstream as with subcritical flow, but will simply lead to drying out of the outflow compartment.

The transition of dynamic response from subcritical to supercritical flow and back must be handled correctly by any dynamic model.

[2] Typical bottom slopes would be $\gtrsim 10^{-2}$ here, whereas $\lesssim 10^{-3}$ for subcritical flow.

1.3.4 Wide channel dynamics

So far the compartments extend laterally over the full width of the channel. Thus the model yields the mean flow velocity uniformly over the whole cross section. This may be a reasonable approximation as long as the channel geometry does not deviate too far from a uniform rectangular cross section. An intuitive upper bound (to avoid vortex detaching) would be a conical shape with slopes of ± 0.10 or $\pm 6^0$ per each wall and the bottom.
This upper bound fits well to a recent specific project ('NeueWelt', Birs, 2011, see section 7.2, (WGS 84 location: 47.525863 7.621635)).

Otherwise the compartment structure must be laterally extended to model the additional dynamics.

Two typical layouts may be considered:

- a long channel where a *shallow pool* at least ten times wider than the channel width is inserted over a partial length (say one fifth) of the total channel length.
- a long channel where *shoulders* are added laterally over the total channel length in order to absorb very high flows by flooding those shoulder forelands. The shoulders should be about three times wider and about three times shallower than the channel.

The first item is motivated by a recent specific project ('Hänggelgiesse', Linth-Kanal, 2012, WGS 84 location 41.16889 9.00915).

1.3.5 Short channel dynamics

Here water level changes in the channel shall *not* be small and slow (as has been assumed implicitly above), but large compared to depth and fast compared to flow travel time. Such cases are:

- outflows from large pools through a short channel with a lowered flap at its end,
- or similarly overflow through an open-ended spillway,
- a sudden large opening in a dam and subsequent emptying of the upstream reservoir ('dam break' study),
- or a 'surge wave' produced by a sudden large increase in flow (for instance downstream of a 'dam break')
- 'hydraulic jumps' in a channel, where a supercritical flow transits to a subcritical flow while essentially conserving both volume and momentum flows.

These transitions have in common, that from experiments the surface level difference may be at least 50 % of final water depth over typically a length of five times the final water depth. This motivates the designation 'short channel dynamics'.

1.4 Contents Overview

The aim of the following is to clarify the structure of the investigation and of this report.

In Chapter 2 the basic longitudinal element is investigated in detail. It consists of two longitudinally adjacent volume compartments and a connecting longitudinal momentum compartment. It shows the basic 'filling mode' (or 'open integrator mode') and the first (lowest frequency) 'sloshing mode' (or 'oscillatory mode') of the channel.
The first step is to obtain a 'first principles' model by writing down the basic equations of (dynamic) conservation of mass or volume and of momentum, where several ways of spacial discretisation shall be used. Also different upstream and downstream border conditions are looked into.
Then the basic dynamic properties (stability, etc.) are investigated.
The effects of numerical integration (as a time sampled loop) are a distinct problem in the mainstream 'space- and time-discretized' approach, see e.g. [1]. However this is not a problem here, as the integration routines in simulation environments for such systems of ODE's do handle this internally well enough.
Next the model equations are implemented in the 'Scilab'-environment, and its standard handling for simulation runs is given. And finally the transient response of the model is documented for a few typical operating cases, with discussion.

In Chapter 3 a higher order system is covered by connecting a number of such basic elements in series. The individual element will be shorter in length and thus have a higher sloshing frequency. And the series connection will produce a series of higher natural frequencies above the basic oscillatory mode. It will also produce locally steeper slopes in the spacial surface dynamics. – Here a single case of 20 such elements is investigated. This leads to 20 momentum equations plus 20 + 1 volume compartments, that is to a system of 41 first order ordinary differential equations. And this delivers 21 points for surface level values and 20 points for flow values along the channel. Extensive simulation work has shown that this spacial granularity is required to produce reasonably realistic surface wave shapes. And the simulation runtime is still reasonable. Thus the '20 elements' assumption shall be considered sufficient for practical purposes here. – However if only slow transient behavior is specified in a given project, then a lower order model (for say 4 or 8 elements) may well be sufficient.
This chapter will focus on the implementation of the model equations in 'Scilab', with an ordered list of operating cases, and the corresponding simulation results, with discussion.
A typical case ("Birsfelden") is used to illustrate the application and to compare the results to actual measurements.

In Chapter 4 the basic element from chapter 2 is extended in the lateral direction, but still neglecting the vertical acceleration from chapter 5. This is inserted into the series connection of chapter 3. One typical situation is investigated here.

Finally Chapter 5 addresses situations in short channels where the water surface level gradient variations are no longer small with respect to time and space. Then the vertical acceleration of the mass content in the volume compartment may no longer be neglected. Therefore the basic longitudinal model from chapter 2 is extended to include the vertical acceleration force. This will move the pressure at the bottom surface away from its hydrostatic value. And this modifies the pressure difference applied on the momentum compartment in the longitudinal direction...
The implementation in 'Scilab' is given.
Then the extended basic element is inserted into the series connection of those 20 elements of chapter 3, proceeding as above up to simulation results for six typical operating cases, with discussion.

In Chapter 6 three typical configurations of level control systems are investigated on three types of models.

First the shallow water equations (PDE) are solved using Laplace Transforms and applying some notions from sampled date control systems. A model based control design strategy is developed, which is known as 'impedance matching' in electrical engineering. It also yields a simple gain-scheduling rule for the controller gains.

Next a linear state space model with 'floating coefficients' is derived. It allows to study the effect of varying N (number of momentum compartments) and Froude number F (of the operating point) both in the frequency and time domains in a more compact manner than with the nonlinear time domain model from the previous chapters. The controller design from the previous section is tested here as a first step.

Then the controller design with gain-scheduling included is tested on the nonlinear time domain model for a typical variation of inflow to a typical turbine inflow channel.

Finally in Chapter 7 the design is applied to two application cases from specific past projects to check if typical specifications on level control performance are met.

All simulations are fully documented by their Scilab/Xcos file triplets
(diagram `s_... .zcos, s_... _context.sce, s_... _crunplot.sce`)
in the text to allow easy inspection of all details on structure and coefficients.

These file triplets are appended in a 'zip-container' to the 'report' in executable form for easy reproduction and further development (as 'open source'):
LevelDynamics_Scilab/Xcos_v541_sources.7z (Research Data Model) at
https://www.research-collection.ethz.ch/handle/20.500.11850/175543
Note that these files need to be run with Scilab version 5.4.1 [3]. Newer versions 5.5 and 6.0 are available but seem not to be stable enough at this time of publication.

A second 'zip-container' (also as 'open source')
LevelDynamics_Matlab/Simulink_R2016b_sources.7z (Research Data Model) at
https://www.research-collection.ethz.ch/handle/20.500.11850/175549
carries the corresponding Matlab/Simulink pairs using the release 'R2016b'
(`m_..._s .slx, m_... .m`).

[3] Download from www.scilab.org, → Download, and use link to 'Previous versions' at the page bottom, → scroll for 5.4.1

Chapter 2

The Longitudinal Basic Element

2.1 Modeling

The modeling will be done with the compartment method.
The momentum equation shall be addressed first, see Fig. 2.1

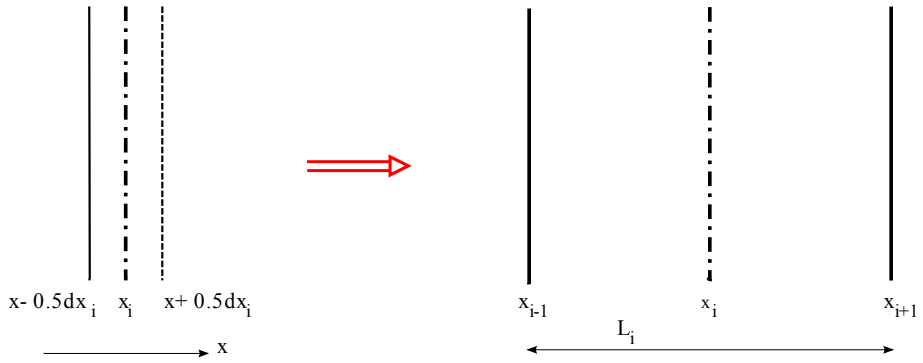


Figure 2.1: Longitudinal discretization: From the infinitesimal around x_i to the finite length compartment centered at x_i from x_{i-1} to x_{i+1}

.

For both the infinitesimal and the finite length cases, the center of mass of the compartment content is concentrated at its center of gravity at x_i. Assume further:

horizontal bottom, slope $S_b = 0$;

no friction, slope $S_f = 0$;

and constant channel width $B_i = B_{i-1} = B_{i+1}$

So the remaining 'forces' changing the momentum content are

pressure forces P_{i-1} and P_{i+1} (from the difference of hydrostatic pressure at the cross sections at x_{i-1} and x_{i+1};

reaction forces (from the momentum inflow $\overset{*}{I}_{i-1}$ and momentum outflow $\overset{*}{I}_{i+1}$ associated with the mass flows across both cross sections).

And the longitudinal momentum content I_i is (with mass M_i and depth D_i and volume flow Q_i):

$$I_i = M_i \cdot U_i = \rho \cdot (L_i \cdot B_i \cdot D_i) \cdot U_i \quad = \quad \rho \cdot L_i \cdot (B_i \cdot D_i \cdot U_i) \quad = \quad \rho \cdot L_i \cdot Q_i$$

This produces the (dynamic) momentum balance, see Fig.2.2.

$$\rho \cdot L_i \frac{d}{dt} Q_i = P_{i-1} - P_{i+1} \quad + \quad \overset{*}{I}_{i-1} - \overset{*}{I}_{i+1}$$

For the pressure forces and the reaction forces

$$\begin{aligned} P_{i-1} &= B_i \cdot D_{i-1} \cdot \rho \cdot g \cdot D_{i-1} \cdot \frac{1}{2} = \frac{1}{2}\, \rho\, g\, B_i\, D_{i-1}^2 \\ P_{i+1} &= \frac{1}{2}\, \rho\, g\, B_i\, D_{i+1}^2 \end{aligned}$$

$$\begin{aligned} \overset{*}{I}_{i-1} &= \rho \cdot B_i \cdot D_{i-1} \cdot U_{i-1} \cdot U_{i-1} = \rho \cdot Q_{i-1} \cdot U_{i-1} = \rho \cdot \frac{Q_{i-1}^2}{B_i \cdot D_{i-1}} \\ \overset{*}{I}_{i+1} &= \rho \cdot \frac{Q_{i+1}^2}{B_i \cdot D_{i+1}} \end{aligned}$$

That is for the momentum balance

$$L_i\, \frac{d}{dt} Q_i = \frac{1}{2}\, g\, B_i\, \left(D_{i-1}^2 - D_{i+1}^2\right) + \frac{1}{B_i} \left(\frac{Q_{i-1}^2}{D_{i-1}} - \frac{Q_{i+1}^2}{D_{i+1}} \right)$$

where ρ has been taken off from both sides.

Now the dynamic mass balances are addressed. This uses the concept of 'staggered' or 'interleaved' compartments (see also [1]): The upstream mass compartment is shifted one half length of L_i upstream to be centered on x_{i-1}, and correspondingly the downstream mass compartment is centered on x_{i+1}, see Fig.2.3:

$$L_i \cdot B_i \frac{d}{dt} D_{i-1} = (Q_{i-2} \; - \; Q_i)$$

and

$$L_i \cdot B_i \frac{d}{dt} D_{i+1} = (Q_i \; - \; Q_{i+2})$$

where ρ has been taken off from both sides again.
The reason for this interleaving is that the right hand ('input') side uses the volume flow Q_i from the left hand ('result') side of the momentum balance, and yields the depths D_{i-1}, D_{i+1} on the 'result' side which are needed for the pressure forces P_{i-1}, P_{i+1} on the input side of the momentum equation.

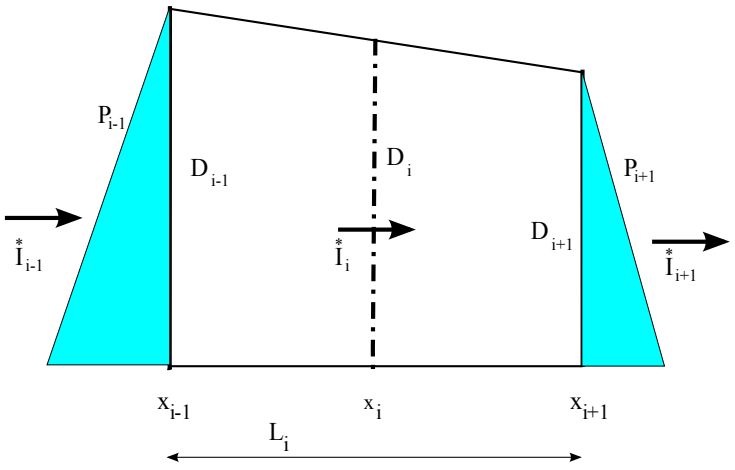


Figure 2.2: Pressure and reaction forces around x_i

.

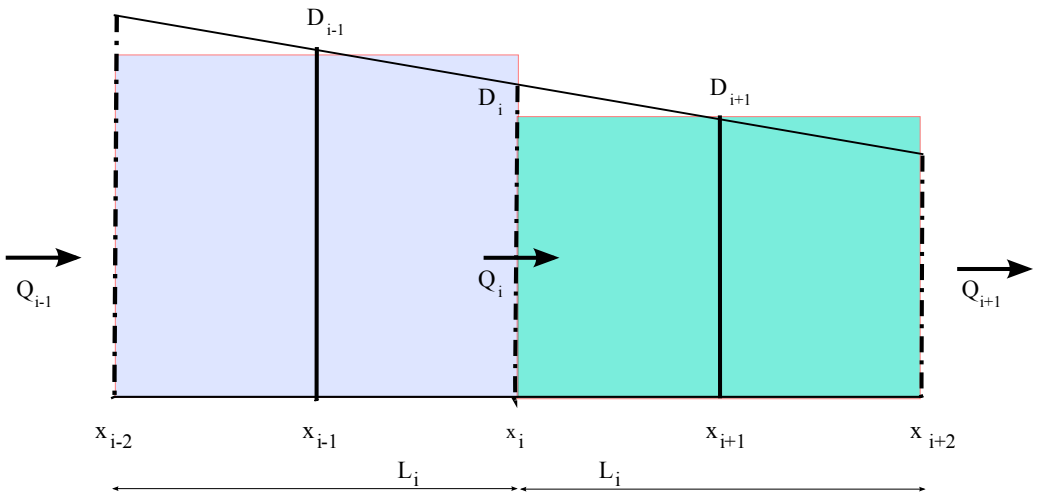


Figure 2.3: Mass or volume flow balances around x_{i-1} and x_{i+1}

But:
Note that the 'reaction' forces $\overset{*}{I}_{i-1}$, $\overset{*}{I}_{i+1}$ would need Q_{i-1}, Q_{i+1} which are *not* available as 'results' !

A first idea to resolve this would be to replace both reaction forces $\overset{*}{I}_{i-1}, \overset{*}{I}_{i+1}$ by arithmetic means:

$$\overset{*}{I}_{i-1} = \frac{1}{2}\left(\overset{*}{I}_{i-2} + \overset{*}{I}_{i}\right) \text{ and } \quad \overset{*}{I}_{i+1} = \frac{1}{2}\left(\overset{*}{I}_{i} + \overset{*}{I}_{i+2}\right)$$

But then in the momentum equation

$$\overset{*}{I}_{i-1} - \overset{*}{I}_{i+1} = \frac{1}{2}\left(\overset{*}{I}_{i-2} + \overset{*}{I}_{i} \quad - \quad \overset{*}{I}_{i} - \overset{*}{I}_{i+2}\right) = \frac{1}{2}\left(\overset{*}{I}_{i-2} - \overset{*}{I}_{i+2}\right)$$

Thus the local momentum flow $\overset{*}{I}_{i}$ is no longer present in the local momentum balance, which makes the response oscillate around its steady state (as remarked by C.Beffa, [3]).

A second idea is to shift the reaction forces difference upstream by one half compartment length,[1] that is

$$\overset{*}{I}_{i-1} - \overset{*}{I}_{i+1} \quad \rightarrow \quad \overset{*}{I}_{i-2} - \overset{*}{I}_{i}$$

In other words the momentum outflow $\overset{*}{I}_{i+1}$ from the compartment is now replaced by the momentum flow at its center (approx. its center of mass) $\overset{*}{I}_{i}$

Then for evaluating e.g. $\overset{*}{I}_{i}$:

$$\overset{*}{I}_{i} = \frac{Q_i^2}{D_i}$$

[1] This means that the 'Finite Element'(FE)-approach used up to now is replaced by the 'Finite Volume'(FV)-approach, (see [3])

Q_i now is available from the 'result' side of the momentum equation, but D_i is still not available directly from the volume balance equations. Essentially there are three alternatives,

1 use the downstream shifted value $D'_i := D_{i+1}$
This covers the FE-approach from Fig.2.2

2 compute and use the arithmetic mean $D'_i := 0.5 \cdot (D_{i-1} + D_{i+1})$
and this implements the FiniteVolume-approach from above.

3 or use the upstream shifted value $D'_i := D_{i-1}$.
This non-intuitive idea of taking the 'upwind' approximation is well established in the computational fluid dynamics (CFD)-field (see e.g. [4], [1]). The motivation for this strange notion will become clear in the next section.

The three alternatives can be covered by one sliding parameter $\varkappa = 0.0, 0.5, 1.0$ for cases 1, 2 and 3 respectively:

$$\begin{aligned} D'_i &:= \varkappa D_{i-1} + (1-\varkappa) D_{i+1} \\ D'_{i-2} &:= \varkappa D_{i-3} + (1-\varkappa) D_{i-1} \end{aligned}$$

All are approximations, and are the better the shorter the compartments are ....

2.2 Dynamic Properties

The system under investigation is of third order:

$$\begin{array}{rcccccc} L_i B_i \frac{d}{dt} D_{i-1} &=& & -Q_i & & +Q_{i-2} & \\ L_i \frac{d}{dt} Q_i &=& +\frac{1}{2} g B_i D_{i-1}^2 & -\frac{1}{B_i}\frac{Q_i^2}{D'_i} & -\frac{1}{2} g B_i D_{i+1}^2 & +\frac{1}{B_i}\frac{Q_{i-2}^2}{D'_{i-2}} & \\ L_i B_i \frac{d}{dt} D_{i+1} &=& & +Q_i & & & -Q_{i+2} \end{array}$$

It is ordered in typical state-space form, with time derivatives of the state variables D_{i-1}, Q_i, D_{i+1} on the left side. The first three columns on the right side contain the state variables and the second three are for the external inputs $Q_{i-2}, D'_{i-2}, Q_{i+2}$.

2.2.1 Linearized state space form

The next step is writing the state variables and external inputs as deviations $\delta Q, \delta D$ from a given steady state operating point at $\overline{Q}, \overline{D}$, for instance

$$\begin{aligned} Q_i(t) &\rightarrow \overline{Q}\left(1 + \frac{\delta Q_i(t)}{\overline{Q}}\right) \\ D_{i-1}^2 &\rightarrow \overline{D}^2\left(1 + 2\frac{\delta D_{i-1}(t)}{\overline{D}} + 0\right) \\ \frac{Q_i^2}{D'_i} &\rightarrow \frac{\overline{Q}^2}{\overline{D}}\left(1 + 2\frac{\delta Q_i(t)}{\overline{Q}} - \frac{\delta D'_i(t)}{\overline{D}} + 0\right) \\ &= \frac{\overline{Q}^2}{\overline{D}}\left(1 + 2\frac{\delta Q_i(t)}{\overline{Q}} - \varkappa\frac{\delta D_{i-1}(t)}{\overline{D}} - (1-\varkappa)\frac{\delta D_{i+1}(t)}{\overline{D}} + 0\right) \end{aligned}$$

and further using the abbreviations

$$\begin{aligned}
\frac{\delta D_{i-1}(t)}{\overline{D}} &\rightarrow x_1 \\
\frac{\delta Q_i(t)}{\overline{Q}} &\rightarrow x_2 \\
\frac{\delta D_{i+1}(t)}{\overline{D}} &\rightarrow x_3 \\
\frac{\delta Q_{i-2}(t)}{\overline{Q}} &\rightarrow u_1 \\
\frac{\delta D_{i-2}(t)}{\overline{D}} &\rightarrow u_2 \\
\frac{\delta Q_{i+2}(t)}{\overline{Q}} &\rightarrow u_3
\end{aligned}$$

And for the coefficients in the system

$$\begin{aligned}
\text{time constant for filling compartment } i-1 \quad & \frac{L_i B_{i-1} \overline{D}}{\overline{Q}} = T_1 \\
\text{time constant for accelerating compartment } i \quad & \frac{L_i \overline{Q}}{g\overline{D}^2 \overline{B}} = T_2 \\
\text{time constant for filling compartment } i+1 \quad & \frac{L_i B_{i+1} \overline{D}}{\overline{Q}} = T_3 \\
\text{coefficient for momentum flows in compartment } i \quad & \frac{\overline{Q}^2}{g\overline{B}^2\overline{D}^3} = \frac{\overline{B}^2\overline{D}^2\overline{U}^2}{g\overline{B}^2\overline{D}^3} = \frac{\overline{U}^2}{g\overline{D}} = \overline{F}^2 := \phi
\end{aligned}$$

Equations for x_1 and x_3 are straightforward, but equation for x_2 needs some further steps

$$\begin{aligned}
T_2 \frac{d}{dt} x_2 &= +1x_1 - 1x_3 + \phi\left[2u_1 - \varkappa u_2 - (1-\varkappa)x_1\right] - \phi\left[2x_2 - \varkappa x_1 - (1-\varkappa)x_3\right] \\
&= (1 - \phi + 2\varkappa\phi)x_1 - 2\phi x_2 - (1 - \phi + \varkappa\phi)x_3 + 2\phi u_1 - \varkappa\phi u_2
\end{aligned}$$

The equation for x_2 shall be rewritten in condensed form by using the abbreviations

$$\begin{aligned}
a &= (1 - (1 - 2\varkappa)\phi) \\
b &= (1 - (1 - \varkappa)\phi) \\
c &= (\varkappa\phi) \\
p &= (2\phi) = (2\overline{F}^2)
\end{aligned}$$

Then

$$\begin{array}{lcccccc}
T_1 \frac{d}{dt} x_1 & = & & -1 \cdot x_2 & & +1 \cdot u_1 & \\
T_2 \frac{d}{dt} x_2 & = & a \cdot x_1 & -p \cdot x_2 & -b \cdot x_3 & +p \cdot u_1 & -c \cdot u_2 \\
T_3 \frac{d}{dt} x_3 & = & & +1 \cdot x_2 & & & -1 \cdot u_3
\end{array}$$

2.2.2 Stability properties

The standard way to check the stability of this continuous linear system is by analyzing the characteristic equation [2].

$$0 \;=\; \begin{vmatrix} sT_1 & +1 & 0 \\ -a & sT_2+p & +b \\ 0 & -1 & sT_3 \end{vmatrix}$$

$$\;=\; s^2T_1T_3(sT_2+p) \;+\; asT_1 \;+\; bsT_3 \;+\; 0 \;+\; 0$$

Let $T_3 := T_1$. Then:

$$0 \;=\; s^3\cdot(T_1^2T_2) \;+\; s^2\cdot(T_1^2\cdot p) \;+\; s^1\cdot(2T_1)\cdot(a+b) + s^0\cdot(0)$$

$$\text{with}\quad a+b \;=\; (1-\phi+2\varkappa\phi)+(1-\phi+\varkappa\phi) \;=\; 2\cdot(1-(1-\frac{3}{2}\varkappa)\phi)$$

For the three values of $\varkappa$ from above and a fourth value of $\varkappa := 2/3$

$$\varkappa = 1.0 \;:\quad 0 \;=\; s(2T_1)\cdot\left[\; s^2\frac{T_1T_2}{2} \;+\; s(\phi\cdot T_1) \;+\; (1+\frac{\phi}{2}) \;\right]$$

$$\varkappa = 0.5 \;:\quad 0 \;=\; s(2T_1)\cdot\left[\; s^2\frac{T_1T_2}{2} \;+\; s(\phi\cdot T_1) \;+\; (1-\frac{\phi}{4}) \;\right]$$

$$\varkappa = 0.0 \;:\quad 0 \;=\; s(2T_1)\cdot\left[\; s^2\frac{T_1T_2}{2} \;+\; s(\phi\cdot T_1) \;+\; (1-\phi) \;\right]$$

$$\varkappa = \frac{2}{3} \;:\quad 0 \;=\; s(2T_1)\cdot\left[\; s^2\frac{T_1T_2}{2} \;+\; s(\phi\cdot T_1) \;+\; (1-0\cdot\phi) \;\right]$$

which exhibits a single zero at the origin (yielding an open integrator response corresponding to the overall filling behaviour with time constant $2T_1$), which is independent of $\varkappa$
and a pair of zeros (yielding the oscillatory mode of the sloshing), which depend on $\varkappa$ and $\phi = \overline{F^2}$.
The response of the sloshing subsystem is asymptotically stable (shortened to 'a.s') as long as all coefficients in the square brackets [...] are positive. Thus only the last term needs to be considered here if $F \neq 0$

- For $\varkappa = 0.0$, the last term increases with $\phi = \overline{F^2}$, that is the sloshing subsystem response will be a.s. only for all Froude number values $0 < \overline{F} < 1.0$, that is only for *subcritical* flow regimes.
- For $\varkappa = 0.5$, the a.s.-region extends somewhat into the supercritical domain, $0 < \overline{Fr} \leq 2.0$,
- and for $\varkappa \geq 2/3$, the model response will be stable on the whole domain of Froude numbers $\overline{F} > 0$.

Remarks:
It is not evident from the beginning of simulating transients for a given situation that the flow regime over time will always remain within the stability bounds for F. Therefore it is strongly recommended to use only $\varkappa \geq 2/3$ for the simulation model.
Note that setting $\varkappa = 1.0$ means that only D_{i-1} may enter into the calculation of the momentum outflow at i. This finding also neatly confirms the assumption in [1] that "always the *upstream grid point* on water depth should be used while discretizing the momentum equation".
Note that this rule holds for the classic discretisation of the partial differential equation, where a finite but *short* compartment length L_i is used, and where only grid point values of the dependent variables are used. In our case however, we use much longer compartment lengths. And it is intuitively preferable to use a value of D_i' which is as close as possible to D_i in order to minimize the discretisation error incurred by using D_{i-1} in the momentum equation. – $\varkappa = 1.0$ is used in the simulations...

[2] Note that the stability analysis is straightforward in the 'compartment' approach, whereas in the 'time- and space-discretization' approach ([2],[1]) it is less so...

2.2.3 Analysis of the model coefficients

For the volume compartment filling times T_1, T_3:

$$T_1 = T_3 = \frac{L \cdot B \cdot \overline{D}}{\overline{Q}} = \frac{L \cdot B \cdot \overline{D}}{\overline{B} \cdot \overline{D} \cdot \overline{U}} = \frac{L}{\overline{\overline{U}}}$$

and for the momentum compartment acceleration time T_2

$$T_2 = \frac{L\overline{Q}}{g\overline{D}^2 B} = \frac{L \cdot \overline{B} \cdot \overline{D} \cdot \overline{U}}{g\overline{D}^2 B} = \frac{L \cdot \overline{U}}{g\overline{D}} = \frac{L}{\overline{\overline{U}}} \cdot \frac{\overline{U}^2}{g\overline{D}} = \frac{L}{\overline{\overline{U}}} \cdot \overline{F}^2$$

that is, while using for the Froude ground wave propagating velocity $\overline{U}_F = \sqrt{g \cdot \overline{\overline{D}}}$

$$T_1 \cdot T_2 = \frac{L}{\overline{\overline{U}}} \cdot \frac{L \cdot \overline{U}}{g \cdot \overline{D}} = \frac{L^2}{g \cdot \overline{D}} = \left(\frac{L}{\overline{\overline{U}}_F} \right)^2$$

that is the coefficient on s^2 is *independent* of $\overline{Fr}$.

And further the coefficient on s:

$$T_1 \cdot \phi = \frac{L}{\overline{\overline{U}}} \cdot \left(\frac{\overline{U}^2}{g \cdot \overline{D}} \right) = \frac{L}{\overline{\overline{U}}} \cdot \frac{\overline{U}^2}{\overline{U_F}^2} = \frac{L}{\overline{\overline{U}}_F} \cdot \overline{F}$$

increases linearly with the Froude number $\overline{Fr}$.

Consider now the case of $\overline{Fr} > 0$ with $\varkappa = 2/3$. The part in sqare brackets [...] yields a second order oscillator with resonance frequency Ω_F and damping ratio $2D_F$:

$$\begin{aligned} 0 &= s \cdot (2T_1) \left[s^2 \cdot \left(\frac{T_1 T_2}{2} \right) \; + \; s \cdot (T_1 \phi) \; + \; 1.0 \right] \\ &\stackrel{!}{=} s \cdot (2T_1) \left[\left(\frac{s}{\Omega_F} \right)^2 \; + \; (2D)_F \left(\frac{s}{\Omega_F} \right) \; + \; 1.0 \right] \end{aligned}$$

$$\text{with} \quad \Omega_F = \sqrt{\frac{2}{T_1 T_2}} = \omega_E$$

$$\text{and} \quad 2D_F = \Omega_F \cdot \phi \cdot T_1 = \sqrt{\frac{2}{T_1 T_2}} \cdot T_1 \phi = \sqrt{2} \sqrt{\frac{T_1}{T_2}} \cdot \phi = \sqrt{2} \sqrt{\phi} \cdot \phi = \sqrt{2}\, \overline{F}$$

that is if $F := 1.0$, then $2D_F = \sqrt{2}$, and if $F = \sqrt{2}$ then $2D_F = 2.0$

And for the time constant T_F of the open integrator part

$$T_F = 2\, T_1$$

For $\phi := 0$ the response generated by the part in brackets [...] is no longer asymptotically stable, but still stable. It is an undamped periodic oscillation with ω_F and associated period T_F:

$$T_F = \frac{2\pi}{\overline{\omega_F}} = \frac{2\pi}{\sqrt{2}} \cdot \frac{L}{\overline{U}_F} = 4.443 \cdot \frac{L}{\overline{U}_F}$$

Compare this to the echo traveling time $T_{echo} = 4.0 L/\overline{U}_F$, which is about 11% less than T_F. This deviation is considered acceptable within this framework. – Note that this also holds if both $\varkappa = 1.0$ and $F = 0$.

2.2.4 Extensions to the basic model

The following extensions are considered in the next sections

- adding friction losses
- admitting nonzero channel bottom slope but equal channel width
- and admitting both conical side walls and non-zero bottom slope
- considering other inflow and outflow conditions

2.3 Adding friction on bottom and side walls

The usual GMS-law [3] shall be used here, which is well established with engineers in this area. Note that it assumes steady state flow both in time and in spacial directions, that is no acceleration of flow. So it will be not more than an approximation for friction losses in the dynamic context considered here. At least in steady state conditions on typical channels with more or less constant cross section, it should be valid for modeling. But its validity for transients remains doubtful, and no error margins are available.

The standard formula for the GSM-law is:

$$\overline{U} = k_s \; R^{2/3} \; I^{1/2} \quad \text{or} \quad \overline{U}^2 = k_s^2 \; R^{4/3} \; I$$

where

- $\overline{U}$ is the mean longitudinal flow velocity, $\overline{U} = \overline{Q}/(B \cdot \overline{D})$.
- k_s is the Strickler-coefficient, which depends essentially on the bottom roughness (see Table 2.1)
- R is the 'hydraulic radius' defined by $R := (B \cdot \overline{D}) \; / \; (B + 2 \cdot \overline{D})$ for a rectangular cross section $B \times D$.
- I is the longitudinal level slope $I := \Delta H_i / L_i$, with H_i as the water level deviation from a given geodesic horizon.

The bottom deviation from the said horizon is denoted as S_i. Then the bottom slope $\Delta S_i / L_i$ is implied in the GMS-law to be equal to the water level slope I_i, such that water depth $D_i := H_i - S_i$ is constant along x_i. Also both slopes are assumed constant along x_i.

Table 2.1: Strickler coefficients

surface	$k_s, \quad \approx$
steel, smooth concrete	100
bitumen	70
brickwork	50
gravel	40
natural river bed	30
mountain river bed	20

Next R shall be addressed. To make the discussion more transparent and simpler, the following approximations can be made:

$$R_i = \frac{B_i \cdot D_i}{B_i + 2D_i} = \frac{D_i}{1 + \frac{2D_i}{B_i}}, \qquad \text{and for} \;\; B_i \gg D_i \qquad \text{set (approx.)} \qquad R_i \;\; := \;\; D_i$$

[3] Gaukler-Manning-Strickler

and also by setting (approx.)

$$R_i^{4/3} \;:=\; D_i$$

The friction force F_{f_i} for the momentum balance is calculated next. With $I_i = \Delta H_{i_f}/L_i$ the GMS-law is:

$$U_i^2 \;\approx\; k_s^2 \cdot D_i \cdot \left(\frac{\Delta H_{i_f}}{L_i}\right)_f \quad \text{or} \quad D_i \cdot \Delta H_{i_f} = \frac{L_i}{k_s^2} \cdot U_i^2$$

$$\text{where} \quad D_i \cdot \Delta H_{i_f} \;\approx\; \frac{1}{2}(D_{i-1} + D_{i+1})(D_{i-1} - D_{i+1})_f = \frac{1}{2}\left[D_{i-1}^2 - D_{i+1}^2\right]_f$$

Compare to the pressure force difference

$$(P_{i-1} - P_{i+1}) \;=\; \frac{1}{2}\rho g B_i \left[D_{i-1}^2 - D_{i+1}^2\right]_p \quad \text{and thus} \quad F_{f_i} = \frac{1}{2}\rho g B_i \left[D_{i-1}^2 - D_{i+1}^2\right]_f \;=\; \rho g B_i \frac{L_i \cdot U_i^2}{k_s^2}$$

$$\text{and further} \quad F_{f_i} \;=\; \frac{g}{k_s^2} \cdot \frac{L_i}{D_i} \cdot \rho \cdot \frac{Q_i^2}{(B_i D_i)} \quad \text{finally} \quad F_{f_i} = \left[\frac{g}{k_s^2} \cdot \frac{L_i}{D_i}\right] \cdot \overset{*}{I}_i \;=\; \psi \cdot \overset{*}{I}_i$$

with the relative weight ψ of the friction force to the momentum reaction force.
Consider some quantitative cases:

- with $L = 10m$, $\overline{D} = 2.5m$ and $k_s = 70$, then $\psi \approx 100/(2.5 * 5000) = 0.008$,
- and for $L = 200m$ this increases to $\psi = 0.16$.
- But for $L = 150m$, $\overline{D} = 1.2m$, and with $k_s = 25$ (aka 'NeueWelt', natural river zone) then $\psi = 2.0$,
- and for $L = 4'000m$, $\overline{D} = 8m$, and with $k_s = 32.5$ (aka 'Birsfelden') then $\psi = 4.73$!

So the friction contribution to damping can be either significantly smaller or greater than the reaction force contribution...!

It was assumed that the surface level and the bottom level are parallel. In other words the depth D is constant along x. Thus the bottom needs to be inclined accordingly

$$(S_{i-1} - S_{i+1})_f = \Delta S_{i_f} \overset{!}{=} \Delta H_{i_f} = (H_{i-1} - H_{i+1})_f$$

where S_{i+k} is the geodetic height of the bottom at x_{i+k} for $k = -1, 0+1$ with respect to the 'horizon', and is taken positive for the upward direction. And for the surface level difference ΔH_{i_f} from the GMS-law:

$$\Delta H_{i_f} := \frac{\overline{U_i}^2 \cdot L_i}{k_s^2 \cdot \overline{R}^{4/3}} \quad \text{with} \quad \overline{U_i} = \frac{\overline{Q_i}}{B_i \cdot \overline{D_i}} \quad \text{and with} \quad R = \frac{B_i \overline{D_i}}{B_i + 2\overline{D_i}}$$

Usually the 'reference' or 'design' operating point or condition of the channel is given by U_{ref}, D_{ref} or $Q_{ref}, B_{ref}, D_{ref}$. then $\Delta H_{Fr_{ref}}$ can be evaluated directly, and thus the 'reference' bottom slope ΔS_{ref}.

To get an idea how a different $Q \neq Q_{ref}$ will affect both D and U, let again $R^{4/3} \approx D$.
For steady state conditions, again $\Delta H_f := \Delta S_{ref}$,and with $B := B_{ref}, k_s^2 := k_{s_{ref}}^2$

$$\begin{aligned}
\Delta S \;&=\; \frac{U^2 \cdot L}{k_s^2 D} \;=\; \frac{L}{k_s^2} \cdot \frac{Q^2}{B^2 D^3} \\
\Delta S_{ref} \;&=\; \frac{U_{ref}^2 \cdot L}{k_s^2 D_{ref}} \;=\; \frac{L}{k_s^2} \cdot \frac{Q_{ref}^2}{B^2 D_{ref}^3} \\
\rightarrow \quad \frac{\Delta S}{\Delta S_{ref}} \;&=\; 1.0 \;=\; \left(\frac{Q}{Q_{ref}}\right)^2 \cdot \left(\frac{D_{ref}}{D}\right)^3 \\
\text{that is} \quad \frac{D}{D_{ref}} \;&=\; \left[\left(\frac{Q}{Q_{ref}}\right)\right]^{(2/3)} \quad \text{and} \quad \frac{U}{U_{ref}} = \left[\left(\frac{Q}{Q_{ref}}\right)\right]^{(1/3)}
\end{aligned}$$

Note that avoiding the approximation on $R^{(4/3)}$ from above leads to an implicit equation for D, which will have to be solved iteratively (see subsection 2.6.3).

2.4 Pressure forces on non-zero bottom slope and constant width

Consider the situation shown in Fig.2.4 (lateral or side view).

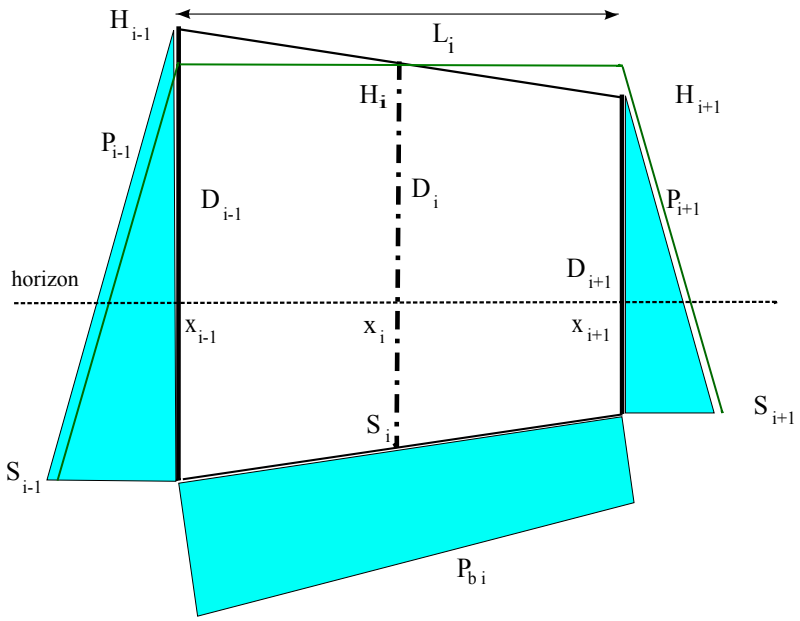


Figure 2.4: Pressure forces around x_i for a non-zero bottom slope

.

Let

$$\begin{aligned} & B_{i-1} = B_i = B_{i+1} \\ \text{but} \quad & S_{i-1} \neq S_i \neq S_{i+1}; \quad \text{where} \quad S_i \stackrel{!}{=} \frac{1}{2} \cdot (S_{i-1} + S_{i+1}) \end{aligned}$$

that is the bottom slope is constant over x_{i-1} to x_{i+1} ('linear interpolation').
The bottom slope $(S_{i-1} - S_{i+1})/L$ shall be small, such that the cosine of the angle with the horizontal still is ≈ 1.0

For the pressure force balance, three elements must be considered, on the left (P_{i-1}) and on the right hand (P_{i+1}) cross sections, and the horizontal component (P_i) of the pressure force on the bottom.

$$\begin{aligned} P_{i-1} &= \frac{1}{2}\rho g B_i D_{i-1}^2 \\ P_{i+1} &= \frac{1}{2}\rho g B_i D_{i+1}^2 \\ p(x)_{hor} &= \rho g D(x) \frac{(S_{i-1} - S_{i+1})}{L_i} \end{aligned}$$

and integrated over x

$$\begin{aligned} P_{i_{hor}} &= B_i L_i \rho g \frac{(S_{i-1} - S_{i+1})}{L_i} \int_0^{L_i} D(x) dx \\ &= B_i L_i \rho g \frac{(S_{i-1} - S_{i+1})}{L_i} \int_0^{L_i} \left[D_{i-1} - (D_{i-1} - D_{i+1})\, x \right] dx \\ &= B_i L_i \rho g \frac{(S_{i-1} - S_{i+1})}{L_i} \left[D_{i-1} (L_i - 0) - (D_{i-1} - D_{i+1}) \frac{1}{2} (L_i - 0) \right] \end{aligned}$$

$$
\begin{aligned}
P_{i_{hor}} &= B_i L_i \rho g (S_{i-1} - S_{i+1}) \left[D_{i-1} - (D_{i-1} - D_{i+1}) \frac{1}{2} \right] \\
&= B_i L_i \rho g (S_{i-1} - S_{i+1}) \frac{1}{2} \left[D_{i-1} + D_{i+1} \right] \\
&= \rho g B_i D_i [S_{i-1} - S_{i+1}]
\end{aligned}
$$

- The last equation implies that the water surface has constant slope from x_{i-1} to x_{i+1} ('linear interpolation', no curvature).
- If $(S_{i-1} < S_{i+1})$ (as shown in Fig.2.4), and the depth $S(x)$ is taken as negative from the horizon downwards (as in Fig.2.4), then $P_{i_{hor}}$ is automatically < 0

Then the pressure force balance is

$$
\begin{aligned}
\Delta P_{i_{hor}} &= P_{i-1} - P_{i+1} + P_{i_{horiz}} \\
&= \rho g B_i \frac{1}{2} \left[D_{i-1}^2 - D_{i+1}^2 \right] + \rho g B_i D_i [S_{i-1} - S_{i+1}] \\
&= \rho g B_i \left\{ \frac{1}{2} (D_{i-1} + D_{i+1}) \cdot (D_{i-1} - D_{i+1}) + D_i [S_{i-1} - S_{i+1}] \right\} \text{ with } \frac{1}{2}(D_{i-1} + D_{i+1}) = D_i \\
&= \rho g B_i D_i \left\{ (D_{i-1} - D_{i+1}) + (S_{i-1} - S_{i+1}) \right\} \text{ where } D_{i+k} = H_{i+k} - S_{i+k} \text{ for } k = -1, 0, +1 \\
&= \rho g B_i D_i \left\{ (H_{i-1} - S_{i-1}) - (H_{i+1} - S_{i+1}) + (S_{i-1} - S_{i+1}) \right\}
\end{aligned}
$$

$$
\Delta P_{i_{hor}} = \rho g B_i D_i \left\{ H_{i-1} - H_{i+1} \right\} = \rho g B_i \; D_i \; \Delta H_i
$$

- Note that H_{i+k} is taken positive if for values upwards of the horizon, see Fig.2.4,
- and if the content is not moving at all, then $H_{i-1} - H_{i+1} \stackrel{!}{=} 0$ (see the green contour in Fig.2.4) and thus $\Delta P_{i_{hor}} = 0$, and the content stays at rest, q.e.d.

Remark

A second approach to compute the pressure force balance for non-zero bottom slope is to consider a mass point on an inclined plane, with the cross section pressure forces acting on it.
The weight force of the content is

$$
F_{G_i} = g \rho V = g \rho L_i B_i D_i
$$

and its component parallel to the inclined bottom plane is

$$
F_{G_{incl}} = F_{G_i} \frac{S_{i-1} - S_{i+1}}{L_i} = \rho g B_i D_i (S_{i-1} - S_{i+1})
$$

and due to the small inclination angle assumed above, this is also approx. equal to its horizontal component $F_{G_{hor}}$.
This is the same result as for $P_{i_{hor}}$ from above. Using the (horizontal) pressure forces on the cross sections from above, the force balance will be the same, and finally also

$$
\Delta P_i = \rho g B_i \; D_i \; \Delta H_i
$$

2.5 Pressure forces on conical side walls and non-zero bottom slope

Consider the geometry of the element in Fig.2.5

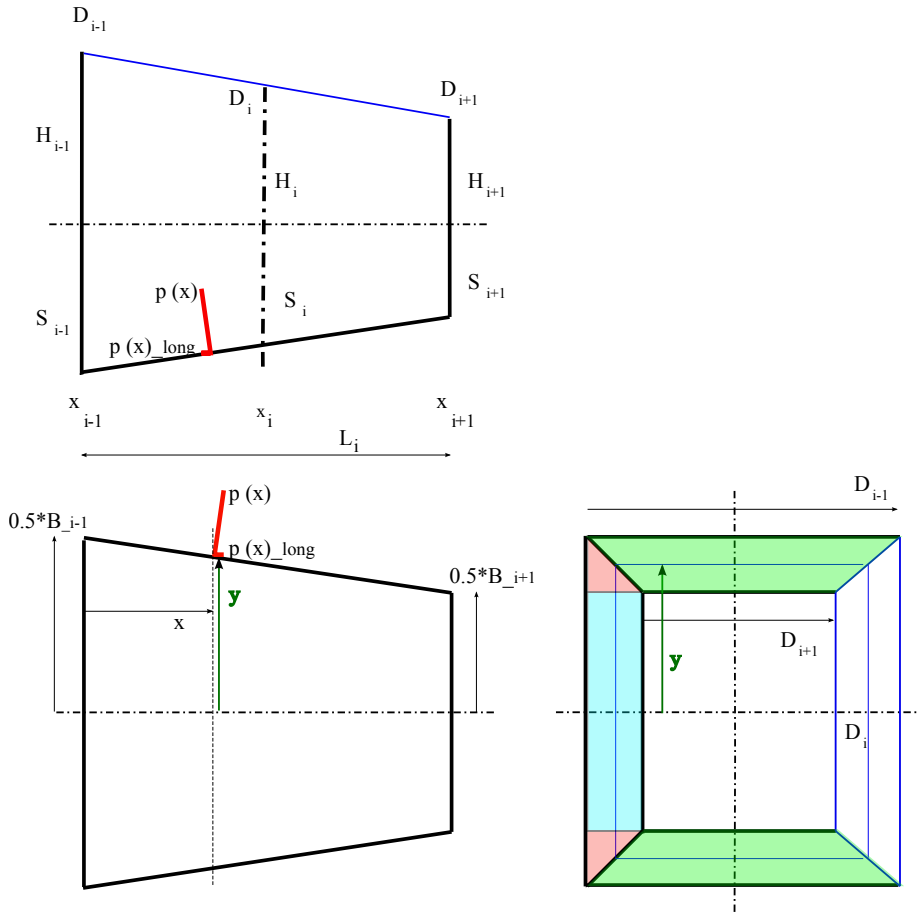


Figure 2.5: Pressure forces for conical side walls and zero bottom slope,
top: side view, bottom left: top view, bottom right: longitudinal view, in direction of flow

2.5.1 cross section surfaces

Pressure forces on the cross sections are:

$$P_{i-1} = D_{i-1} \cdot B_{i-1} \cdot \frac{1}{2}\rho g D_{i-1} \quad \text{and} \quad P_{i+1} = D_{i+1} \cdot B_{i+1} \cdot \frac{1}{2}\rho g D_{i+1}$$

2.5.2 side wall surfaces

On one side wall (shaded green in Fig.2.5, the local mean pressure orthogonal to the wall surface is at position y

$$p_{orth}(y) = \rho g \frac{1}{2} D^2(y)$$

and the longitudinal component is

$$p_{long}(y) = \frac{0.5(B_{i-1} - B_{i+1})}{L_i} \cdot p_{orth}(y)$$

Then the longitudinal pressure force at position y with strip width $d\ell$ is

$$dP_{long}(y) = p_{long}(y) \cdot d\ell = \frac{0.5(B_{i-1} - B_{i+1})}{L_i} \cdot p_{orth}(y) \cdot d\ell = \frac{0.5(B_{i-1} - B_{i+1})}{L_i} \cdot \rho g \frac{1}{2} D^2(y) \cdot d\ell$$

here $d\ell$ must be replaced by dy for integration, where

$$d\ell = \frac{L_i}{0.5(B_{i-1} - B_{i+1})} \cdot dy$$

that is

$$dP_{long}(y) = \frac{0.5(B_{i-1} - B_{i+1})}{L_i} \cdot \rho g \frac{1}{2} D^2(y) \cdot \frac{L_i}{0.5(B_{i-1} - B_{i+1})} \cdot dy = \rho g \frac{1}{2} D^2(y) dy$$

Next $D(y)$ is a linear function of dy, to be inserted

$$D(y) = D_{i-1} - (D_{i-1} - D_{i+1}) \cdot \frac{y}{0.5(B_{i-1} - B_{i+1})} = D_{i-1} - \delta_i \eta$$

where for $y = y_{start} := 0$ is $D(y) = D_{i-1}$ and $\eta = 0$
and for $y = y_{end} := 0.5(B_{i-1} - B_{i+1})$ is $D(y) = D_{i+1}$ and $\eta = 1.0$

Integrating over the lateral wall surface is:

$$\begin{aligned}
P_{long} &= \rho g \frac{1}{2} \int_0^{0.5(B_{i-1} - B_{i+1})} D^2(y) dy = \rho g \frac{1}{2} \left[0.5(B_{i-1} - B_{i+1})\right] \int_0^1 D^2(\eta) d\eta \\
&= \alpha \int_0^1 D^2(\eta) d\eta \qquad \text{which defines abbreviation } \alpha \\
&= \alpha \cdot \int_0^1 (D_{i-1} - \delta_i \cdot \eta)^2 \cdot d\eta = \alpha \cdot \left[D_{i-1}^2 \int_0^1 1 \cdot d\eta \; - \; 2 D_{i-1} \delta_i \int_0^1 \eta \cdot d\eta \; + \; \delta_i^2 \int_0^1 \eta^2 \cdot d\eta \right] \\
&= \alpha \cdot \left[D_{i-1}^2 \left[1 - 0\right] \; - \; 2 D_{i-1} \delta_i \frac{1}{2} \left[1 - 0\right] \; + \; \delta_i^2 \frac{1}{3} \left[1 - 0\right] \right] \; = \; \alpha \left[D_{i-1}^2 \; - \; D_{i-1} \delta_i \; + \; \delta_i^2 \frac{1}{3} \right] \\
&= \alpha \left[D_{i-1}^2 \; - \; D_{i-1} \left(D_{i-1} - D_{i+1}\right) \; + \; \frac{1}{3} \left(D_{i-1}^2 - 2 D_{i-1} D_{i+1} + D_{i+1}^2 \right) \right] \\
P_{long} &= \alpha \frac{1}{3} \left[D_{i-1}^2 + D_{i-1} D_{i+1} + D_{i+1}^2 \right]
\end{aligned}$$

A short check: Let $D_{i-1} := D_{i+1} := D_i$ then $P_{long} = \alpha \frac{1}{3} \left[D_i^2 + D_i^2 + D_i^2 \right] = \alpha D_i^2$ q.e.d.

The next step is to replace D_{i-1}, D_{i+1} by variables D_i, Δ_i:

$$\begin{aligned}
D_i &= \frac{1}{2} \left(D_{i-1} \; + \; D_{i+1} \right) \\
\Delta_i &= \frac{1}{2} \left(D_{i-1} \; - \; D_{i+1} \right)
\end{aligned}$$

Inserting this in the last result above yields after a short bit of algebra:

$$P_{long} \; = \; \alpha \frac{1}{3} \left[3 \cdot D_i^2 - 0 \cdot D_i \Delta_i + 1 \cdot \Delta^2 \right] \; = \; \alpha D_i^2 \left[1 \; + \; \frac{1}{3} \left(\frac{\Delta_i}{D_i} \right)^2 \right]$$

And finally the longitudinal pressure force on *both* side walls is twice this value, due to assumed geometrical symmetry.

$$P_{w_{long}} = \rho g \left(B_{i-1} - B_{i+1} \right) \cdot D_i^2 \left[1 \; + \; \frac{1}{3} \left(\frac{\Delta_i}{D_i} \right)^2 \right]$$

A numerical example: Let $|\Delta_i / D_i| \leq 0.20$, then the deviation would be $\leq 1.33\%$, if the simplified relation $P_{w_{long}} := \rho g \left(B_{i-1} - B_{i+1} \right) \cdot D_i^2$ valid for the *constant cross section* would be used instead.
In other words the effect of moderately conical side walls is weak, and the constant cross section around x_i is a reasonable approximation.

2.5.3 bottom surfaces

Consider again Fig.2.5. The bottom surface is split into a rectangular part (shaded blue) and two triangular parts (shaded red). This is to simplify the integration.

The *rectangular part* has been investigated in the previous section, with the result for the longitudinal pressure force, adated to the current geometry:

$$P_{i_{blue}} = \rho g B_{i+1} \left(S_{i-1} - S_{i+1}\right) D_i$$

For one *triangular part* let

$$\Delta B_i := \frac{1}{2}\left(B_{i-1} - B_{i+1}\right) \quad \text{and} \quad \Delta S_i := \frac{1}{2}\left(S_{i-1} - S_{i+1}\right)$$

Then

$$\begin{aligned} p_i(\eta) &= \rho g D(\eta) = \rho g \left[D_{i+1} + (D_{i-1} - D_{i+1})\eta_i\right] \\ \ell_i(\eta) &= \Delta B_i \cdot \eta \\ d\ell &= \Delta S_i \cdot d\eta \\ \text{and for the surface at } \eta \quad dA_i(\eta) &= \ell_i(\eta) d\ell \;=\; \Delta B_i \cdot \Delta S_i \cdot \eta d\eta \\ dP_i(\eta) &= \rho g \Delta B_i \cdot \Delta S_i \left[D_{i+1} + (D_{i-1} - D_{i+1})\eta_i\right] \cdot \eta_i d\eta \\ &= \rho g \Delta B_i \cdot \Delta S_i \left[D_{i+1} \cdot \eta_i + (D_{i-1} - D_{i+1})\eta_i^2\right] d\eta \end{aligned}$$

and integrating

$$\begin{aligned} P_{i_{red}} &= \rho g \Delta B_i \cdot \Delta S_i \left[D_{i+1} \int_0^1 \eta_i d\eta + (D_{i-1} - D_{i+1}) \int_0^1 \eta_i^2 d\eta\right] \\ &= \rho g \Delta B_i \cdot \Delta S_i \left[D_{i+1}\frac{1}{2} + (D_{i-1} - D_{i+1})\frac{1}{3}\right] \\ &= \rho g \Delta B_i \cdot \Delta S_i \left[\frac{1}{3}D_{i-1} + \frac{1}{6}D_{i+1}\right] \end{aligned}$$

Again using D_i, Δ_i instead of D_{i-1}, D_{i+1}, that is

$$\begin{aligned} \left[\frac{1}{3}D_{i-1} + \frac{1}{6}D_{i+1}\right] &= \frac{1}{3}\left[D_{i-1} + \frac{1}{2}D_{i+1}\right] \\ &= \frac{1}{3}\left[(D_i + \Delta_i) + \frac{1}{2}(D_i - \Delta_i)\right] \\ &= \frac{1}{3}\left[\frac{3}{2}D_i + \frac{1}{2}\Delta_i\right] \\ &= \frac{1}{2}\left[D_i + \frac{1}{3}\Delta_i\right] \end{aligned}$$

and for *both* red triangles

$$2 \cdot P_{i_{red}} = \rho g \Delta B_i \cdot \Delta S_i \cdot D_i \left[1 + \frac{1}{3}\frac{\Delta_i}{D_i}\right]$$

Finally for the total bottom surface:

$$P_{i_{bottom}} = \rho g \cdot \Delta S_i \cdot D_i \left[B_{i+1} + \Delta B_i \left[1 + \frac{1}{3}\frac{\Delta_i}{D_i}\right]\right]$$

with

$$B_{i+1} + \Delta B_i = B_{i+1} + \frac{1}{2}\left(B_{i-1} - B_{i+1}\right) \quad = \quad \frac{1}{2}\left(B_{i-1} + B_{i+1}\right) \quad = \quad B_i$$

and thus

$$P_{i_{bottom}} = \rho g \cdot \Delta S_i \cdot D_i \cdot B_i \left[1 + \frac{\Delta B_i}{B_i} \cdot \frac{1}{3} \cdot \frac{\Delta_i}{D_i}\right]$$

A numerical example: Let $\frac{\Delta B_i}{B_i} := 0.20$ and $\frac{\Delta_i}{D_i} := 0.20$. Then a deviation of 1.333% results, if the simplified relation $P_i := \rho g \cdot \Delta S_i \cdot D_i \cdot B_i$ valid for the *constant cross section* would be used instead. Again the effect of a moderate bottom slope is weak, and the constant cross section element around x_i is a reasonable approximation.

2.6 Modeling Operating Conditions

The aim here is to check whether the model derived so far behaves as expected from experience on a selection of test cases.

Two typical 'plant' layouts are considered

- a section of a river with both constant and varying cross rectangular section (note that this is meant as a rough approximation of the natural trapezoidal shape) with GMS-friction due to a natural river bed ($k_s = 40.0$). The operating point is given as for low flow and very high flow conditions.
- a built channel (typical inflow to a hydropower station) with constant cross section and with smooth bottom and sides ($k_s = 70.7$). The outflow fits the typical operating range of low head hydro turbines.

The reference 'plant' data are set to

Q_r	50	m^3/s
L_r	20	m
B_r	10	m
S_r	-1.0	m
H_r	1.5	m
D_r	2.5	m
U_r	2.0	m/s
k_{s_r}	40.0	-
or	70.7	
U_{F_r}	5.0	m/s
F_r	0.40	
Q_{F_r}	125	m^3/s

and the model reference parameters to

$T_1 = T_3 = T_0$	10	s
T_2	1.6	s
ω_e	$\sqrt{2}/4$	rad/s
T_e	$\sqrt{2}\pi 4 = 17.80$	s
T_E	$4L_r/U_{Frr} = 16$	s
L_{tot}	40	m
T_{fill}	$L_{tot}/U_r = 20$	s

The 'reference' values v_r (or v_r in the `.sce`-scripts) are what the plant is designed for. They are often called 'nominal' or 'design' values. The 'operating' values at the actual steady state operating conditions of the plant are indicated by v_0, and are given proportionally to the 'reference' values.

2.6.1 River Section

The following three cases are investigated[4]

- flow operating points at $Q\_0 = Q_r * [0.25, 1.0, 4.0, 16.0]$, where jointly $B\_0 = B_r * [1.0, 1.0, 2.0, 4.0]$ to model additional lateral overflow areas for high flow conditions;
- Froude number operating points $F\_0 = F_r * [0.5, 1.0, 2.0, 4.0] = [0.2, 0.4, 0.8, 1.6]$, where $Q\_0 = Q_r$ and $k_s\_0 = k_{s_r}$ are both fixed. This will lead to different D_0 and different bottom slopes;
- conical cross sections with $Q\_0 = Q_r$ and $k_s\_0 = k_{s_r}$ both fixed. And at the reference cross section is set at location $i = 2$, that is $B\_2 := B_r$ and $S\_2 := S_r$. Also the adjacent section downstream is set to *continued* cross section, that is $B_{i+2} := B_{i+1} + 0.5 \cdot (B_{i+1} - B_{i-1})$ and $S_{i+2} := S_{i+1} + 0.5 \cdot (S_{i+1} - S_{i-1})$ where $i = 2$.
  - Side walls convergent from $B_{i-1} = 1.2 * B_r$ to $B_{i+1} = 0.8 * B_r$ and side walls divergent from $B_{i-1} = 0.8 * B_r$ to $B_{i+1} = 1.2 * B_r$. This results in an opening angle of the diffusor wall of $0.1 rad$ that is 6^o. It also fits the results of the previous section on allowable conicity, as $\Delta B_i / B_i = 0.5 * 4.0m / 10m = 0.20$

[4]Note that not all parameter sets will be documented by plots but only a selection to conserve space. However the parameter sets are made available in the `context.sce`-files for 'de-commenting'

– Bottom sloping up from $S_{i-1} = 1.5 * S_r = -1.5m$ to $S_{i+1} = 0.5 * S_r = -0.5m$ and sloping down from $S_{i-1} = 0.5 * S_r = -0.5m$ to $S_{i+1} = 1.5 * S_r = -1.5m$. . Concerning allowable conicity this is $\Delta S_i/D_r = 0.5 * 1.0m/2.5m = 0.20$. And this will produce a slope of $\pm 1m$ by $20m$, that is $\pm 0.05rad$ or $\pm 3^o$
The second case will produce a strongly accelerated flow, and transition to supercritical flow will occur (and thus will require $\varkappa := 1.0$).

– and by combining the convergent & upslope case (to the 'confusor' case) and the divergent & downslope case (to the 'diffusor' case).

For all three cases the *disturbance* shall be a rectangular variation $\pm\Delta Q(t)$ around Q_0 with long enough period to show at least four periods of the sloshing oscillation.

2.6.2 Channel section

The operating point on flow is set to $Q\_0 = Q_r * [0.30, 1.00, 1.25]$. The following variables are 'frozen'

- $D\_0 := D_r$; and thus F_0 varying with Q_0,
- $k_s = 70.7$,
- $B\_0 := B_r; S\_0 := S_r$, and thus constant cross section at reference size.

To model a typical turbine inflow channel to a power station, a small rectangular outflow variation $\pm\Delta Q$ on Q_0 is applied with inflow fixed at Q_0;
Remark: A large flow disturbance such as produced by a turbine trip or a startup and quick loading would generate strong level excursions. Then a flow bypass such as a broad overfall or a fast moving weir is required to keep the water level within its usual operating constraints.

2.6.3 Boundary Conditions

At the Inflow Cross Section, index $_{in}$

case A:
If the upstream inflow is given by the GMS-law, then for the river layout:
Given: $\mathbf{k_{s_r}}$, Q_{in}, $B_{in} := B_r$, $S_{in} := S_r$.
Find: D_{in} (and U_{in})
where the bottom slope is determined from the *reference* operating point and is kept fixed for the other operating points, and where D_{in} must be calculated from the GMS-law by iteration:

```
// s_c2_03_00_iter.sce
// iteration for D_I; Glf 24.3.14
// reference slope

k_s = 40.;
Q_r = 50.0; B_r = 10.0; D_r =  2.5; U_r = 2.0;
R_r = (B_r*D_r)/(B_r + 2*D_r);
I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// flow set; choose by 'comment'
//Q_0 = 800.0; B_0 = 40.0;
//Q_0 = 200.0; B_0 = 20.0;
//Q_0 =  50.0; B_0 = 10.0;
  Q_0 =  12.5; B_0 = 10.0;

// initialzing 'iteration loop'
D_I = D_r; gainQ = 0.01*(D_I/Q_0);
Q = k_s*(B_0)*(D_I)*(D_I)^(2/3)*(I_r)^(1/2);
n = 1; // loop counter
eQ  = -Q_0 + Q;
// iteration loop
while (eQ<-0.001)|(eQ>0.001)  then
    R_I = (B_0*D_I)/(B_0 + 2*D_I);
    Q   = (k_s*(I_r)^(1/2)*B_0)*D_I*(R_I)^(2/3);
    eQ  = -Q_0 + Q;
    D_I = D_I - gainQ*eQ;
    n = n + 1;
end
// results
//Q_0 = 800.; B_0 = 40; -> D_I = 5.371; n = 1901;
//Q_0 = 200.; B_0 = 20; -> D_I = 3.649; n = 1126;
//Q_0 =  50.; B_0 = 10; -> D_I = 2.500; n =    2;
//Q_0 = 12.5; B_0 = 10; -> D_I = 0.995; n =  253;
```

resulting in D_{in}, and from there $U_{in} = Q_{in}/(B_r \cdot D_{in})$

case B:
If the main flow parameter F (Froude number) is given, then for both the river and the channel layout:
Given: $\mathbf{F_{in}}$, Q_{in}, $B_{in} := B_r$, $S_{in} := S_r$,.
Find: D_{in} (and U_{in})

$$\begin{aligned} U_{in} &= F_{in} \cdot \sqrt{g \cdot D_{in}} = \frac{Q_{in}}{B_{in} \cdot D_{in}} \\ Q_{in} &= D_{in}^{3/2} \cdot B_{in} \cdot F_{in} \cdot \sqrt{g} \\ \rightarrow \quad D_{in} &= \left[\frac{Q_{in}}{B_{in} \cdot F_{in} \cdot \sqrt{g}}\right]^{2/3} \\ \text{and} \quad U_{in} &= [F_{in} \cdot \sqrt{g}]^{2/3} \cdot \left[\frac{Q_{in}}{B_{in}}\right]^{1/3} \end{aligned}$$

Friction Slope, index $_f$

For both the river flow and the channel flow along L at all i, here for $i = 2$
Given: $Q_i := Q_{in}$, $B_i := B_{in}$, $D_i := D_{in}$.
Find: ΔS_{f_i}

$$\begin{aligned} \text{From} \quad U_i &= k_s \cdot R_i^{2/3} \cdot \left[\frac{\Delta S_{f_i}}{L_r}\right]^{1/2} \quad \text{with} \quad R_i = \frac{B_i \cdot D_i}{B_i + 2D_i} \\ \rightarrow \quad \Delta S_{f_i} &= \frac{L_r}{k_s^2} \cdot \left[\frac{Q_i}{B_i} \cdot D_i \cdot R_i^{2/3}\right]^2 \end{aligned}$$

and for all i the bottom profile is constructed step-by-step:

$$\begin{aligned} S_{f_i} &= S_{f_{i-1}} + 0.5 \cdot \Delta S_{f_i} \\ S_{f_{i+1}} &= S_{f_i} + 0.5 \cdot \Delta S_{f_i} \\ \text{and for } i = 2 \text{ starting with } S_{f_{i-1}} &= 0 \\ S_{i-1}|_{total} &= S_{i-1}|_{nofriction} - 0 \\ S_i|_{total} &= S_i|_{nofriction} - S_{f_i} \\ S_{i+1}|_{total} &= S_{i+1}|_{nofriction} - S_{f_{i+1}} \end{aligned}$$

where $S_i|_{nofriction}$ is the 'presumed geometry' input for the bottom (without adding the additional slope necessary to compensate the friction loss).

At the Outflow Cross Section, index $_{ot}$

For the *river model*, the outflow $Q_{ot}(t)$ [5] from the last volume balance compartment at $_{i+1}$ is determined by its context variable $D_{ot}(t) := D_{i+1}$ and the GMS-law of the next downstream river section at $_{i+2}$.
Given: $D_{ot}(t)$, B_{ot}
Find: $Q_{ot}(t)$

$$\begin{aligned} Q_{ot}(t) &= B_{ot} \cdot D_{ot}(t) \cdot U_{ot}(t) \quad \text{with} \quad U_{ot}(t) = k_s \cdot R_{ot}^{2/3} \cdot \left[\frac{\Delta S_{ot}}{L_r}\right]^{1/2} \\ &= k_s \cdot (B_{ot} \cdot D_{ot}(t)) \cdot \left[\frac{B_{ot} \cdot D_{ot}(t)}{B_{ot} + 2D_{ot}(t)}\right]^{2/3} \cdot \left[\frac{\Delta S_{ot}}{L_r}\right]^{1/2} \end{aligned}$$

[5] In the `.sce`-files the $_{ot}$ is replaced by $\_o$

And the 'extended' slope of the river bed into the next downstream river element $_{i+2}$ is

$$\begin{aligned} \Delta S_{f_{i+2}} &:= \Delta S_{f_{ot}} \approx \Delta S_{f_{i+1}} \\ S_{f_{i+2}} &:= S_{f_{ot}} \approx S_{f_{i+1}} - 0.5 \cdot \Delta S_{f_{i+1}} \\ S_{i+2}|_{total} &:= S_{i+2}|_{nofriction} - S_{f_{ot}} \end{aligned}$$

For the *channel model*, the outflow from the last volume compartment is 'forced' to the rectangular sequence around Q_{in}, and the inflow $Q_{in} = Q\_0$ is constant.

2.6.4 Initial Conditions

Volume balances: for all i let $D\_0i := D_{in}$.

> Note that this is correct if and only if the bottom slope and the level slope are parallel. Otherwise this simple assignment will not be equivalent to steady-state initial conditions. It will lead to an initial transient response, which must settle before the subsequent experiments of flow disturbance may be performed.

Momentum balances: for all i let $Q\_0i := Q_{in}$

2.7 Implementation in 'scilab/xcos'

WARNING: Using Scilab v. 5.4.1. throughout is strongly recommended.
Newer versions (such as 5.5.2 or 6.0.0) produce crashes due to unresolved incompatibilities.

The aim of this section is to give an overview on

- how the plant layouts and associated operating conditions are implemented,
- how the signal flow diagrams are supplied with parameters and the results are plotted,
- and how the simulation runs are handled.

2.7.1 Naming of the plant layout cases

As this chapter covers the basic system models with essentially three ordinary differential equations, the names of all software elements within this context start with `03_`

Then each of the plant layouts given in the last section is named as follows

`s_c2_03_01`	river bed, constant cross section, low and very high flow levels, inflow variation
`s_c2_03_02`	river bed, constant cross section, low and high Froude number levels, inflow variation
`s_c2_03_03`	river bed, conical cross sections, inflow variation
`s_c2_03_04`	channel flow, constant depth and width, fixed inflow levels, small outflow variations

2.7.2 The 'Triplet Files' for each layout case

Each layout case is covered by a triplet of files, the original `.zcos`, and two `.sce`-scripts:

- The `s_c2_03_0k.xcos` diagram (k = 1 ... 4) results from the 'click and drag' build-up using the Xcos 'palettes'. It contains the signal flow graph around the integrators, thus implementing the

differential equations. Several Xcos blocks 'mathematical expressions' make the signal flow graph more compact. The blocks `To workspace` put the simulation run data into the `scilab`-workspace for post-processing, i.e. plotting. The actual parameter values for the diagram are passed to the graph through the `'context'`-feature.

- The `s_c2_03_0k.zcos` diagram is its 'zipped' companion, which is called by the batch-simulation function `scicos_simulate(...)`, see below.
- A first `scilab`-script named `s_c2_03_0k_context.sce` defines and evaluates all parameters for the `s_c2_03_0k.zcos` file which are needed for the simulation at the specified operating point. Specific operating point variable values v_0 are selected by 'comment'-ing/'de-comment'-ing (starting lines with twin backlashes). At the end of the texttt_context-script, the data transfer into the `scilab`-workspace is set up.
- The second `scilab`-script named `s_c2_03_0k_crunplot.sce` lets the simulation run in batch-mode by calling `scicos_simulate(...)` and plots using standard `scilab`-plot routines.

2.7.3 Handling a simulation run

The group of the above triplet files must be placed adjacent in the same Explorer folder.

After having started `scilab`, the scilab console window opens. Navigate in the file browser (can be found under 'applications' on the console menu bar of the `scilab-Console` window to the appropriate (sub-)directory. Then

1. From the file browser within the `scilab`-console window open all files of your target case triplet by double clicks.
   Both `.sce` scripts will open in the `sce-editor`-window. The `.zcos`-graph will open in the `.zcos-window` ('loading...' may take several seconds).
2. Edit the data in the `_context.sce`-file to fit the specific operating point (using the 'comment'-ing technique), and (if required) also in the `_crunplot.sce`-file. Note that appropriate scaling on the plots for the specific operating point may be necessary and **'save'** both the `_context.sce`-file and the `_crunplot.sce`-file by way of the menu bar of the `sce-editor`.
3. Go to the `.zcos`-window (the zipped 'diagram'), and edit the structure (if required). Then 'save' the diagram by the zcos menu bar, with the appendix `_.zcos`. Note that this may take 10 to 30 seconds for larger diagrams.
   This zipped form is called from the batch simulation function.
   If the diagram is NOT changed, and only new parameter values are inserted, then this 'save' and its subsequent 'save as' are not required. This saves turnaround-time !
4. go back to the `sce-editor`-window and to the `_crunplot.sce`-file: 'save and execute' this file from within the `sce-editor` menu bar. In batch-mode (that is without further user interaction) this
   - executes the `_context.sce`-file to generate the parameter values for the diagram,
   - loads the parameter values into the `.zcos`-diagram,
   - runs the simulation using the library routine `scicos_simulate` [6],
   - extracts the data to be plotted from the data arrays written by the 'to workspace' blocks and
   - plots the results using standard scilab-plot routines. Again this may take several tens of seconds for large diagrams...

Hints: Storing the resulting plot files as .pdf's and not as .png's yields better resolution on both screen and print outputs for a report in LaTeX. – If the report is prepared in MS Word, then only .png's can be inserted (as 'picture from file').
The best way to include the 'diagrams' in both LaTeXand MS-Word is to 'export' them from the `.zcos`-window within Xcos to .png's.

[6] The alternative `xcos_simulate` which will not be supported in newer releases of `scilab`

2.8 Simulation results

2.8.1 case s_c2_03_01_ river bed, constant width, low/high flow

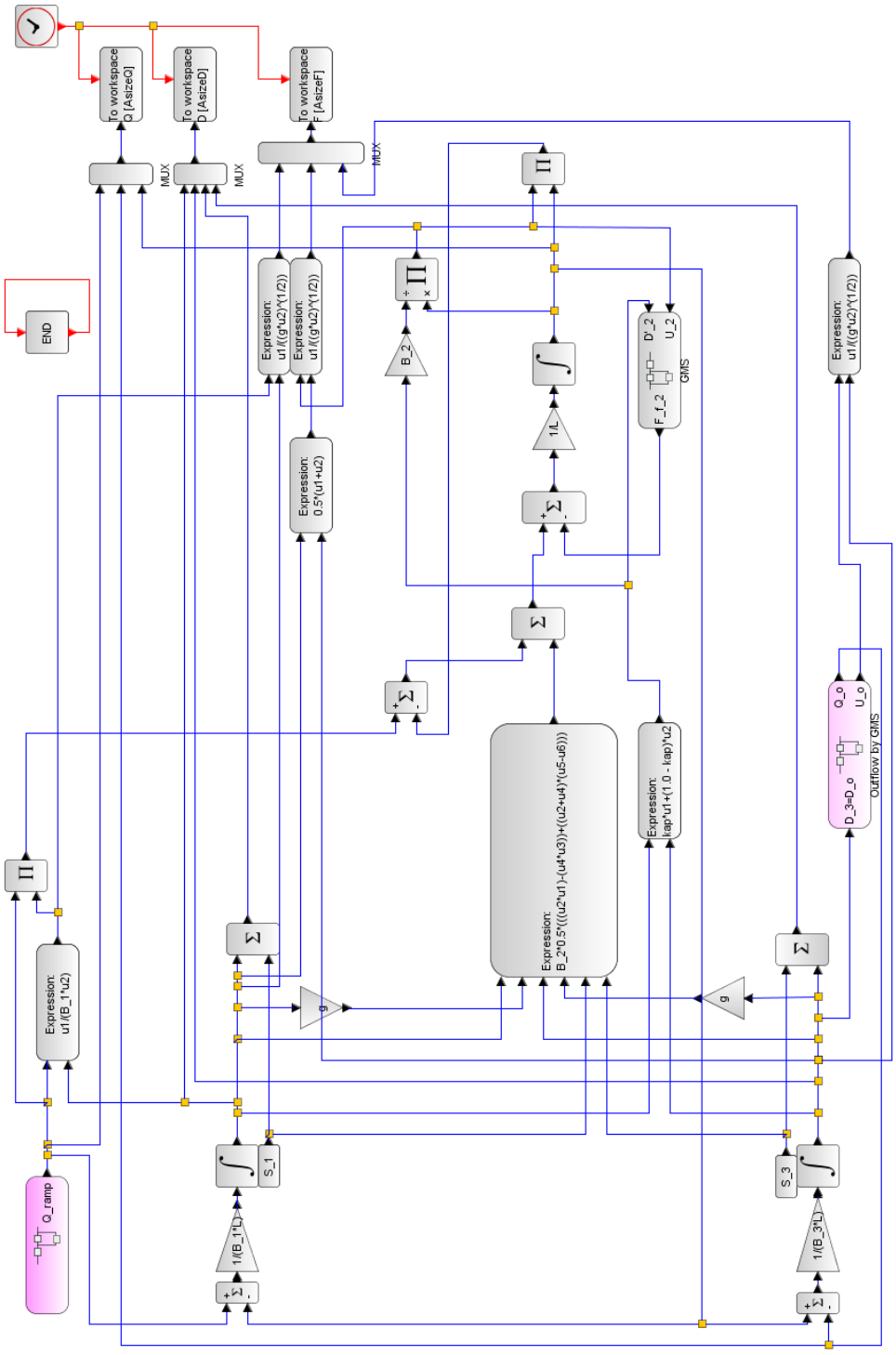


Figure 2.6: *s_c2_03_01*, zcos-Diagram

.sce-files

```
// s_c2_03_01_context
// Glf  24.03.2014
// Low/High Flow

g = 10.;
L = 20.;

// flow set; choose by 'comment'
   Q_0 = 800.0; B_0 = 40.0; S_0 = 1*(-1.0);
// Q_0 =  50.0; B_0 = 10.0; S_0 = 1*(-1.0);
// Q_0 =  12.5; B_0 = 10.0; S_0 = 1*(-1.0);

// reference bottom slope
// friction coeff
//k_s =  31.6;
  k_s =  40.0;
  Q_r = 50.0; B_r = 10.0;
  D_r =  2.5; S_r = 1*(-1.0);
  U_r =  2.0;
  R_r = (B_r*D_r)/(B_r + 2*D_r);
  I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// initialzing 'iteration loop'
D_0 = D_r;
gainQ = 0.01*(D_0/Q_0);
Q = k_s*(B_0)*(D_0)*(D_0)^(2/3)*(I_r)^(1/2);
eQ  = -Q_0 + Q;

// iteration loop
while (eQ<-0.001)|(eQ>0.001)  then
    R_0 = (B_0*D_0)/(B_0 + 2*D_0);
    Q   = (k_s*(I_r)^(1/2)*B_0)*D_0*(R_0)^(2/3);
    eQ  = -Q_0 + Q;
    D_0 = D_0 - gainQ*eQ;
end

  kap = 1.00;

// channel shape:
  B_1 = 1*B_0;
  B_3 = 1*B_0;
  B_2 = 0.5*(B_1 + B_3);
  B_4 = B_3;

  S_1 = 1.0*S_0;
  S_3 = 1.0*S_0;
  S_2 = 0.5*(S_1 + S_3);
  S_4 = S_3;

//Bottom shape at _r conditions
  delSf = + L*I_r;

  S_f_1 = 0;
  S_f_2 = S_f_1 + 0.5*delSf;
  S_f_3 = S_f_2 + 0.5*delSf;
  S_f_4 = S_f_3 + 0.5*delSf;

  S_1 = S_1 - S_f_1 ;
  S_2 = S_2 - S_f_2 ;
  S_3 = S_3 - S_f_3 ;
  S_4 = S_4 - S_f_4 ;

// outflow following GMS
 g_4 = k_s*((I_r)^(1/2));

// initial conditions
  D_10 = D_0; D_30 = D_0;
  Q_20 = Q_0;

// inflow sequence
  t_st_1 = 50.0;   r_1_0 = Q_0;  r_1_1 = (1.5)*Q_0;
  t_st_2 = 200.0;  r_2_0 = 0.;   r_2_1 = -1.00*Q_0;
  t_st_3 = 350.0;  r_3_0 = 0.;   r_3_1 = +0.50*Q_0;

  T_fin = 500.;

  g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
  tau_st = 1.0*(Q_r/Q_0); Q_st_0 = Q_0;

// Datatransfer to Plots
CCQ = 4;      // number of channels Q + 1 for time
CCD = 5;      // number for D,H
CCF = 4;      // number for F
CN = 1000;     // number of clockticks until Tfin
delT = T_fin/CN;  // read-interval for clock ticks
AsizeQ = 1.1*CCQ*CN; // size of Dataarrays for Q
AsizeD = 1.1*CCD*CN;
AsizeF = 1.1*CCF*CN;
```

```
// s_c2_03_01_crunplot
// Glf,   18.03.14  26.10.2015
// Low/High Flow

stacksize('max'); exec('s_c2_03_01_context.sce', -1);
importXcosDiagram('s_c2_03_01.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************


vcolorQ = [1, 2, 5];
vcolorD = [1, 2, 5, 13];
vcolorF = [1, 2, 13];

f1 = scf(1); clf();
f1.auto_resize = "on";
f1.figure_size = [626.,420.];
   plot2d(Q.time,Q.values,vcolorQ,rect=[0,0,500,1400]);
// plot2d(Q.time,Q.values,vcolorQ,rect=[0,0.,500,80.]);
// plot2d(Q.time,Q.values,vcolorQ,rect=[0,0.,500,20.]);

xtitle("Q_in, Q_out, Q_2");
xgrid(1); h1 = legend(['Q_in';'Q_out';'Q_2'],4);

f2 = scf(2); clf();
f2.auto_resize = "on";
f2.figure_size = [626.,420.];
   plot2d(D.time,D.values,vcolorD,rect=[0,-3.0,500,8]);
// plot2d(D.time,D.values,vcolorD,rect=[0,-3.0,500,4]);
// plot2d(D.time,D.values,vcolorD,rect=[0,-3.0,500,2]);
xtitle("D_1 and D_3, H_1 and H_3");
xgrid(1);h1 = legend(['D_1';'D_3';'H_1';'H_3'],4);

f3 = scf(3); clf();
f3.auto_resize = "on";
f3.figure_size = [626.,420.];
plot2d(F.time,F.values,vcolorF,rect=[0.,0.,500,1.0]);
xtitle("F_1, F_2, F_3");
xgrid(1);h1 = legend(['F_1';'F_2'; 'F_3'],4);
```

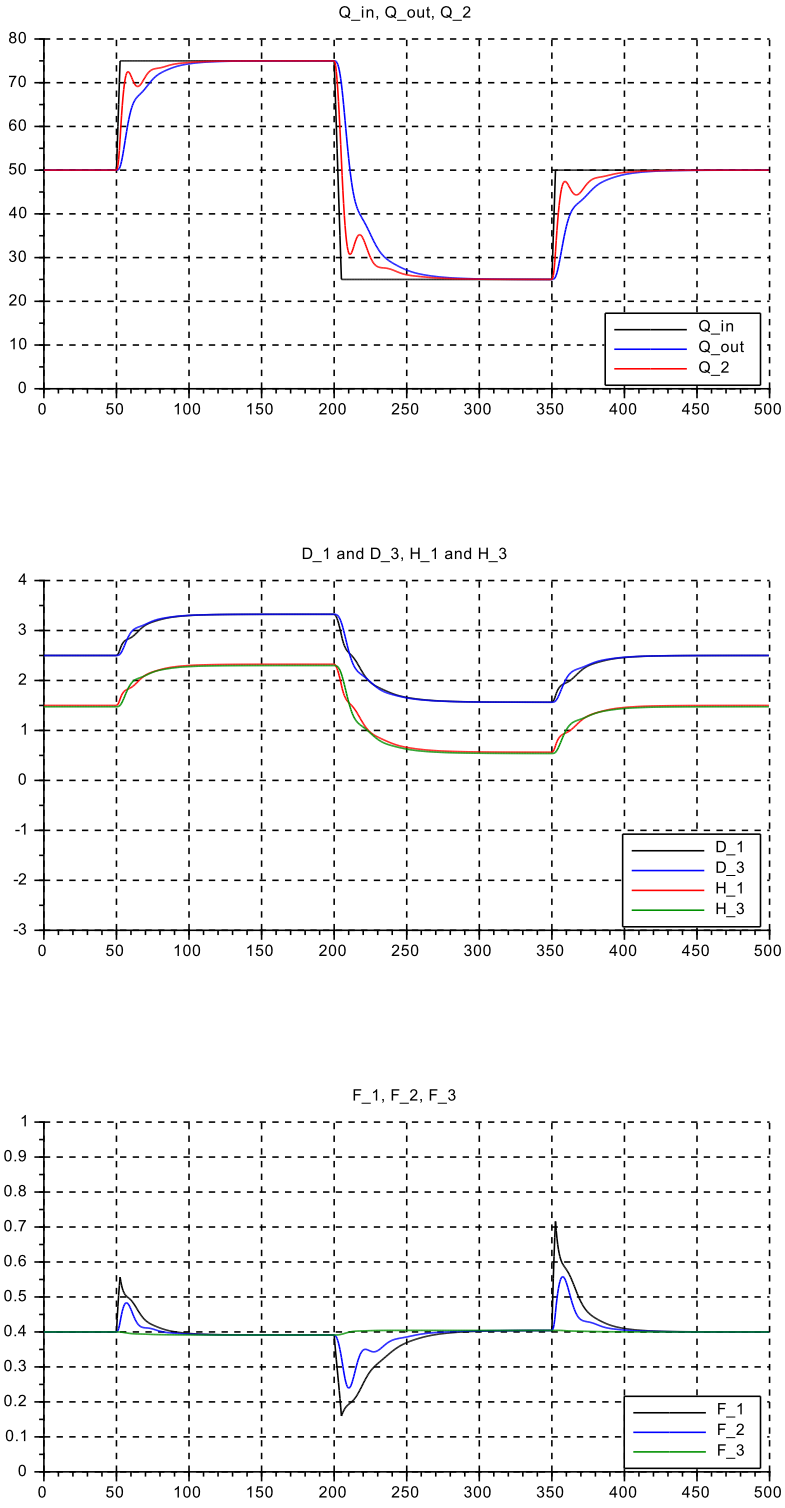


Figure 2.7: $s\_c2\_03\_01\_$ for reference flow $Q\_0 = 1.0 \cdot Q\_r$

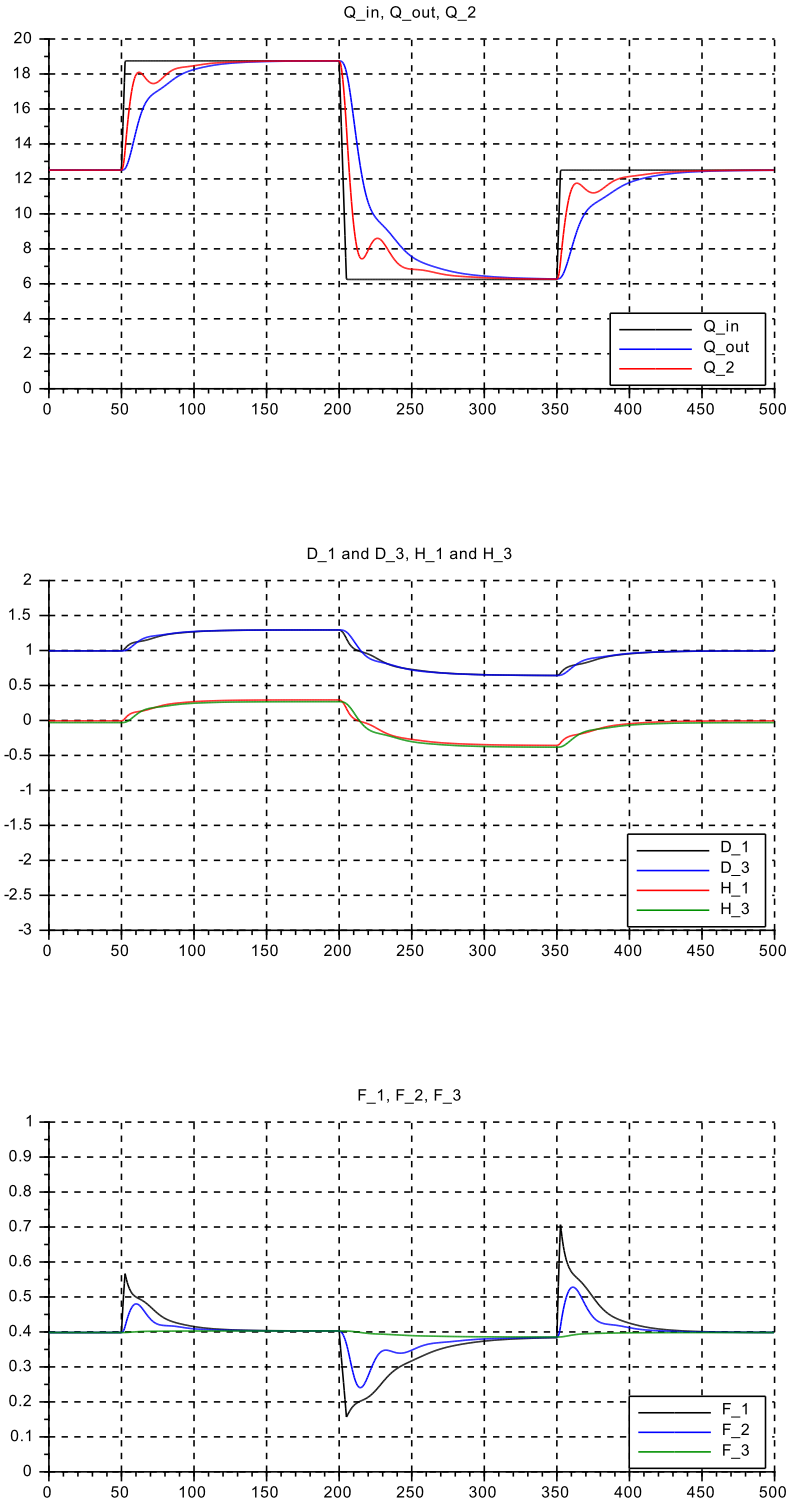


Figure 2.8: $s\_c2\_03\_01\_$ for very low flow $Q\_0 = 0.25 \cdot Q\_r$

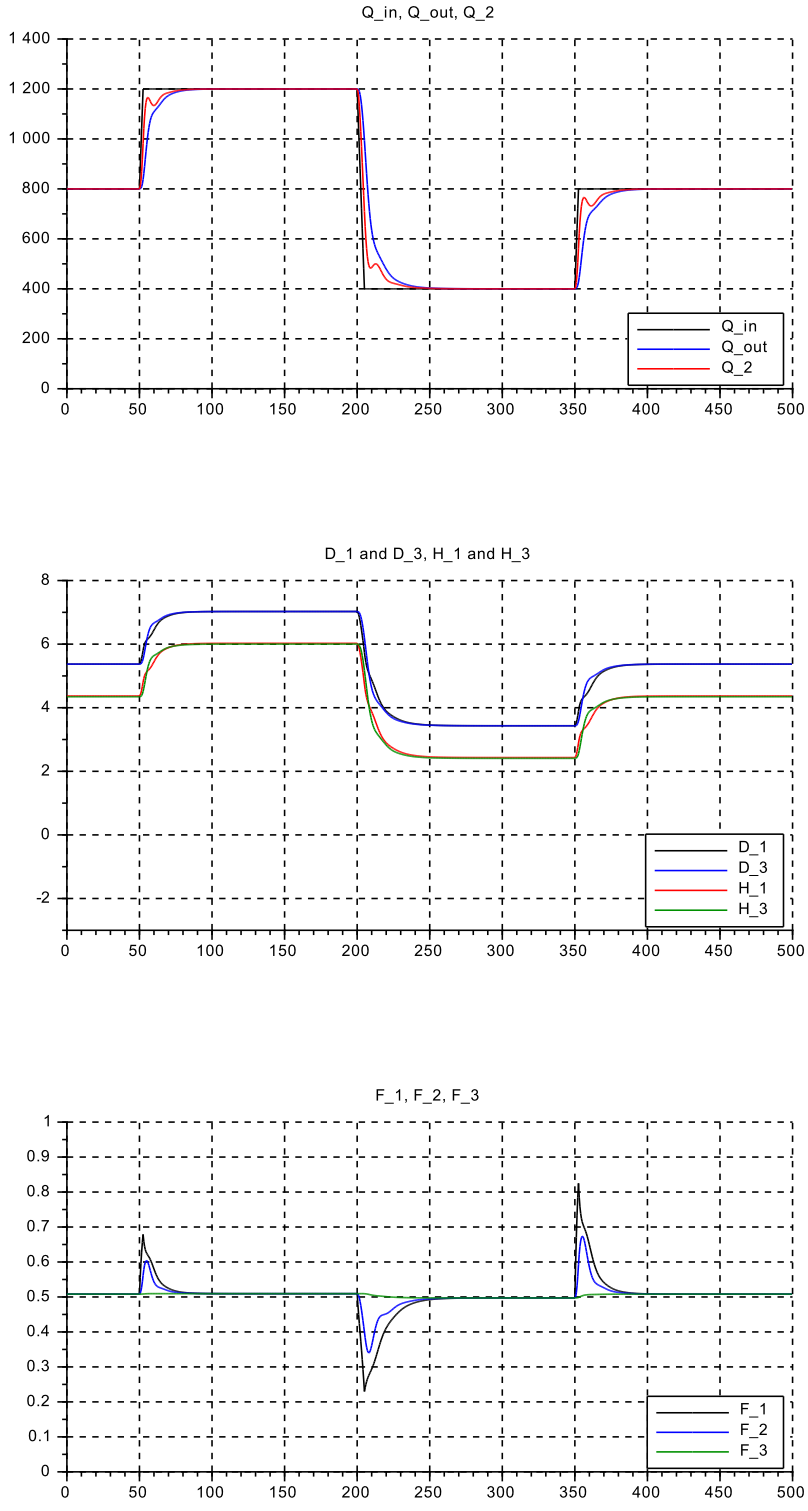


Figure 2.9: $s\_c2\_03\_01\_$ for very high flow, $Q\_0 = 16.0 \cdot Q\_r$ with $B\_0 = 4 \cdot B\_r$

2.8.2 case s_c2_03_02_ river bed, constant width, reference flow, but different Froude numbers

.sce-files

```
// s_c2_03_02_context
// Glf  18.03.2014, 26.10.2015


g = 10.;
L = 20.;
Q_r = 50.0;
D_r = 2.5;
S_r = 1*(-1.0);
B_r = 10.;
U_r = 2.0;

// Operating point, by Froude-number
//F_0 = 0.2;
//F_0 = 0.8;
  F_0 = 1.6;

  Q_0 = Q_r; B_0 = B_r;


// friction coeff
//k_s = 100.;
//k_s =  70.7;
  k_s =  40.0;
//k_s =  31.6;

  kap = 1.0;

// channel shape:
  B_1 = 1*B_r;
  B_3 = 1*B_r;
  B_2 = 0.5*(B_1 + B_3);
  B_4 = B_3;

  S_1 = 1.0*S_r;
  S_3 = 1.0*S_r;
   S_2 = 0.5*(S_1 + S_3);
   S_4 = S_3;

// inflow boundary conditions
   // Given: Q_0 and F_0; determine D_0:
D_0 = ((1/g)*((Q_0/(B_0*F_0))^2))^(1/3);

// friction slope
Rtilda_0 = ((B_0*D_0)/(B_0+2*D_0))^(2/3);
I_0 = (Q_0/(B_0*D_0*k_s*Rtilda_0))^2;

delSf = + L*I_0;

S_f_1 = 0;
S_f_2 = S_f_1 + 0.5*delSf;
S_f_3 = S_f_2 + 0.5*delSf;
S_f_4 = S_f_3 + 0.5*delSf;

S_1 = S_1 - S_f_1 ;
S_2 = S_2 - S_f_2 ;
S_3 = S_3 - S_f_3 ;
S_4 = S_4 - S_f_4 ;

// outflow following GMS
 g_4 = k_s*((I_0)^(1/2));

// initial conditions
D_10 = D_0;
D_30 = D_0;
Q_20 = Q_0;

// inflow sequence
t_st_1 = 50.0;   r_1_0 = Q_0; r_1_1 = (1.0+0.5)*Q_0;
t_st_2 = 200.0;  r_2_0 = 0.;  r_2_1 = -1.0*Q_0;
t_st_3 = 450.0;  r_3_0 = 0.;  r_3_1 = +0.50*Q_0;

T_fin = 600.;

g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st = 1.0; Q_st_0 = Q_0;


// Datatransfer to Plots
CCQ = 4;      // number of channels Q + 1 for time
CCD = 5;      // number for D,H
CCF = 4;      // number for FroudeZ
CN = 1200;     // number of clockticks until Tfin
delT = T_fin/CN;  // read-interval for clock ticks
AsizeQ = 1.1*CCQ*CN; // size of Dataarrays for Q
AsizeD = 1.1*CCD*CN;
AsizeF = 1.1*CCF*CN;
```

```
// s_c2_03_02_crunplot
// Glf  18.03.14 26.10.2015

stacksize('max'); exec('s_c2_03_02_context.sce', -1);
importXcosDiagram('s_c2_03_02.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

vcolorQ = [1, 2, 5];
vcolorD = [1, 2, 5, 13];
vcolorF = [1, 2, 13];

f1 = scf(1); clf();
f1.auto_resize = "on";
f1.figure_size = [626.,420.];
plot2d(Q.time,Q.values,vcolorQ,rect=[0.,0.,600,100.]);
xtitle("Q_in, Q_out, Q_2");
xgrid(1); h1 = legend(['Q_in';'Q_out';'Q_2'],4);

f2 = scf(2); clf();
f2.auto_resize = "on";
f2.figure_size = [626.,420.];
plot2d(D.time,D.values,vcolorD,rect=[0.,-3.0,600,8.0]);
xtitle("D_1 and D_3, H_1 and H_3");
xgrid(1);h1 = legend(['D_1';'D_3';'H_1';'H_3'],4);

f3 = scf(3); clf();
f3.auto_resize = "on";
f3.figure_size = [626.,420.];
// plot2d(F.time,F.values,vcolorF,rect=[0.,0.,600,0.4]);
// plot2d(F.time,F.values,vcolorF,rect=[0.,0.,600,1.2]);
   plot2d(F.time,F.values,vcolorF,rect=[0.,0.,600,2.5]);
xtitle("F_1, F_2, F_3");
xgrid(1);h1 = legend(['F_1';'F_2'; 'F_3'],4);
```

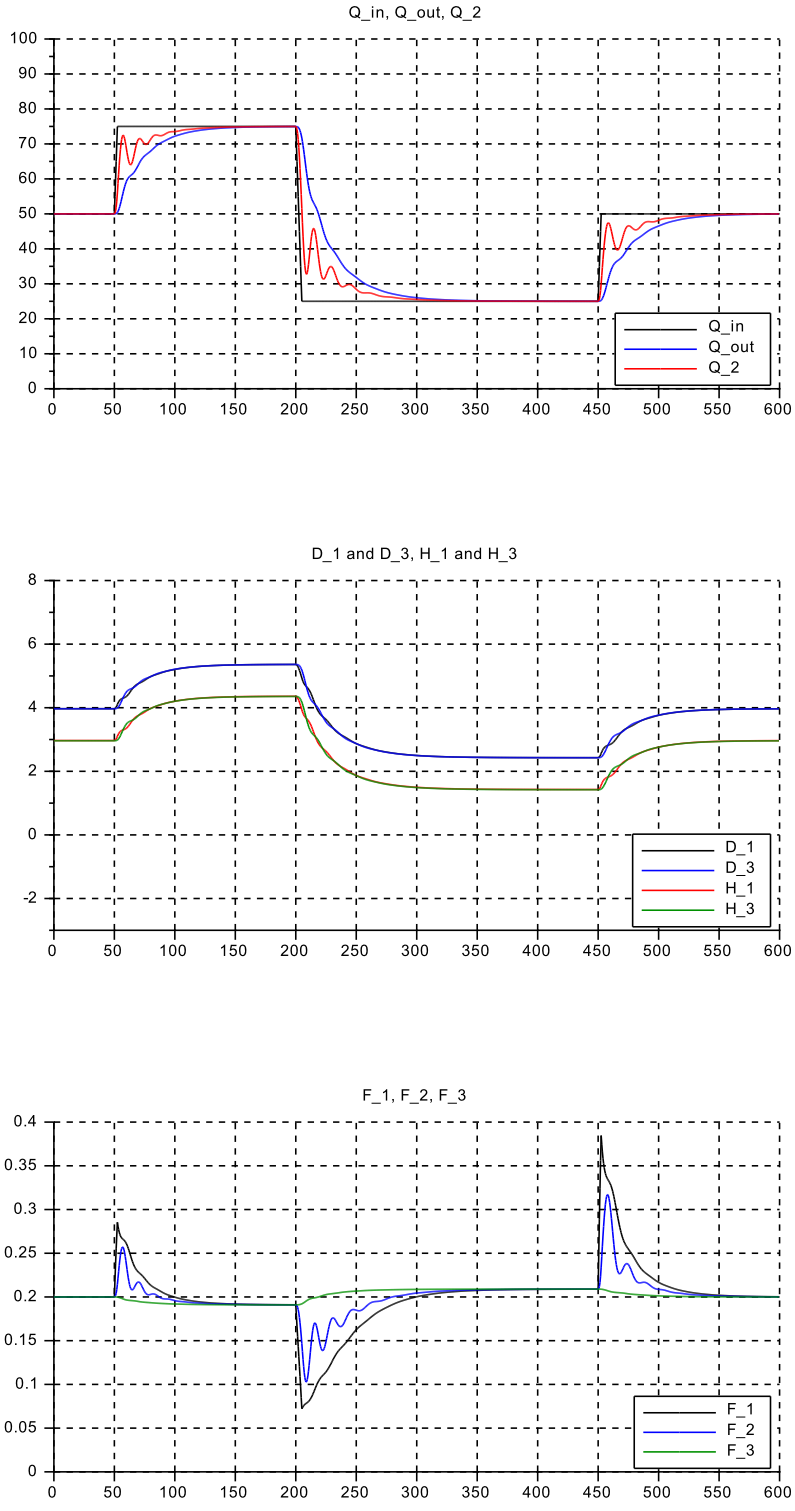


Figure 2.10: $s\_c2\_03\_02\_$ for low Froude number $F = 0.20$

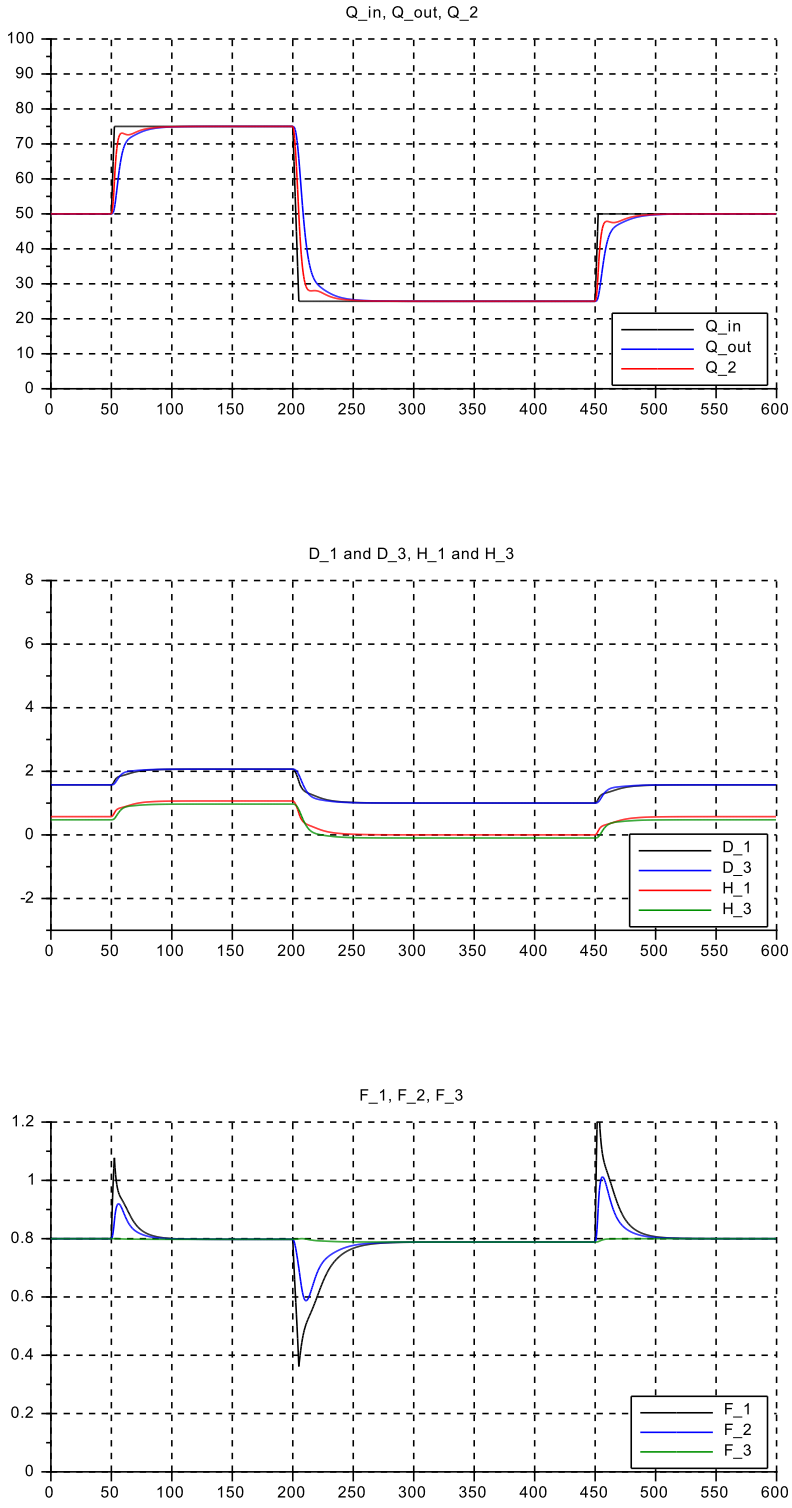


Figure 2.11: $s\_c2\_03\_02\_$ for high, but subcritical Froude number $F = 0.80$

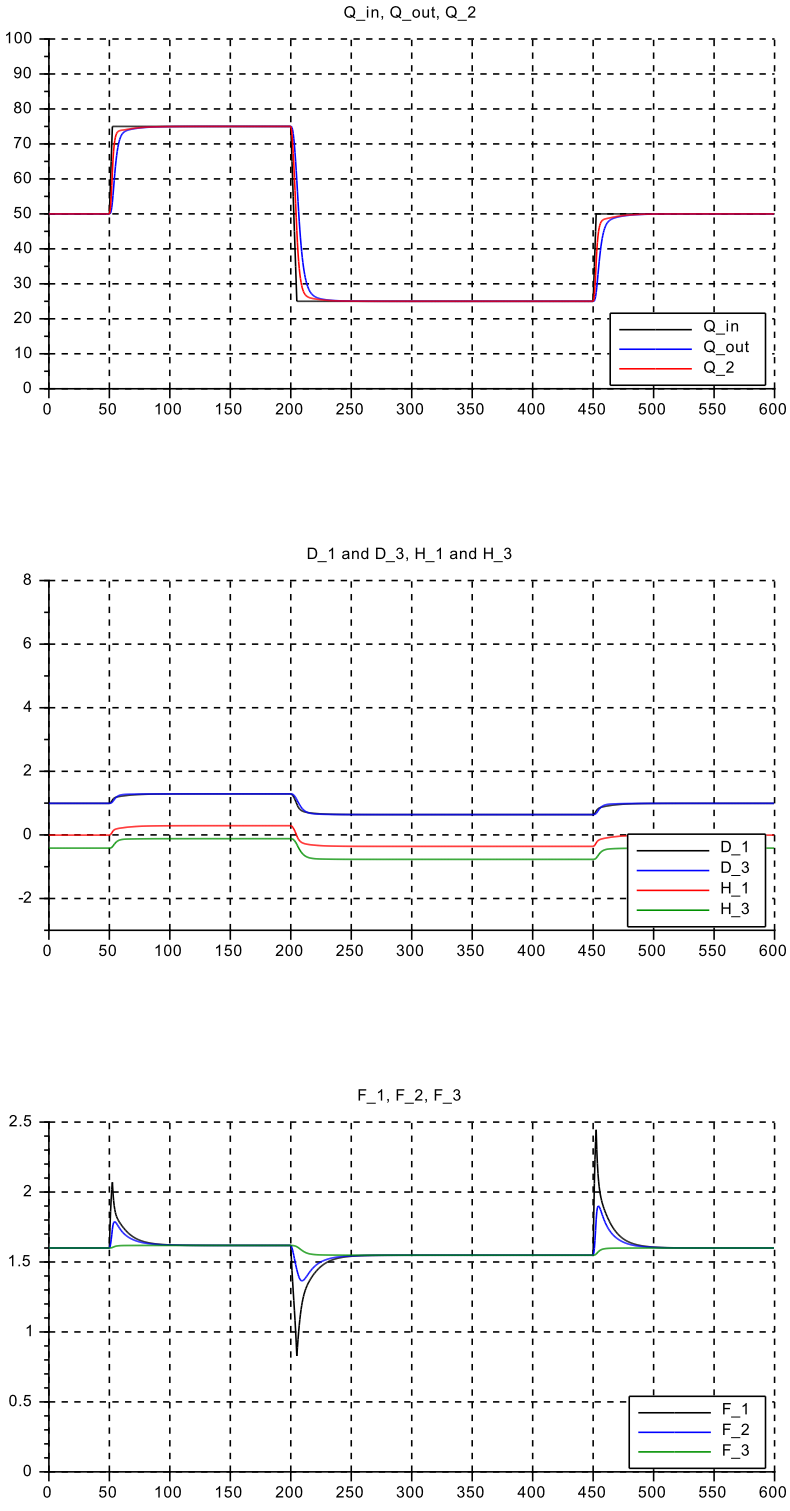


Figure 2.12: $s\_c2\_03\_02\_$ for supercritical Froude number, $F = 1.60$

2.8.3 case s_c2_03_03_ river bed, conical cross section

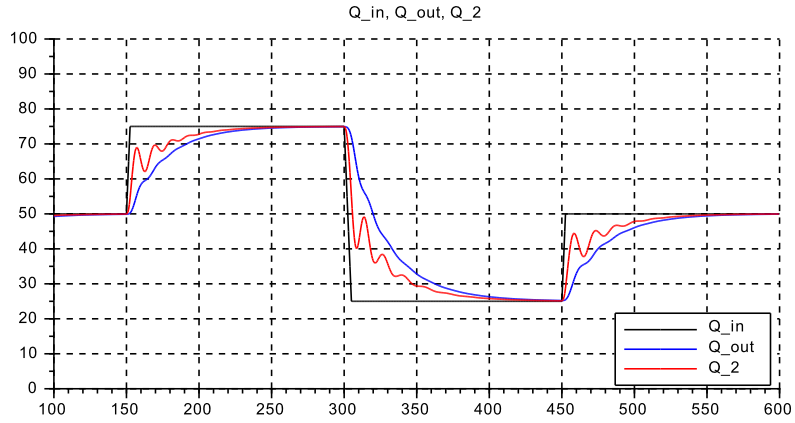
.sce-files

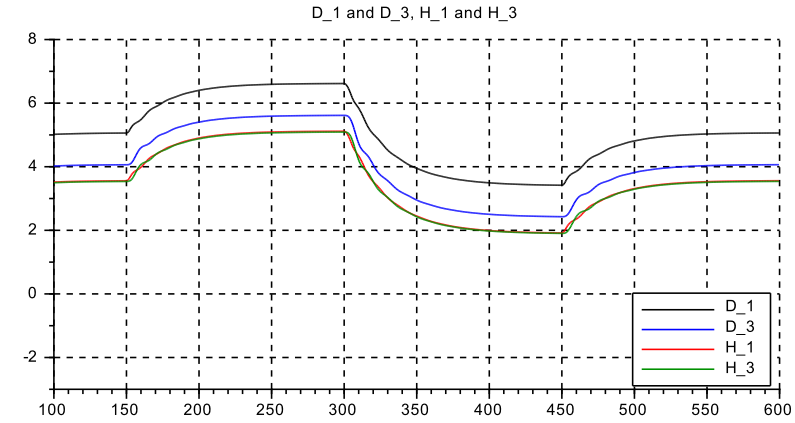
```
// s_c2_03_03_context
// Glf  18.03.2014 26.10.2015


g = 10.;
L = 20.;
Q_r = 50.0;
D_r = 2.5;
S_r = 1*(-1.0);
B_r = 10.;
U_r = 2.0;

// Operating point, by Froude-number
//F_r = 0.2;
  F_r = 0.4;
//F_r = 0.8;
//F_r = 1.6;
//F_r = 2.4;


// friction coeff
// k_s = 100.;
// k_s =  70.7;
   k_s =  40.;
// k_s =  31.6;

  kap = 1.0;

// channel shape:
// confusor:
// B_1=1.2*B_r; B_3=0.8*B_r; S_1=1.5*S_r; S_3=0.5*S_r;

   //diffusor:
  B_1= 0.8*B_r; B_3=1.2*B_r; S_1=0.5*S_r; S_3=1.5*S_r;

   B_2 = 0.5*(B_1 + B_3);
   B_4 = B_3 + 0.5*(B_3 - B_1);
   S_2 = 0.5*(S_1 + S_3);
   S_4 = S_3 + 0.5*(S_3 - S_1);

// reference depth
   // Given: Q_r and Fr_r; determine D_r:
D_r = ((1/g)*((Q_r/(B_r*F_r))^2))^(1/3);
```

```
// bottom shape for _r conditions
Rtilda_r = ((B_r*D_r)/(B_r+2*D_r))^(2/3);
I_r = (Q_r/(B_r*D_r*k_s*Rtilda_r))^2;
delSf = + L*I_r;

S_f_1 = 0;
S_f_2 = S_f_1 + 0.5*delSf;
S_f_3 = S_f_2 + 0.5*delSf;
S_f_4 = S_f_3 + 0.5*delSf;

S_1 = S_1 - S_f_1 ;
S_2 = S_2 - S_f_2 ;
S_3 = S_3 - S_f_3 ;
S_4 = S_4 - S_f_4 ;

// outflow following GMS
 g_4 = k_s*((I_r)^(1/2));

// initial conditions
D_0 = D_r;   D_10 = D_0; D_30 = D_0;
Q_0 = Q_r;   Q_20 = Q_0;

// inflow sequence
t_st_1 = 150.0;  r_1_0 = Q_0; r_1_1 = (1.0+0.5)*Q_0;
t_st_2 = 300.0;  r_2_0 = 0.;  r_2_1 = -1.00*Q_0;
t_st_3 = 450.0;  r_3_0 = 0.;  r_3_1 = +0.50*Q_0;

T_fin = 600.;

g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st = 1.0; Q_st_0 = Q_0;

// Datatransfer to Plots
// =====================
CCQ = 4;      // number of channels Q + 1 for time
CCD = 5;      // number for D,H
CCF = 4;      // number for FroudeZ
CN = 1200;     // number of clockticks until Tfin
delT = T_fin/CN;  // read-interval for clock ticks
AsizeQ = 1.1*CCQ*CN; // size of Dataarrays for Q
AsizeD = 1.1*CCD*CN;
AsizeF = 1.1*CCF*CN;
```

```
// s_c2_03_03_crunplot
// Glf  24.03.14  26.10.2015

stacksize('max'); exec('s_c2_03_03_context.sce', -1);
importXcosDiagram('s_c2_03_03.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

vcolorQ = [1, 2, 5];
vcolorD = [1, 2, 5, 13];
vcolorF = [1, 2, 13];

T0 = 100.;//suppress initial condition response from plot
T1 = T_fin;

f1 = scf(1); clf();
f1.auto_resize = "on";
f1.figure_size = [626.,420.];
```

```
plot2d(Q.time, Q.values, vcolorQ, rect=[T0,0.,T1,100.]);
xtitle("Q_in, Q_out, Q_2");
xgrid(1); h1 = legend(['Q_in';'Q_out';'Q_2'],4);

f2 = scf(2); clf();
f2.auto_resize = "on";
f2.figure_size = [626.,420.];
plot2d(D.time, D.values, vcolorD, rect=[T0,-3.0,T1,8.0]);
xtitle("D_1 and D_3, H_1 and H_3");
xgrid(1);h1 = legend(['D_1';'D_3';'H_1';'H_3'],4);

f3 = scf(3); clf();
f3.auto_resize = "on";
f3.figure_size = [626.,420.];
// plot2d(F.time,F.values,vcolorF,rect=[T0,0.,T1,0.5]);
   plot2d(F.time,F.values,vcolorF,rect=[T0,0.,T1,5.0]);
xtitle("F_1, F_2, F_3");
xgrid(1);h1 = legend(['F_1';'F_2'; 'F_3'],4);
```





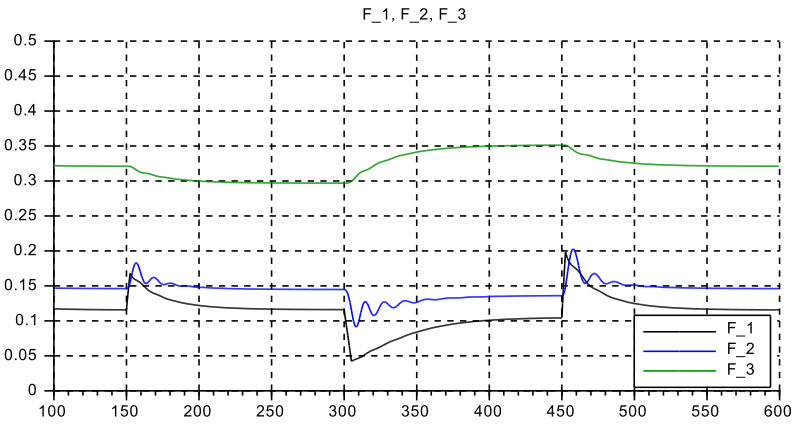


Figure 2.13: *s_c2_03_03_* for 'confusor' cross section

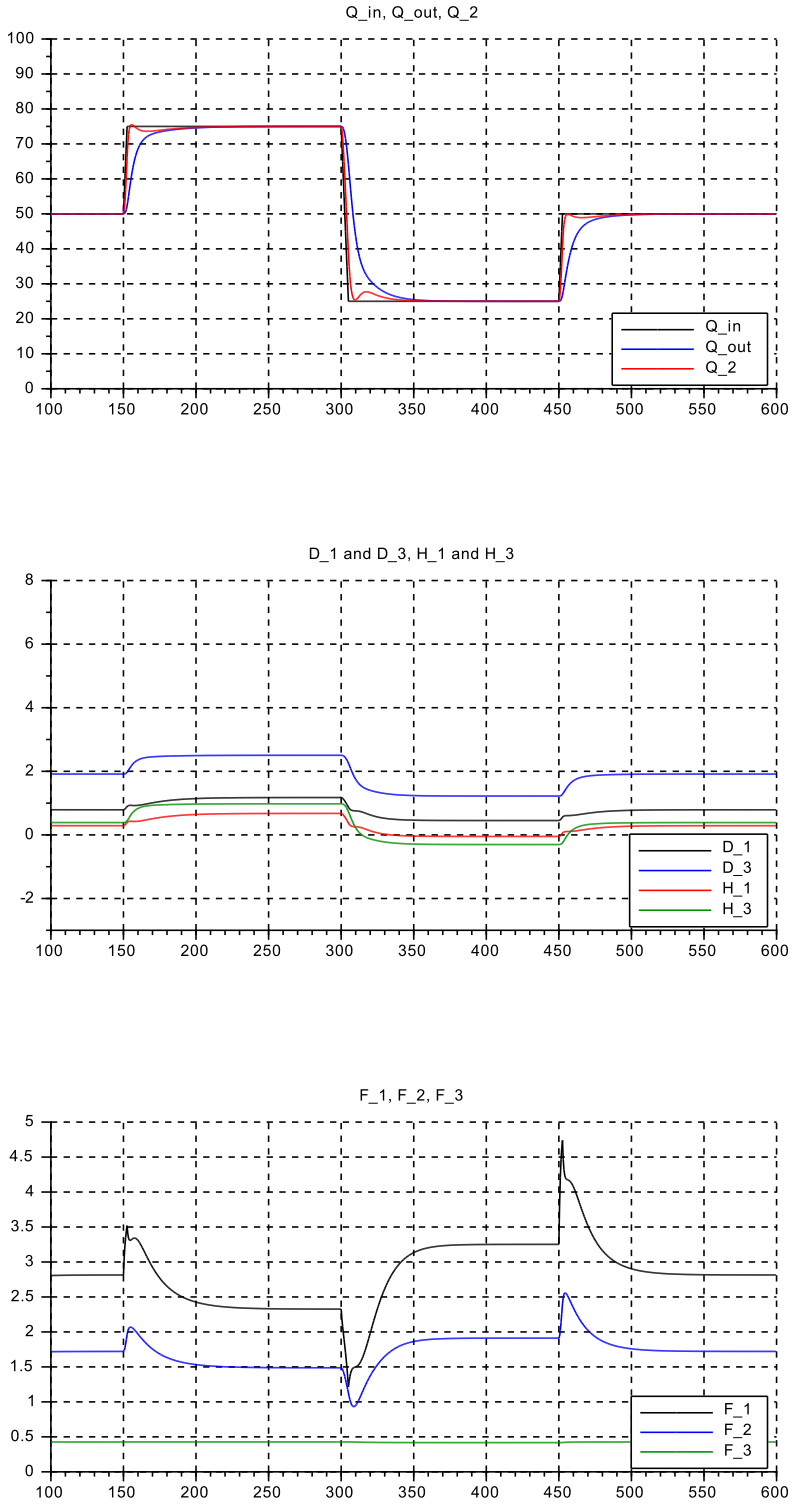
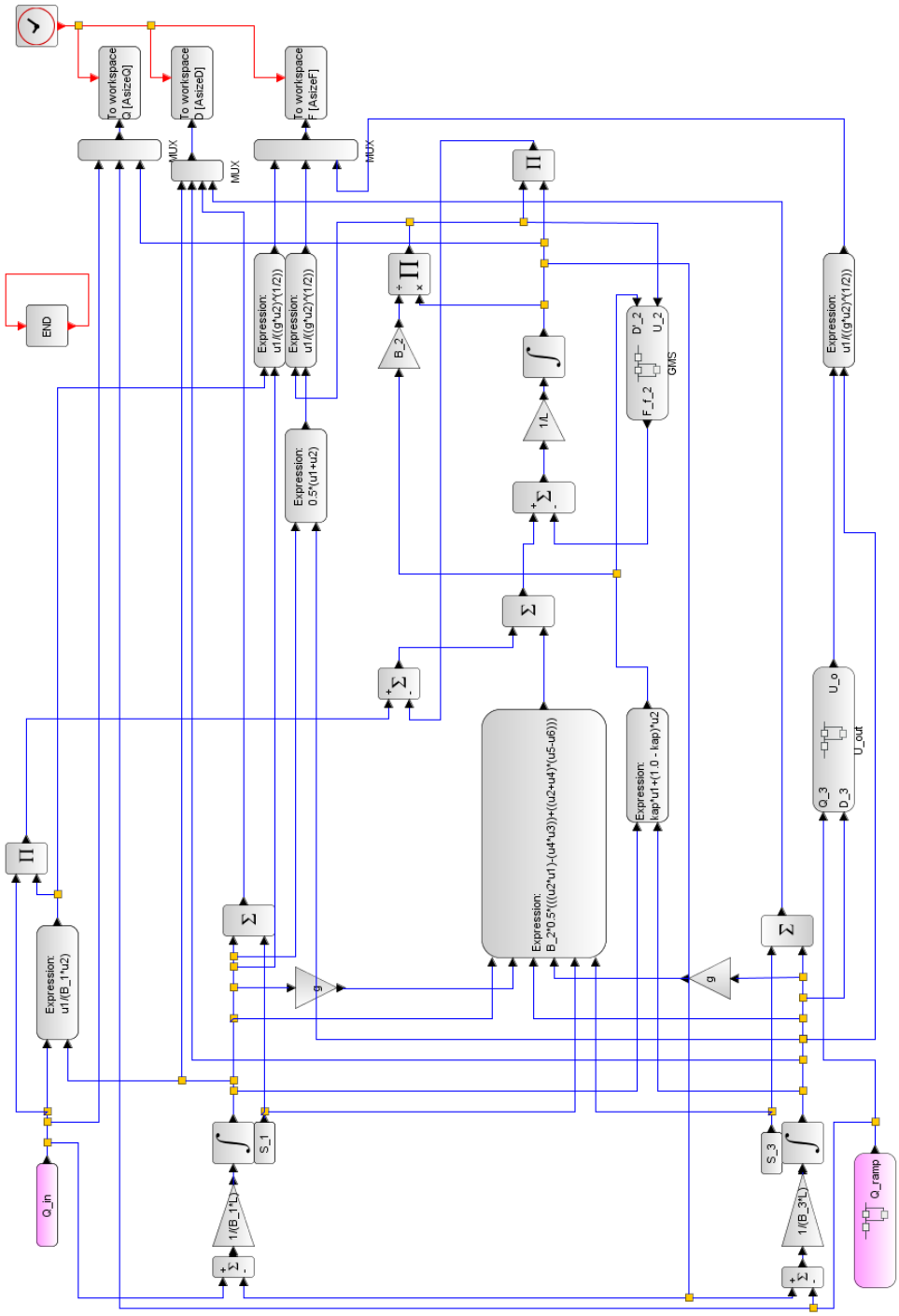


Figure 2.14: *s_c2_03_02_* for 'diffusor' cross section

2.8.4 case s_c2_03_04_ power station inlet channel, outflow variation, inflow fixed

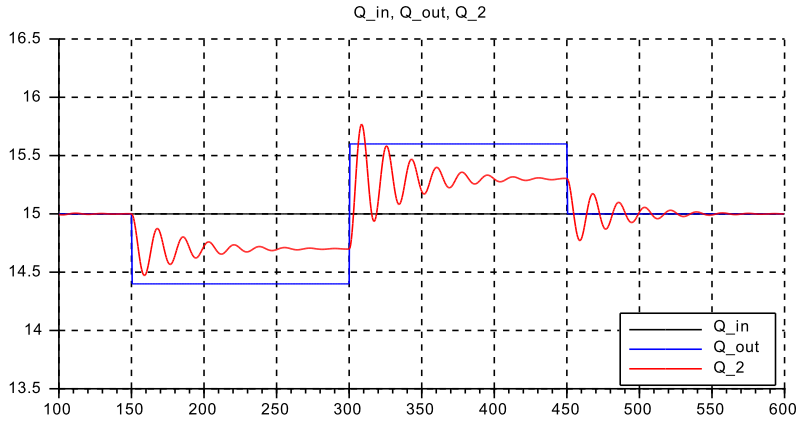
s_c2_03_04.zcos

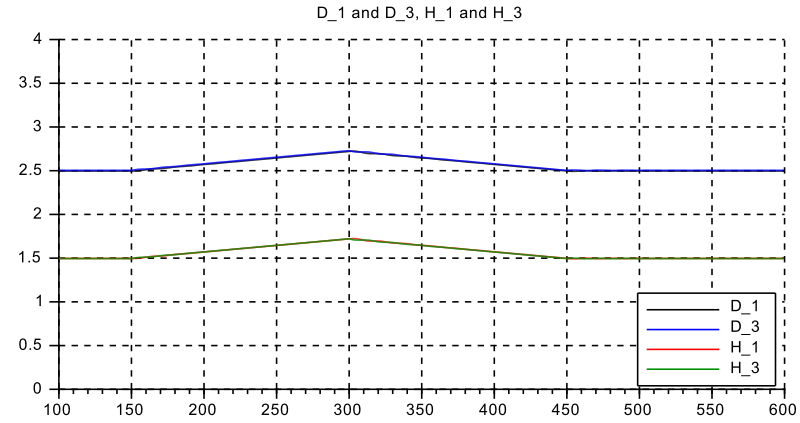


.sce-files

```
// s_c2_03_04_context
// Glf  25.03.2014 26.10.2015

g = 10.;
L = 20.;

D_r = 2.5;
S_r = 1*(-1.0);
B_r = 10.;
U_r = 2.0;
Q_r = 50.0;

// flow operating point: select one
   Q_0 = Q_r*0.30;
// Q_0 = Q_r*1.00;
// Q_0 = Q_r*1.20;

// option: higher U, lower D, -> higher Fr
// D_r = 1.667;
// U_r = 3.0;

// Operating point, by Froude-number
  F_r = 0.2;
//F_0 = 0.4;
//F_r = 0.5;


// friction coeff
   k_s =  70.7;
// k_s =  40.0;
// k_s =  31.6;

   kap = 1.0;

// channel shape:
   B_1 = 1*B_r;
   B_3 = 1*B_r;
     B_2 = 0.5*(B_1 + B_3);
     B_4 = B_3;

   S_1 = 1.0*S_r;
   S_3 = 1.0*S_r;
     S_2 = 0.5*(S_1 + S_3);
     S_4 = S_3;

// reference depth
  D_0 = D_r;

// bottom shape for _r conditions
Rtilda_r = ((B_r*D_r)/(B_r+2*D_r))^(2/3);
I_r = (Q_r/(B_r*D_r*k_s*Rtilda_r))^2;
delSf = + L*I_r;

S_f_1 = 0;
S_f_2 = S_f_1 + 0.5*delSf;
S_f_3 = S_f_2 + 0.5*delSf;
S_f_4 = S_f_3 + 0.5*delSf;

S_1 = S_1 - S_f_1 ;
S_2 = S_2 - S_f_2 ;
S_3 = S_3 - S_f_3 ;
S_4 = S_4 - S_f_4 ;

// initial conditions
   D_10 = D_0; D_30 = D_0;
   Q_20 = Q_0;

// outflow sequence
t_st_1 = 150.0;  r_1_0 = Q_0; r_1_1 = (1.0-0.04)*Q_0;
t_st_2 = 300.0;  r_2_0 = 0.;  r_2_1 = +0.08*Q_0;
t_st_3 = 450.0;  r_3_0 = 0.;  r_3_1 = -0.04*Q_0;

T_fin = 600.;

// outflow flow ramper
g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st = 1.0; Q_st_0 = Q_0;

// inflow 'sequence'
Q_in = Q_0;


// Datatransfer to Plots
// =====================
CCQ = 4;      // number of channels Q + 1 for time
CCD = 5;      // number for D,H
CCF = 4;      // number for FroudeZ
CN = 1200;     // number of clockticks until Tfin
delT = T_fin/CN;  // read-interval for clock ticks
AsizeQ = 1.1*CCQ*CN; // size of Dataarrays for Q
AsizeD = 1.1*CCD*CN;
AsizeF = 1.1*CCF*CN;
```

```
// s_c2_03_04_crunplot
// Glf   25.03.14  26.10.2015

stacksize('max'); exec('s_c2_03_04_context.sce', -1);
importXcosDiagram('s_c2_03_04.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

vcolorQ = [1, 2, 5];
vcolorD = [1, 2, 5, 13];
vcolorF = [1, 2, 13];

T0 = 100.;//suppress initial condition response from plot
T1 = T_fin;

f1 = scf(1); clf();
f1.auto_resize = "on";
f1.figure_size = [626.,420.];
   plot2d(Q.time,Q.values,vcolorQ,rect=[T0,13.5,T1,16.5]);
// plot2d(Q.time,Q.values,vcolorQ,rect=[T0,47.,T1,53.]);
// plot2d(Q.time,Q.values,vcolorQ,rect=[T0,57.,T1,63.]);
xtitle("Q_in, Q_out, Q_2");
xgrid(1); h1 = legend(['Q_in';'Q_out';'Q_2'],4);

f2 = scf(2); clf();
f2.auto_resize = "on";
f2.figure_size = [626.,420.];
   plot2d(D.time,D.values,vcolorD,rect=[T0,0.0,T1,4.0]);
xtitle("D_1 and D_3, H_1 and H_3");
xgrid(1);h1 = legend(['D_1';'D_3';'H_1';'H_3'],4);

f3 = scf(3); clf();
f3.auto_resize = "on";
f3.figure_size = [626.,420.];
   plot2d(F.time,F.values,vcolorF,rect=[T0,0.,T1,0.2]);
// plot2d(F.time,F.values,vcolorF,rect=[T0,0.,T1,0.5]);
// plot2d(F.time,F.values,vcolorF,rect=[T0,0.,T1,0.6]);
xtitle("F_1, F_2, F_3");
xgrid(1);h1 = legend(['F_1';'F_2'; 'F_3'],4);
```





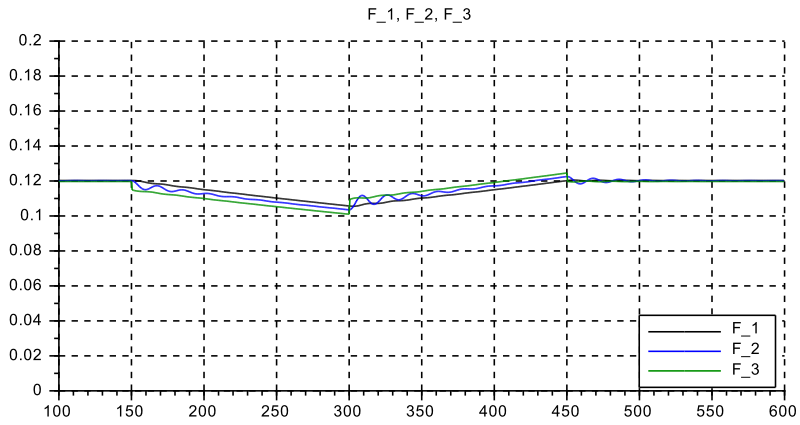


Figure 2.15: $s\_c2\_03\_04\_$ for inflow variations at $Q\_0 = 0.30 \cdot Q\_r$

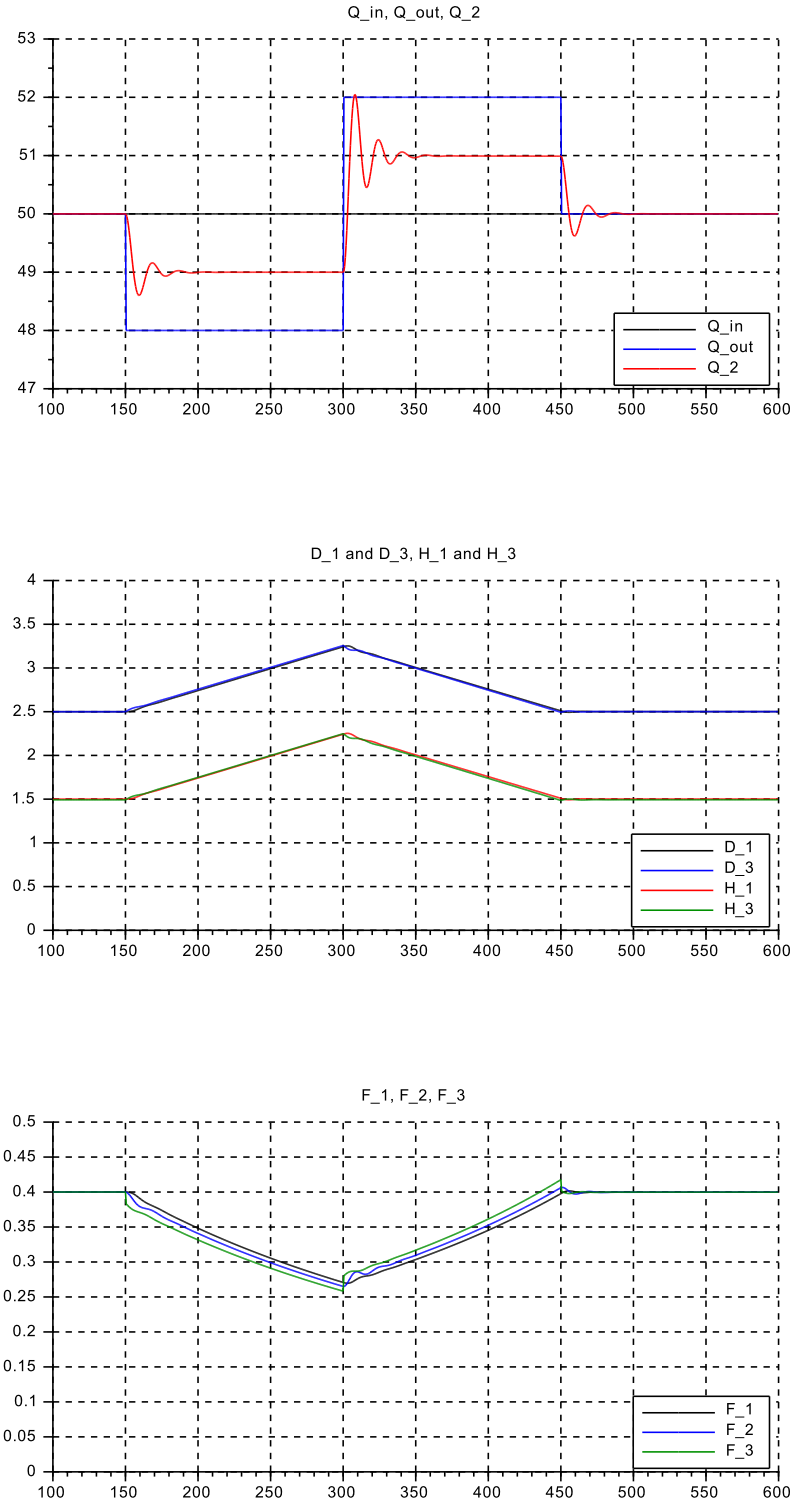


Figure 2.16: $s\_c2\_03\_04\_$ for inflow variations at $Q\_0 = 1.00 \cdot Q\_r$

2.9 Discussion

Case s_c2_03_01_ river bed, constant cross section, low/high flow

Fig. 2.7 There is a strong oscillation on Q_2, but not on $Q_{in},\ Q_{out}$: This is the typical 'sloshing' phenomenon. The oscillation period $T_F := (2 \cdot \pi/\sqrt{2}) \cdot (L/U_F)$ is computed to 17.8 s, and observed at $\approx 18\ s$. The oscillation amplitudes on $H,\ D$ are much reduced by the filter effect of the first order volume balances.
F_{in} has a strong initial peak due to Q_{in} increasing rapidly whereas D_{in} is moving much slower. And F_{out} does not change significantly as Q_{out} is given by D_{out} from the GMS-law.

Fig. 2.8 At low flow ($Q = 12.5\ m^3/s$) the damping ratio is approx. the same, as the Froude number is approx the same. The oscillation period changes, as $D_{in} \approx 1.0\ m \quad \rightarrow \quad U_F \approx 3.16\ m/s$, to 28.1 s, and observed $\approx 30\ s$

Fig. 2.9 At very high flow ($Q = 800\ m^3/s, B = 40\ m$) the damping ratio increases, as the Froude number increases to $F \approx 0.52$. The oscillation period changes, as $D_{in} \approx 5.3\ m \quad \rightarrow \quad U_F \approx 7.3\ m/s$, to 12.2 s, and observed $\approx 11\ s$

Case s_c2_03_02_ river bed, constant cross section, low/high Froude Number

Fig. 2.10 With $F = 0.20$ the damping is significantly weaker, whereas the sloshing period is approx. the same.

Fig. 2.11 With $F = 0.80$ the damping is significantly stronger, and the sloshing nearly disappears.

Fig. 2.12 with $F = 1.60$ that is supercritical flow, the sloshing oscillation fully disappears, and the damping is 'supercritical' (meaning two distinct real poles of the transfer function)

Case s_c2_03_03_ river bed, reference flow, conical cross section

Fig. 2.13 For the 'confusor' case, $F_{in} \approx 0.11,\ F_2 \approx 0.15,\ F_{out} \approx 0.32$, the damping is slightly reduced, and the sloshing period as well, due to the larger depth D_2 and increased U_{2_F}: T_F is observed at $\approx 15\ s$

Fig. 2.14 For the 'diffusor' case, the flow is highly supercritical: $F_{in} \approx 2.8,\ F_2 \approx 1.7,\ F_{out} \approx 0.46$. Q_2 rises nearly as fast as Q_{in}, but then there is a well damped equilibration between Q_2 and Q_{out}.

Case s_c2_03_04_ turbine inlet channel, low/reference flow

Fig. 2.15 For low flow $Q = 15\ m^3/s;\ \Delta Q = \pm 0.60\ m^3/s$ at $F = 0.12$ the damping is very low. 8 (!) sloshing periods are visible, with observed period $\approx 18\ s$. This is due to the change of the outflow impedance, from GMS-outflow (depending on D_{out}) to fixed outflow (not dependent of D_{out}).
The peak level excursion is $\Delta H \approx 0.24\ m$.

Fig. 2.16 For nominal flow, $Q = 50\ m^3/s;\ \Delta Q = \pm 2.0\ m^3/s$ and $F = 0.40$, the peak level excursion is $\Delta H \approx 0.80\ m$. The damping is higher now, but still 3 sloshing periods are visible, in contrast to 1.5 visible periods with the GMS-outflow. The observed period is also $\approx 18\ s$.

To summarize, all the findings conform well with the expected behaviour and trends, and the observed sloshing period values agree well enough with the precalculated ones.

Chapter 3

The Long River/Channel Model

3.1 Overview

The **aim** of this chapter is to extend the model with three compartments discussed in the previous chapter in order to provide a better spacial resolution and to display higher oscillation modes, which is needed for long and comparatively narrow river beds or channels.
This will be achieved by connecting in series ('chaining up') a number N of basic elements, consisting each of one volume balance compartment and one momentum balance compartment. A single volume balance compartment will be placed at the outflow end of this chain. Its outflow is generated by a non-dynamic function, which is selected according to the downstream boundary condition, similar as with the three-compartment-model from above. Thus there will be $N+1$ volume compartments and N momentum interleaved compartments along the longitudinal axis.
In order to keep complexity at a reasonable level the total channel length L_tot for all compartments shall be the same, that is the compartment length L of the modelled system will be $L = L\_tot\ /(N+1)$. Further the number of basic elements N is set to

$$N := 20$$

This has turned out to be a reasonable compromise between sufficient spacial resolution ($\approx 5\%$ of L_tot) and non-excessive simulation runtime per case ($\leq$ 60sec). Overall there will be $40 + 1$ interleaved compartments. The selection of N will be discussed in more detail in the next section.

Next the **implementation framework** in *scilab 5.4.1* will be documented in more detail, and how the simulations are to be run.

The **test cases** for the 'numerical experiments' / 'simulations' will mirror the typical operating conditions of the modelled system. They will be similar to what has been used in the previous chapter. More details will be given in the corresponding sections below.

Finally an **application** to 'Birsfelden' will be presented, where measured responses are available for comparison.

Remarks.
The dynamic models in this chapter are valid only for systems with dominant longitudinal geometry

$$D \ll B \ll L_{tot}$$

with D as nominal water depth and B as nominal channel width.
They are also only valid for relatively small and slow variations of water level H, such that the vertical dynamics (that is the momentum balances in the vertical direction) may be neglected.
And finally they shall cover sub-/ trans-/ super-critical flow regimes. To insure this the spacial discretisation parameter $\varkappa$ (see section 2.2.2) is set to $\varkappa := 1.0$ to 'stay on the safe side' concerning dynamic stability.

3.2 Implementation in scilab/xcos 5.4.1

3.2.1 Basic Structure

As mentioned in section. 2.7, each operating case (x = 0 ... 5) is covered by a triplet of files, the `s_c3_41_0x.zcos`-diagram and two `s_c3_41_0x.sce`-scripts:

- The `s_c3_41_0x.zcos` diagram (Fig.3.1) contains the signal flow graph around the main superblock `Blockk01tk39`, which encapsulates the $N = 20$ basic elements with two compartments each. And there is also the separate outflow volume balance integrator `Blockk41`.
  Tinted in violet are blocks on the upper and lower border of the graph, which generate the inflow and outflow specific to the layout, that is the appropriate boundary conditions.
  The blocks `To workspace` on the right border collect the simulation run data and put them into the `scilab`-workspace for post-processing, essentially plotting.
  The actual parameters for the simulation are transmitted to the graph through the `'context'`-feature.

  The main superblock is structured in three levels of sub-superblocks.
  The first level down (Fig.3.2) contains five further superblocks identified by `SuBlok01to07`, `SuBlok08to15`, `SuBlok16to23`, `SuBlok24to31`, `SuBlok32to39`.
  On the second level down (Fig.3.3) each of them contains four sub-sub-superblocks, for instance for `SuBlok01to07` the sub-sub-superblocks `k01`,`k03`,`k05`,`k07`[1].
  And on the third level down (Fig.3.4), for instance the sub-sub-superblock `k01` contains the signal flow graph for the basic element, consisting of

  - the volume balance integrator for water depth $D(k)$
  - and the momentum balance integrator for $Q(k+1)$

  And in its own context window the index k is set to $k = 01$.
  The parameter entries in the `.zcos`-blocks in the signal flow diagram are given as (for instance) $vB(k)$ for the channel width at location k and $vB(k+1)$ at location $k+1$.
  And the next sub-sub-superblock `k03` will have $k = 03$ in its context window.

  This allows all parameter values for the lowest level sub-sub-superblocks to be entered on the top level, in the `_context.sce`-file in an array of appropriate length: $1 \ldots 2 * N + 1$[2].
  All integrator blocks have lower saturations at +0.0001, which will suppress numerical difficulties at zero depth and/or reverse flow. Several blocks for mathematical expressions[3] are used to keep the signal flow graph more compact.

- A first `scilab`-script named `s_c3_41_0x_context.sce` sets and evaluates all parameter values for the `s_c3_41_0x.zcos` file which are needed for the simulation at the specific operating point.
  Specific operating point variable values are selected manually in this script by 'comment'-ing and 'uncomment'-ing.
  At the lower end of the script, the data transfer into the `scilab`-workspace is set up.

- The second `scilab`-script named `s_c3_41_0x_crunplot.sce` lets the simulation run in batch-mode, and plots the simulation results using standard `scilab`-plot routines. Note that appropriate scaling on the plots for the specific operating point may be necessary, again using the 'commenting' / 'de-commenting' technique.

3.2.2 Handling a simulation run

This is done in the same way as given in section 2.7.3 (see there).

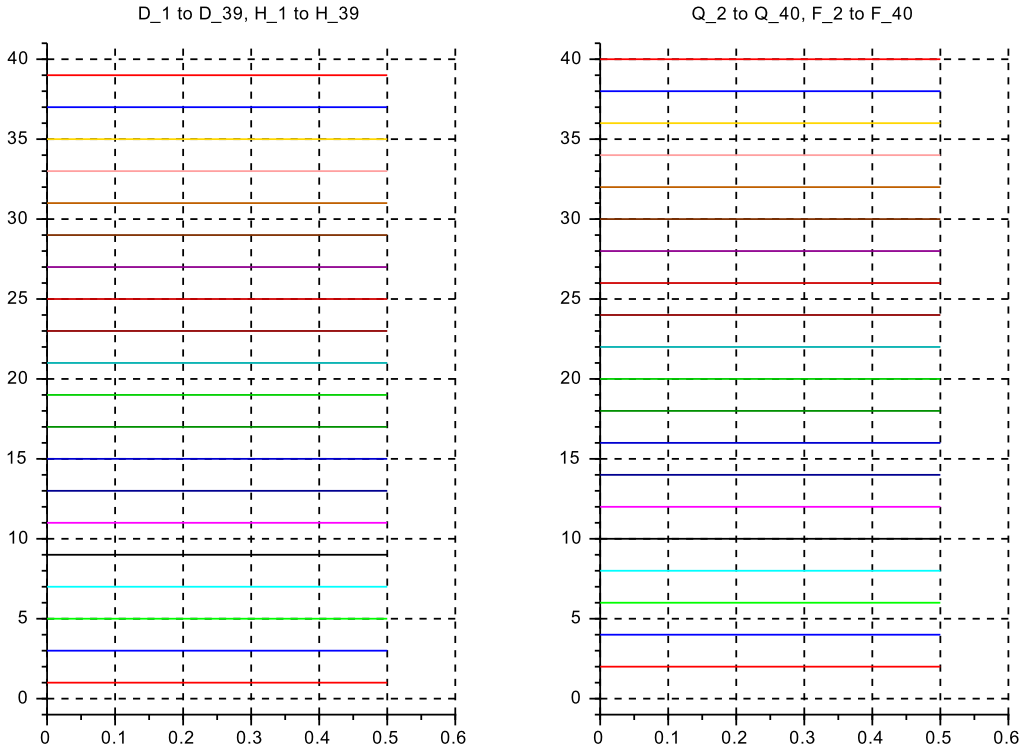
[1] Note that these two levels only have been introduced to produce manageable figure sizes for on-screen display and documentation

[2] ... + 1 allows to include the parameter values for the single outflow integrator `Blockk41`

[3] use 'expression' block from the 'user defined functions' palette

3.2.3 Color coding for simulation run results

Figure 3.1: $s\_c3\_41\_0x$. color codes for transients



3.3 The Basic Case s_c3_41_00

This case represents the segment of the river/channel flow of length $L_{tot} = 400\ m$ at reference conditions constant width $B_r = 10\ m$, constant depth $D_r = 2.5\ m$, and reference flow $Q_r = 50\ m^3/s$. This results in the flow velocity $U_r = 2\ m/s$ and the Froude number $F_r = 0.40$.
The bottom is nominally horizontal $vS(k) = 1.0 \forall k$. But in fact it is inclined to be parallel to the water surface slope which is required by the GMS-friction model in steady state flow conditions at nominal U_r, D_r, B_r and at a given Strickler coefficient $k_s := 50$. In other words, $vD(k) \rightarrow D_r \forall k$.
Then a sequence of step flow variations $dQ = \pm 0.40 \cdot Q_r = \pm 20\ m^3/s$ is applied, with ramp limits at $\pm 0.1\ m^3/s/s$, that is over $200s$ for the specified stroke. This is much slower than the typical filling time constant $T_f = 10\ s$ of one volume element.

.zcos Diagrams

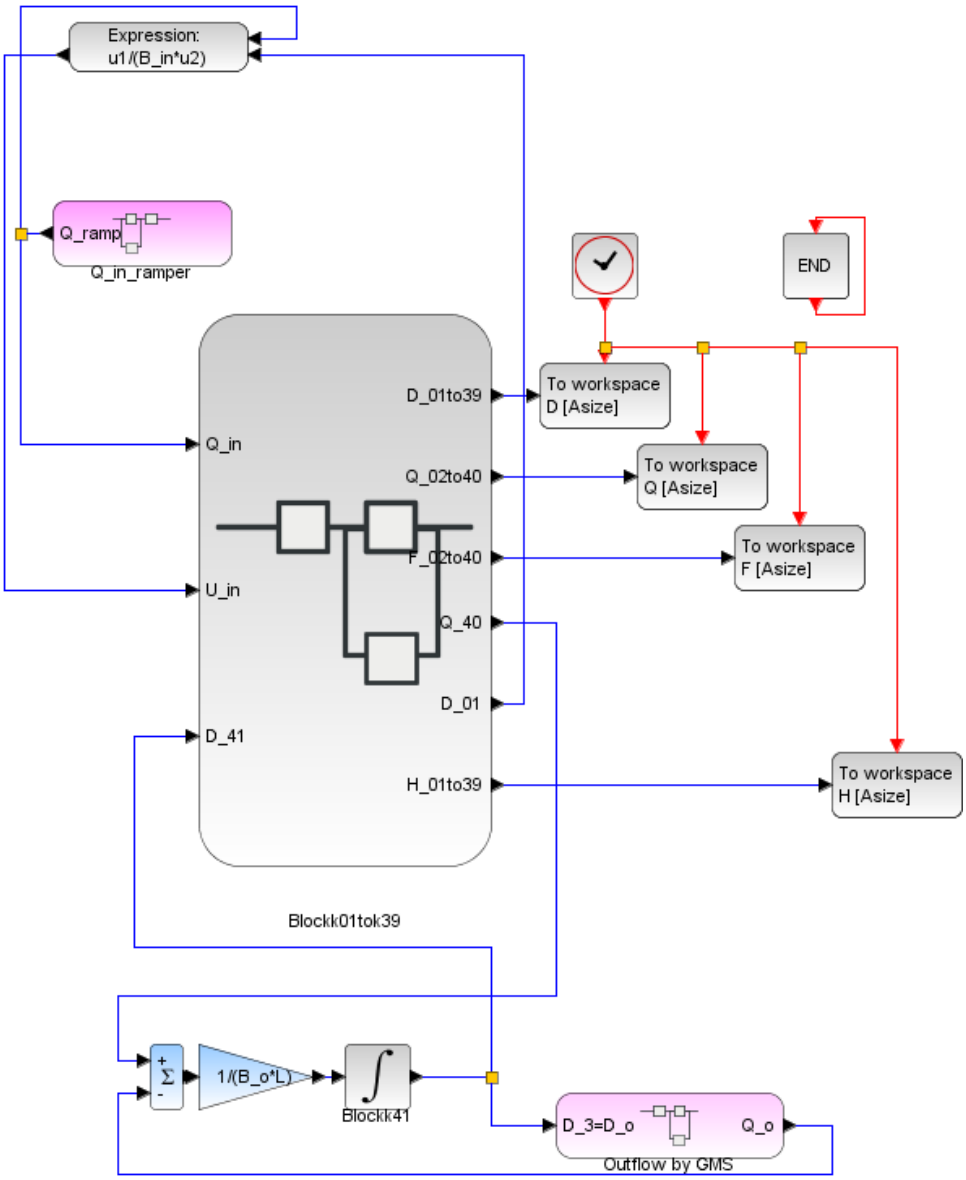


Figure 3.2: *s_c3_41_00*. xcos-Diagram, top level

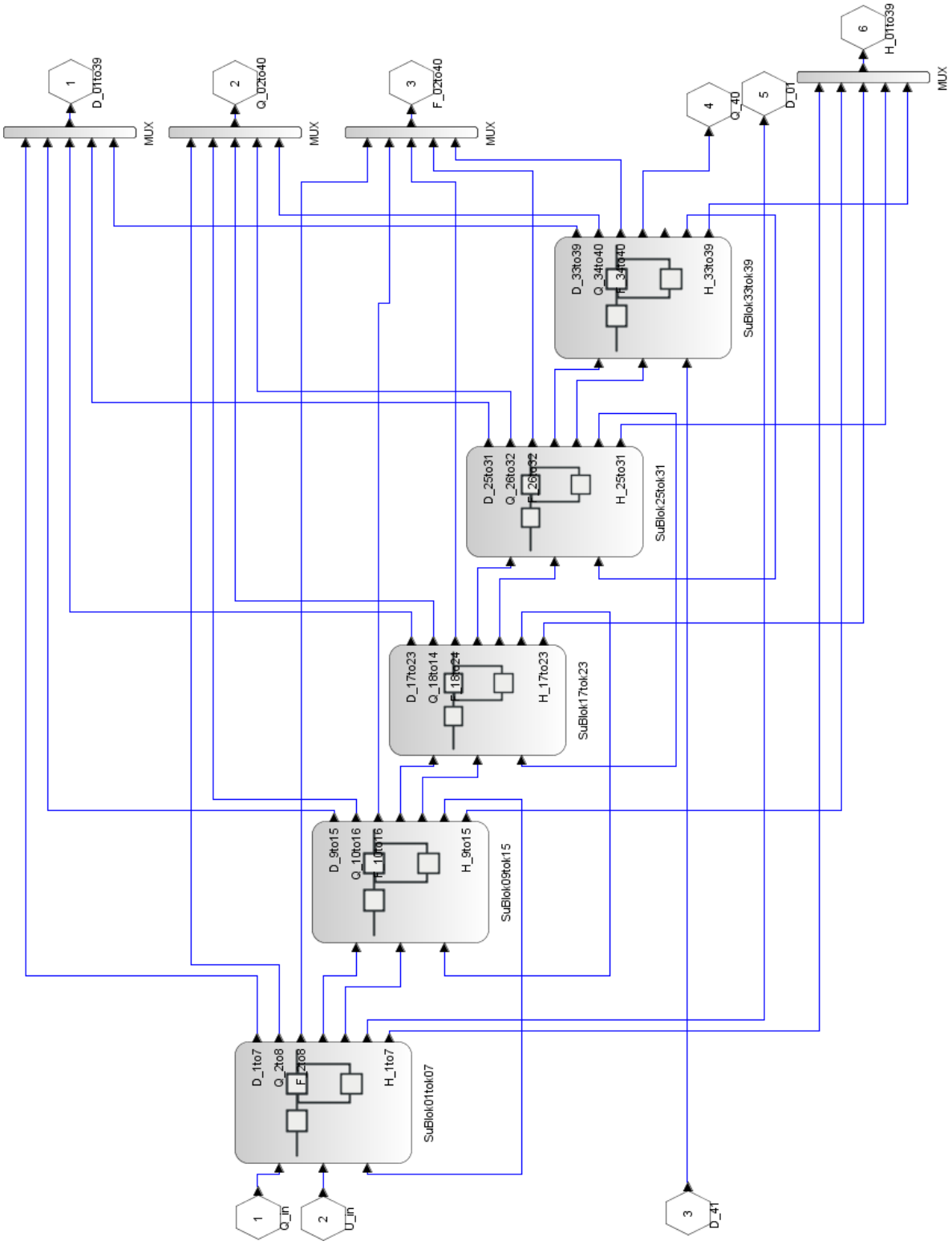


Figure 3.3: *s_c3_41_00.* xcos-Diagram, first level down

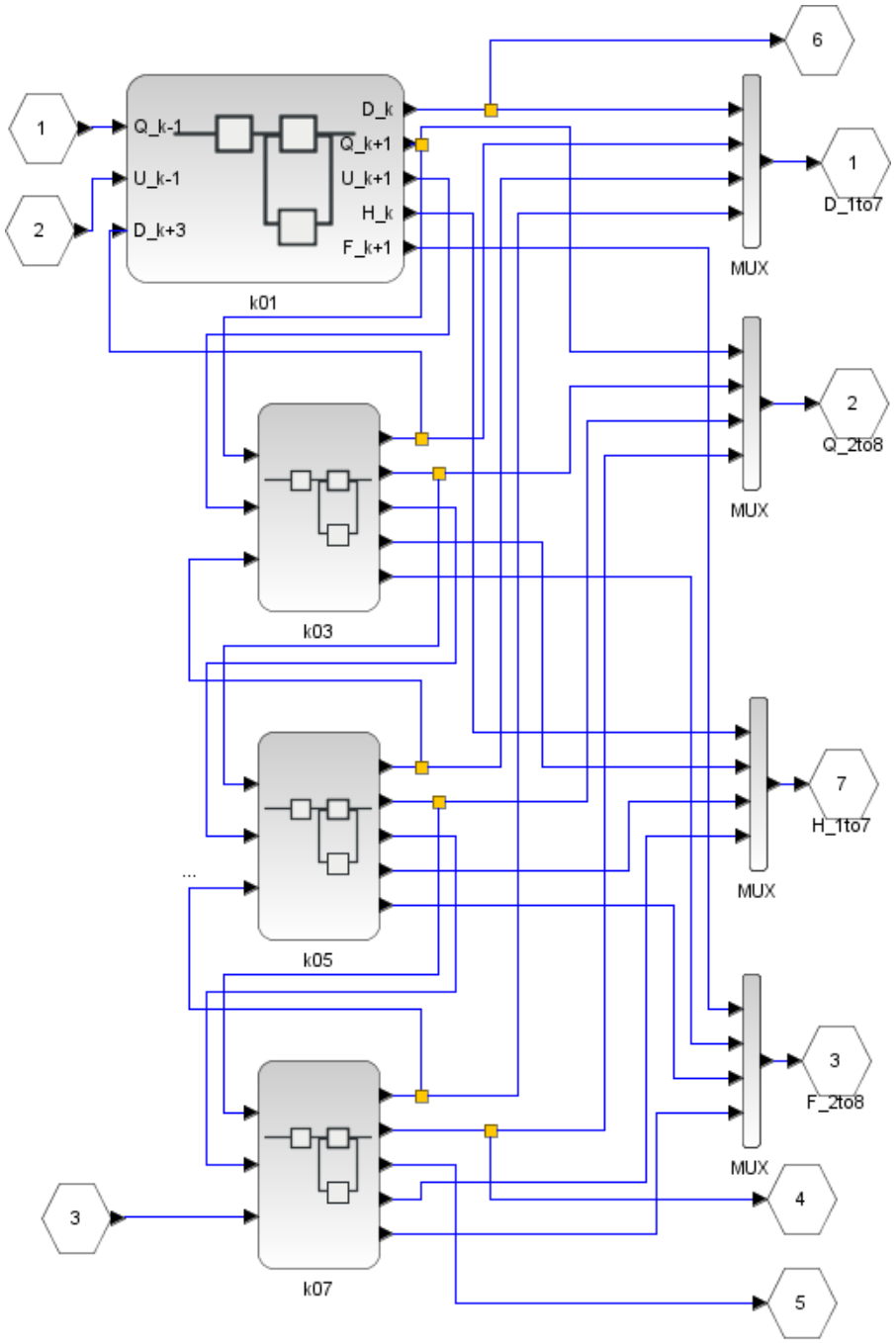


Figure 3.4: *s_c3_41_00*. xcos-Diagram, second level down

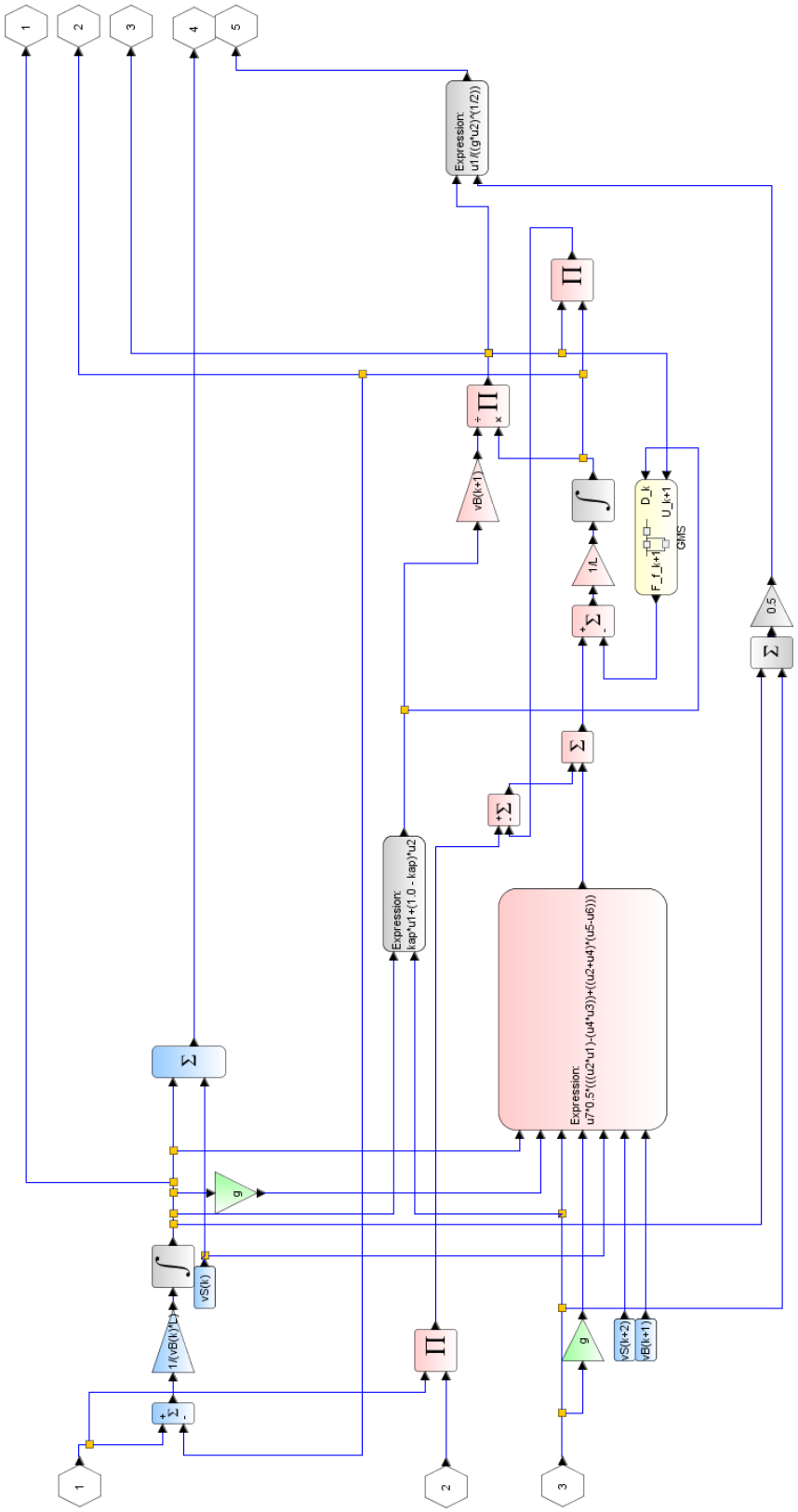


Figure 3.5: *s_c3_41_00*. xcos-Diagram, third level down

.sce file for ‘context’

```
// s_c3_41_01_context
// Glf  26.10.15
// with 20 Segments, in 5 SuBlo4's
// no vertical dynamics

g = 10.;L = 20.;
kap = 1.0;

N= 20;      // no Volume and Momentum-segments

// flow set; choose by 'comment'
   Q_I = 50.0; B_I = 10.0; S_I = 1*(-1.0);

// GMS-friction coefficient
// k_s = 70.7;
   k_s = 50.0;
// k_s = 31.6;

// reference bottom slope
  Q_r = 50.0; B_r = 10.0;
  D_r =  2.5; S_r = 1*(-1.0);
  U_r =  2.0;
  R_r = (B_r*D_r)/(B_r + 2*D_r);
  I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// initialzing 'iteration loop'
D_I = D_r;
gainQ = 0.01*(D_I/Q_I);
Q = k_s*(B_I)*(D_I)*(D_I)^(2/3)*(I_r)^(1/2);
eQ  = -Q_I + Q;

// iteration loop
while (eQ<-0.001)|(eQ>0.001)  then
    R_I = (B_I*D_I)/(B_I + 2*D_I);
    Q   = (k_s*(I_r)^(1/2)*B_I)*D_I*(R_I)^(2/3);
    eQ  = -Q_I + Q;
    D_I = D_I - gainQ*eQ;
end

 D_min = +0.001; D_max = 40*D_r;
 Q_min = +0.001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic layout: constant width
vb = 1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
//****************
vB = B_I*vb;
//******************

//  basic layout: bottom 'horizontal'
vs  = -1*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];

//*****************
vS0 = (-S_I)*vs;
//*************************


vd0 = ones(1,(2*N+1));
vD0 = D_I*vd0;

vq0 = ones(1,(2*N+1));
vQ0 = Q_I*vq0;

// Inflow data
    B_in = vB(1);
    S_in = vS0(1);
    D_in = vD0(1);
    Q_in = vQ0(1);

// friction slope of the bottom vS
vi0    = ones(1,(2*N+1));
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));

for kk=2:2:(2*N),
  vI0(kk)    = I_r*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end


//  outflow data
    B_o = vB(kk+1);
    S_o = vS(kk+1);
    D_o = vD0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
dQ = 0.40;
t_st_1 = 1000.0;  r_1_0 = Q_I; r_1_1=(1.0+1.*dQ)*Q_I;
t_st_2 = 2500.0;  r_2_0 = 0.;  r_2_1 = -2.*dQ*Q_I;
t_st_3 = 4000.0;  r_3_0 = 0.;  r_3_1 = +1.*dQ*Q_I;
T_fin = 6000.;
// inflow slew rate
g_st = 10.0; u_up_st = +1.0; u_dn_st = -1.0;
tau_st = 10.0; Q_st_0 = Q_I;

// outflow fixed:
//Q_o = Q_0;

// outflow by GMS
g_o = k_s*((I_r)^(1/2));

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

.sce file for 'run' and 'plot'

```
// s_c3_41_00_crunplot
//  Glf  2017-03-06
//  no vertical dynamics

stacksize('max'); exec('s_c3_41_00_context.sce', -1);
importXcosDiagram('s_c3_41_00.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:8,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
f1.auto_resize = "off";
plot2d(Q.time,Q.values,vcolor,rect=[0.,0.0,6000,80.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
f2.auto_resize = "off";
plot2d(D.time,D.values,vcolor,rect=[0.,1.0,6000,4.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[0.,0.0,6000,3.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[0.,0.1,6000,0.6]);
xtitle("F_2 to F_40"); xgrid(1);

f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[900.,40.0,2000,80.]);
xtitle("Q_2 to Q_40, zoomed"); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[900.,2.4,2000,3.6]);
xtitle("D_1 to D_39, zoomed"); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[900.,0.0,2000,3.0]);
xtitle("H_1 to H_39,zoomed"); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[900.,0.1,2000,0.6]);
xtitle("F_2 to F_40, zoomed"); xgrid(1);
```

Case s_c3_41_00: Discussion

- Two sets of output plots are shown: Fig.3.6 gives the overall view of the whole responses from 0 to 6000 s whereas Fig.3.7 zooms in on the first flow ramp-up to show more details.

- Note that the traces for $vD(k,t)$ are very close together (nearly identical), as to be expected from the experiment layout, whereas the traces for $vH(k,t)$ show the level inclination due to the GMS-friction model.

- The time delay for the Froude wavefront between the locations of $vQ(2)$ to $vQ(40)$ over the distance of 360 m with $U_{F_r} + U_r = 5.0 + 2.0 = 7.0\ m/s$. would be $\approx 52\ s$, which fits well with the time delay from the simulation result of $\approx 54\ s$

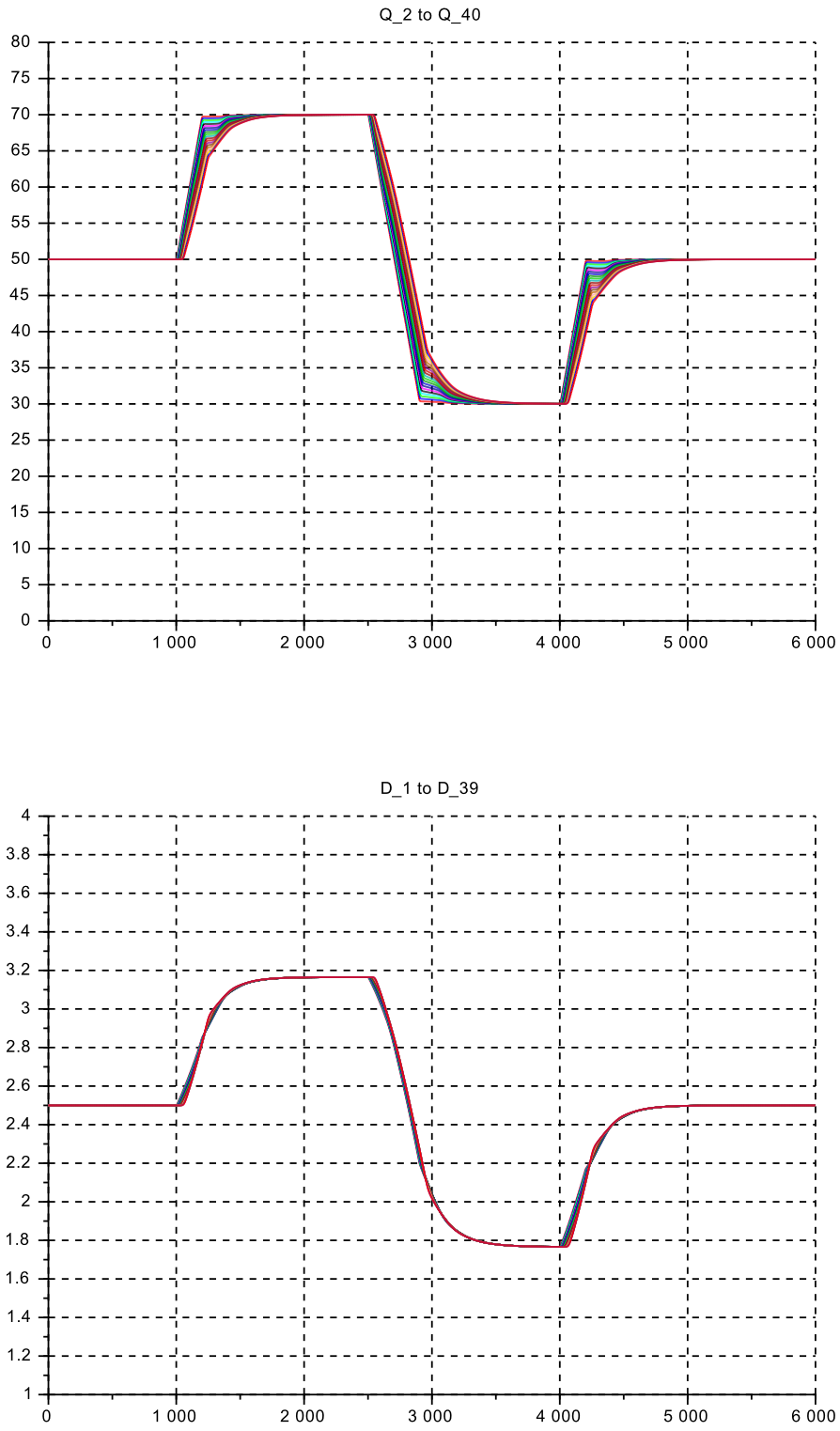
- Essentially no strong sinusoidal (sloshing) oscillations on depth $vD(t)$ and $vH(t)$ are visible. This is to be expected, as the Froude wave will travel downstream through the 'open end' outflow, and will not be significantly reflected backward.

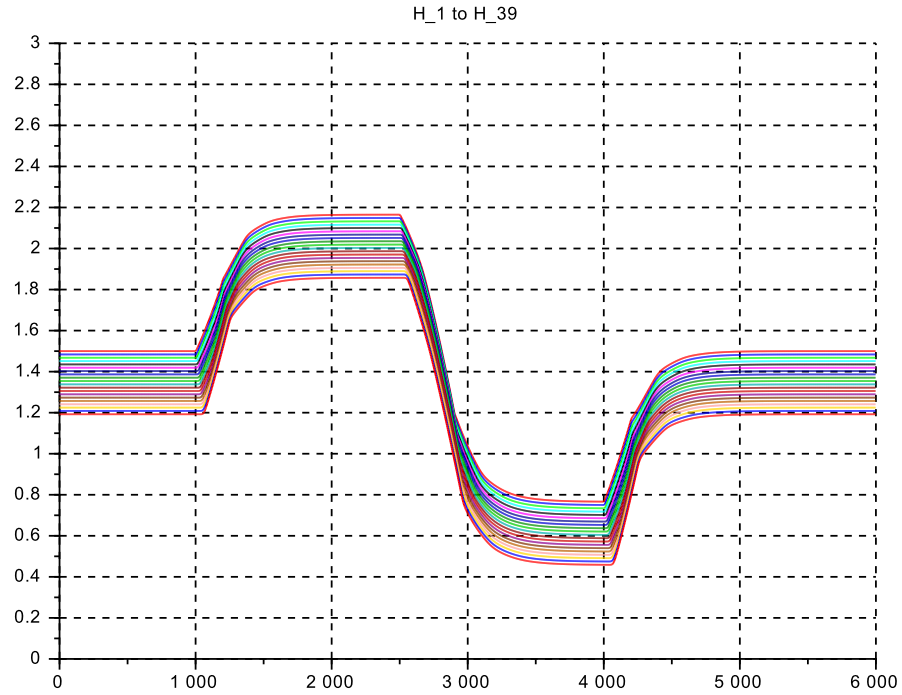
- In $vQ(t)_{zoomed}$ a neat wave front is visible, starting at $t = 1200\ s$ and moving downstream, until $t = 1255\ s$ when it reaches the last $vQ(40)$. Its shape is due to the Froude wave travelling downstream (taking about 2.7 s per compartment) and the filling time of 10 s per compartment.

- However on $vQ(t)$ there is also a much weaker wave front visible which travels upflow from the outflow end. This indicates that the impedance at the outflow end is not perfectly matched.

- Further simulation runs (not shown here) have shown:
  - For faster flow ramps, such as 1 $m^3/s\ /s$, the wave reflection is much stronger
  - and then the compartment length must be increased to $L \approx 160\ m$ in order to reduce the wave reflection to what is seen here.

Figure 3.6: $s\_c3\_41\_00\_$ for flow variations at reference flow $Q = 1.00 * Q\_r$




H_1 to H_39
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
1 000
2 000
3 000
4 000
5 000
6 000

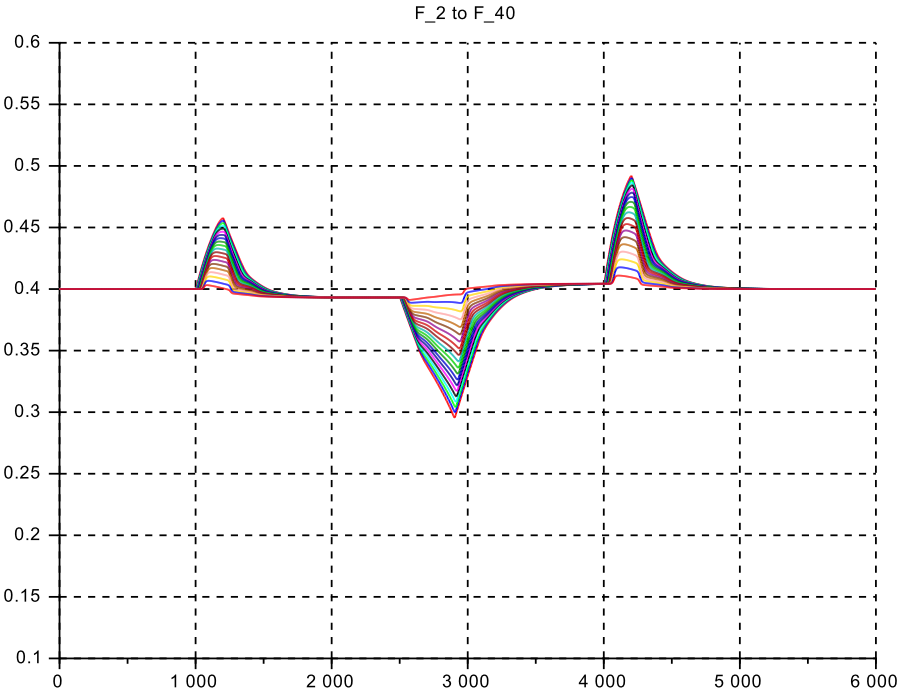
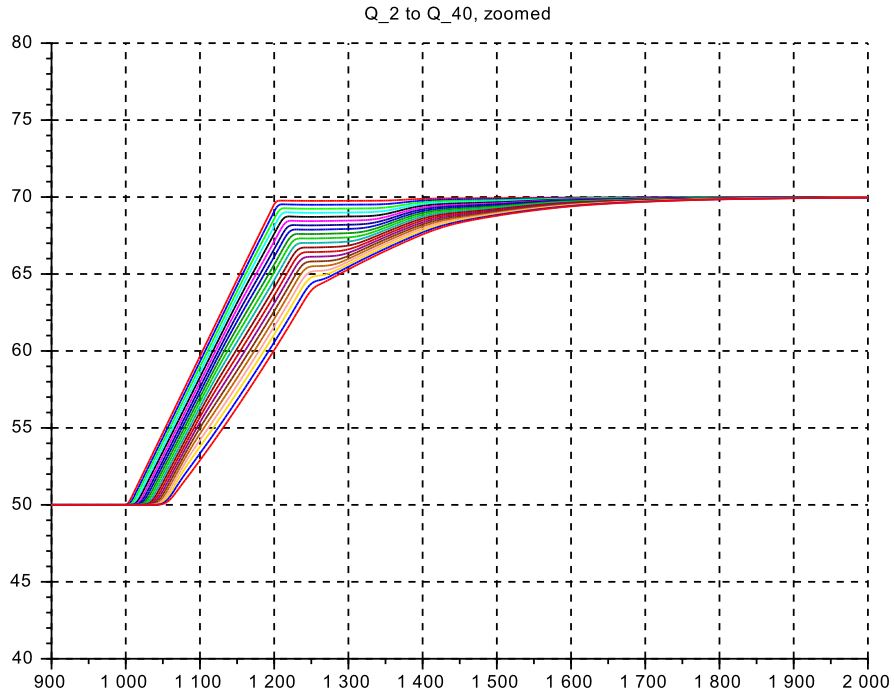
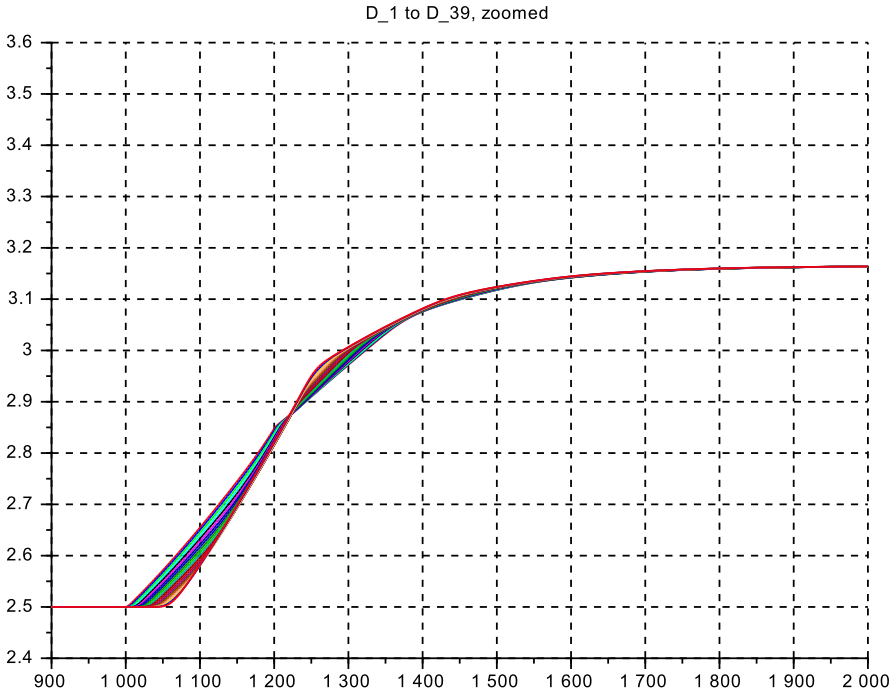
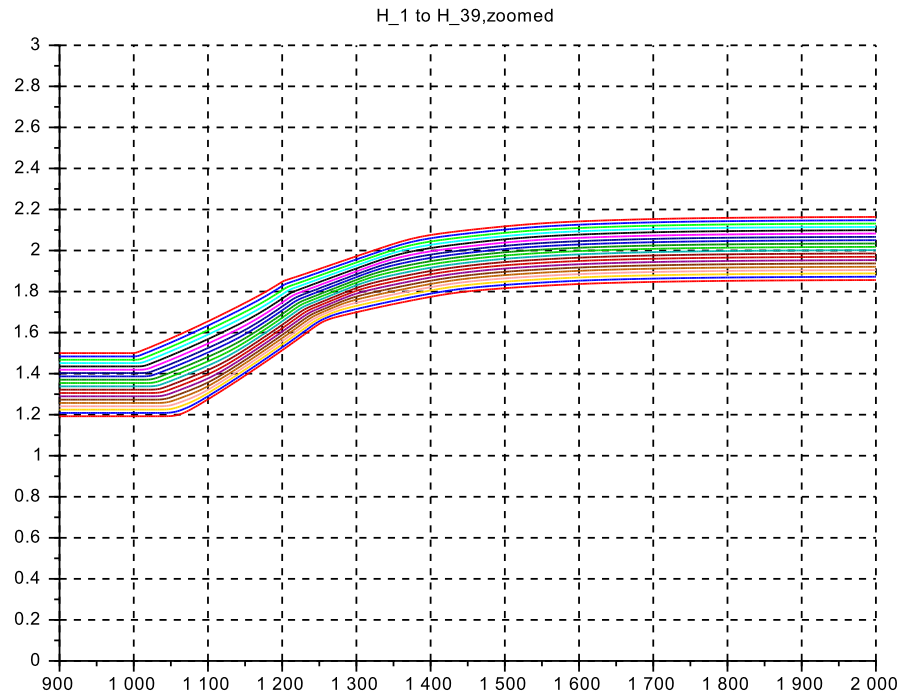
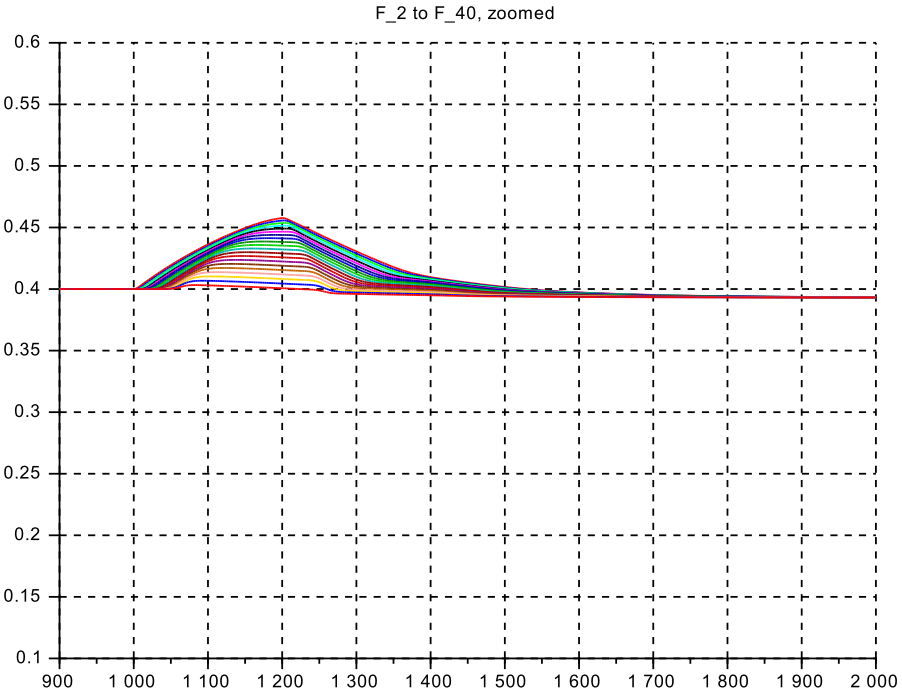

F_2 to F_40
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0
1 000
2 000
3 000
4 000
5 000
6 000

Figure 3.7: 41_01_ at $Q = 1.0 * Q\_r$, zoom-in on first step up






H_1 to H_39,zoomed
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
900
1 000
1 100
1 200
1 300
1 400
1 500
1 600
1 700
1 800
1 900
2 000


F_2 to F_40, zoomed
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
900
1 000
1 100
1 200
1 300
1 400
1 500
1 600
1 700
1 800
1 900
2 000

3.4 Case 41_00_N : The space-discretizing parameter N

The aim of this section is to illustrate the effect of parameter N and to document the final choice. Four cases are shown and discussed

$$N = 4, \quad 8, \quad 16, \quad 32$$

all with nominal flow ($Q = 50\ m^3/s$), nominal depth ($D = 2.5\ m$), nominal Froude number ($F = 0.40$), nominal channel dimensions ($B = 10\ m$, $L\_tot = 500\ m$) and with a sharper inflow ramp ($2.5\ m^3/s$ per s) for more details of the 'sloshing'.

- The focus is on the zoomed-in responses of flow Q and level H along the longitudinal axis. The variables D and F are not shown for brevity.
- Consider the lower end case $N = 1$ from Chap.2 as well for comparison.
- Note that the oscillatory mode seen in Chap.2 at $N = 1$ disappears for increasing N and the Froude waves become dominant.
- Besides the increasing number of lines there is not much difference in the general shape of the Q, H responses between $N = 16$ and $N = 32$. – But the size of the output data arrays and the implementation effort increase substantially.
- For $N = 4,\ 8,\ 16,\ 32$ the simulation run-time was measured as $\approx 8\ s,\ \ 17\ s,\ \ 35\ s,\ \ 90\ s$. Thus the tradeoff between quality and effort is no longer favorable for $N \ll 16$.
- This motivates the choice of $N = 20$[4].
  The responses for this case are documented at the end of the section. The simulation run-time was measured at $\approx 45\ s$.

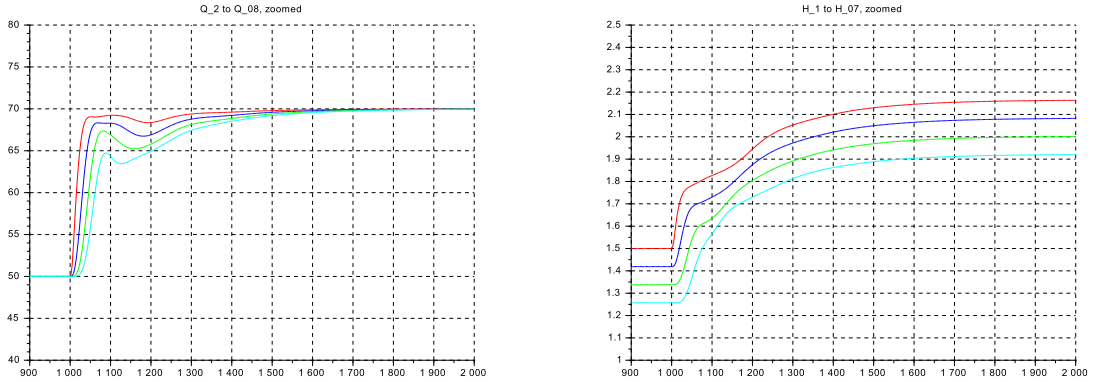


Figure 3.8: $N = 4$: $Q,\ H$ for flow ramp-up at reference flow $Q = 1.00 * Q\_r$

[4] a possible alternative would be $N = 24$, leading to a compartment length of $L = L_{tot}/25$

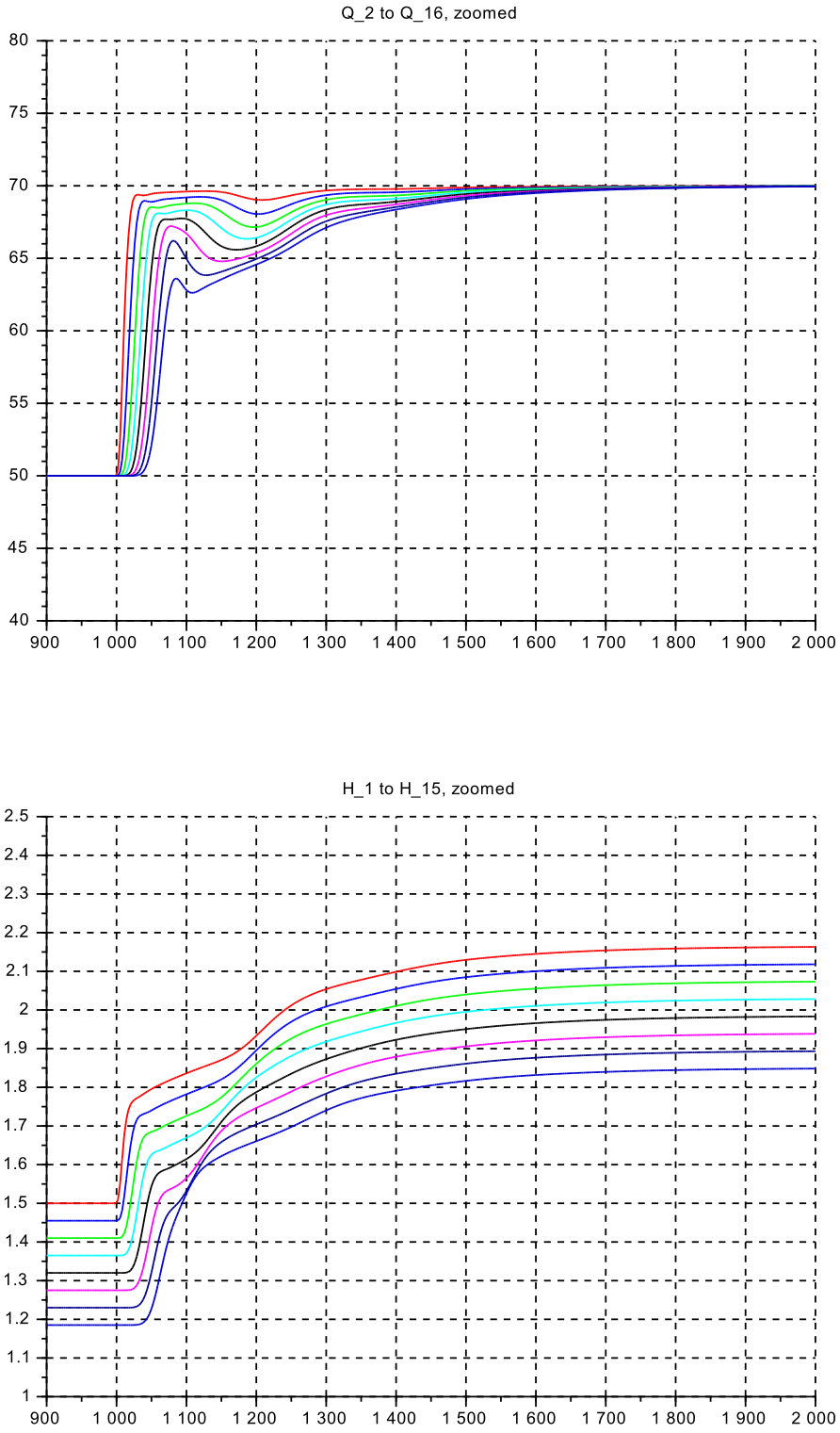


Figure 3.9: $N = 8$: Q, H for flow ramp-up at reference flow $Q = 1.00 * Q\_r$

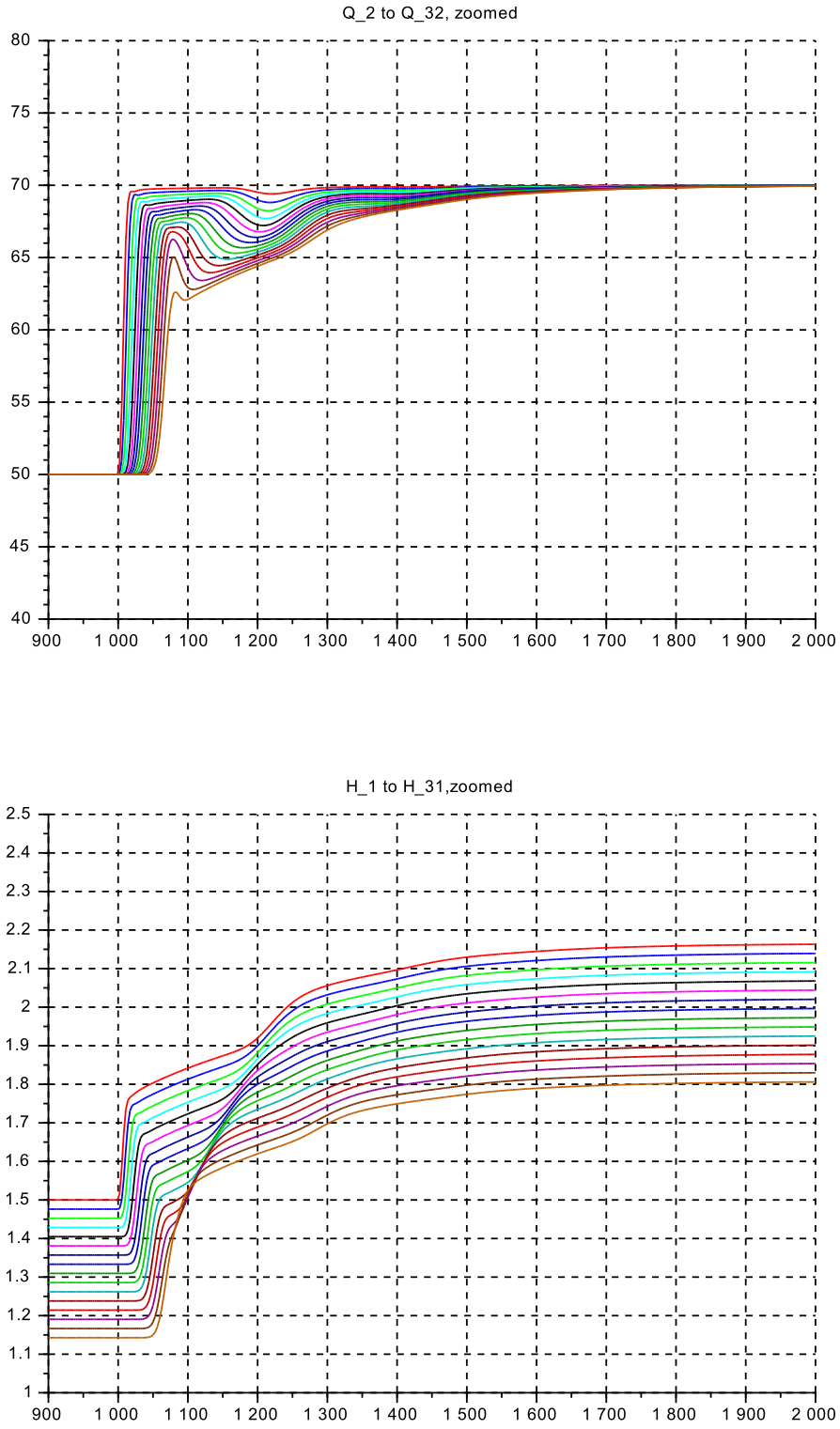
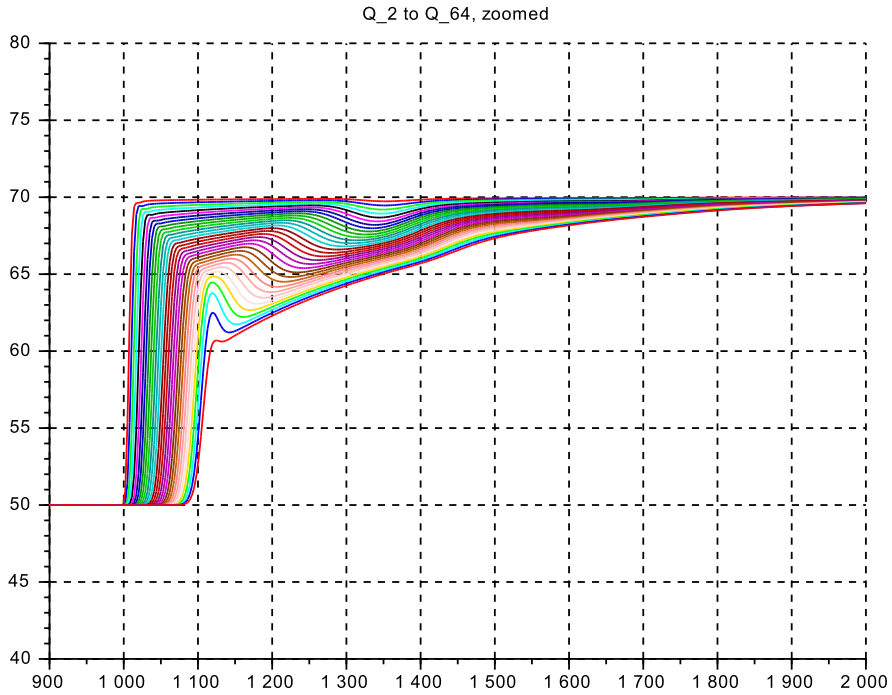


Figure 3.10: $N = 16$: Q, H for flow ramp-up at reference flow $Q = 1.00 * Q\_r$



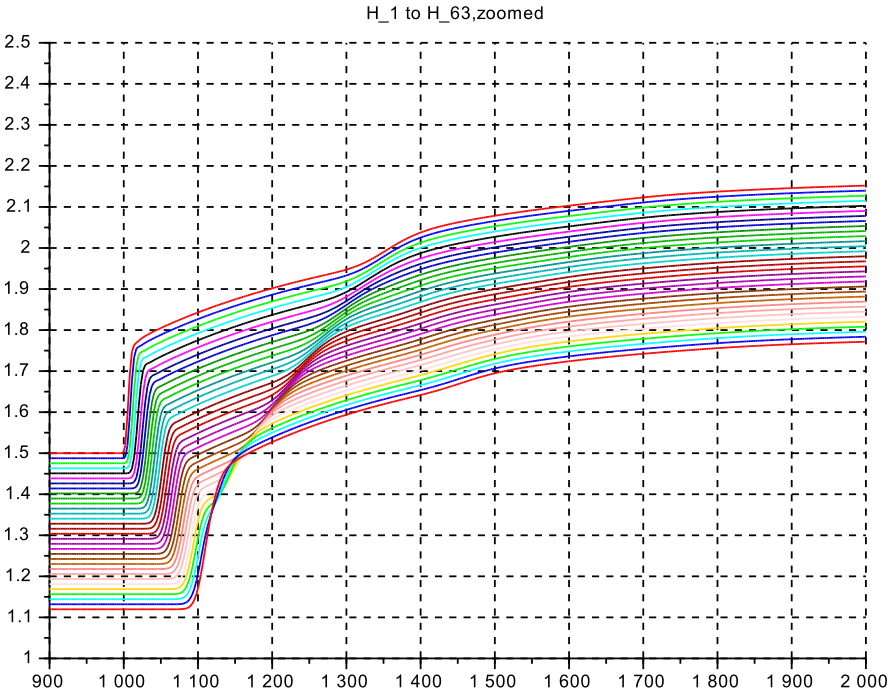
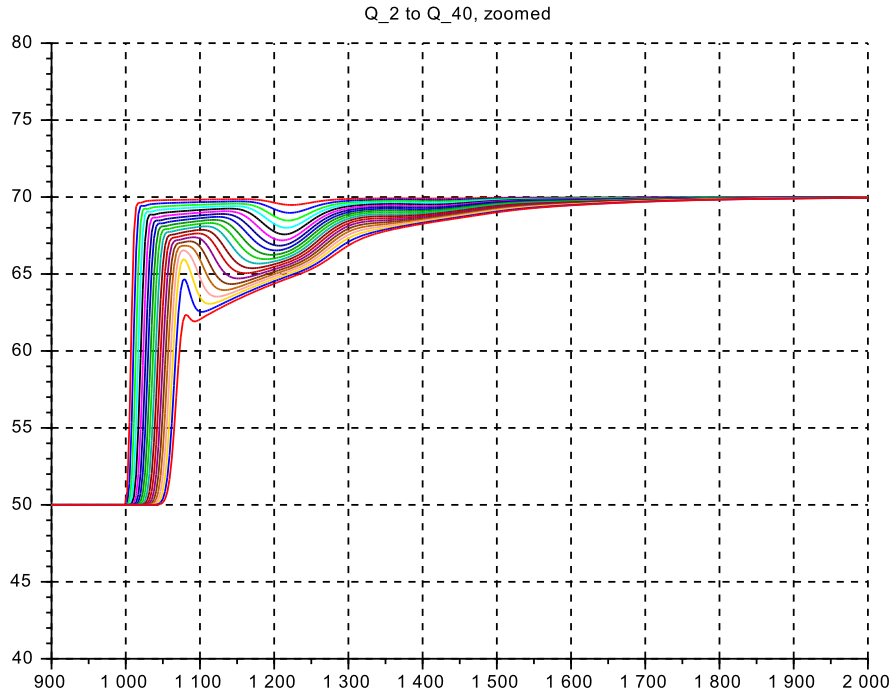
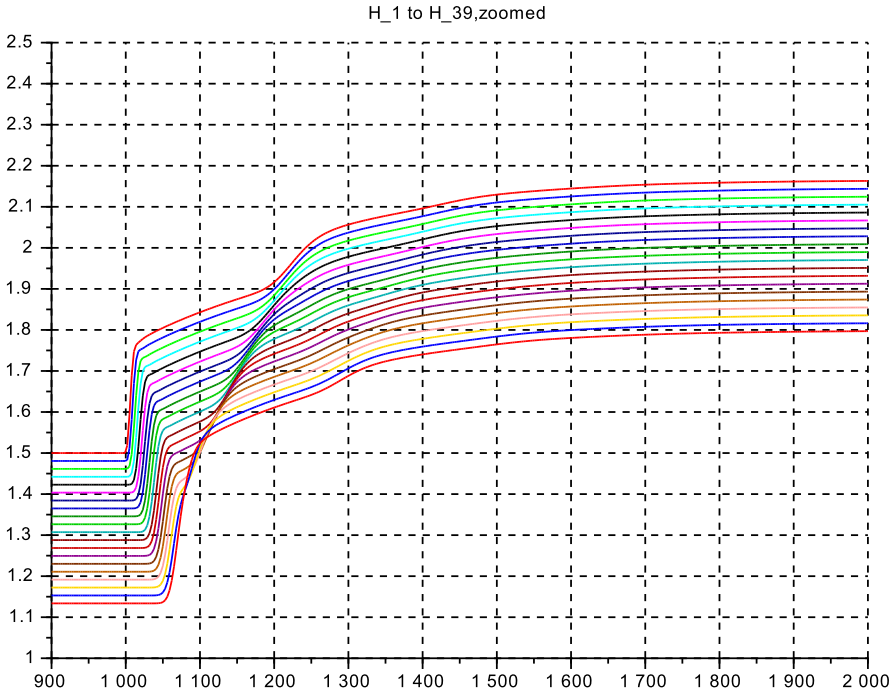


Figure 3.11: $N = 32$: Q, H for flow ramp-up at reference flow $Q = 1.00 * Q\_r$

Figure 3.12: $\mathbf{N = 20}$: Q, H for faster flow ramp-up at reference flow $Q = 1.00 * Q\_r$





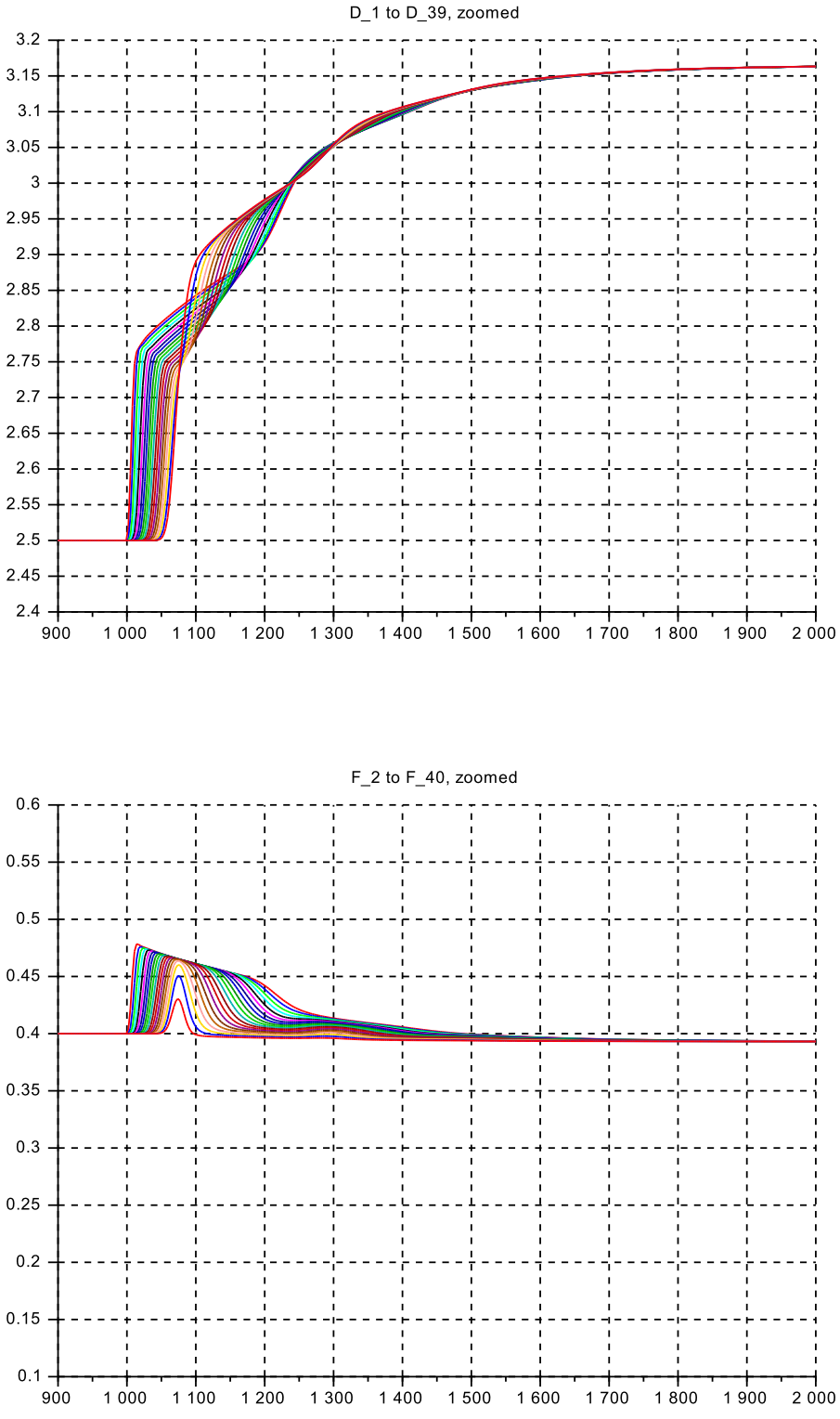


Figure 3.13: $\mathbf{N = 20}$: D, F for faster flow ramp-up at reference flow $Q = 1.00 * Q\_r$

3.5 Case s_c3_41_01_ : Low and High Flow Conditions

This is the case 41_00 taken for low flow $0.25 \cdot Q_r = 12.5\ m^3/s$ and $B(k) = B_r\ \forall\ k$, Fig.3.8 and 3.9, and high flow $8.0 \cdot Q_r = 400.\ m^3/s$, and where $B(k) = 4.0 \cdot B_r\ \forall\ k$ to model the flooding of lateral forelands, Fig.3.10 and 3.11. For both subcases however the bottom slope is not adjusted to the actual flow, but is carried over from the reference case, as it would be in the real world. Therefore the initial level slope needs to be adjusted to the actual flow which is performed in the '_context' file in the 'iteration loop' part.

.sce-files

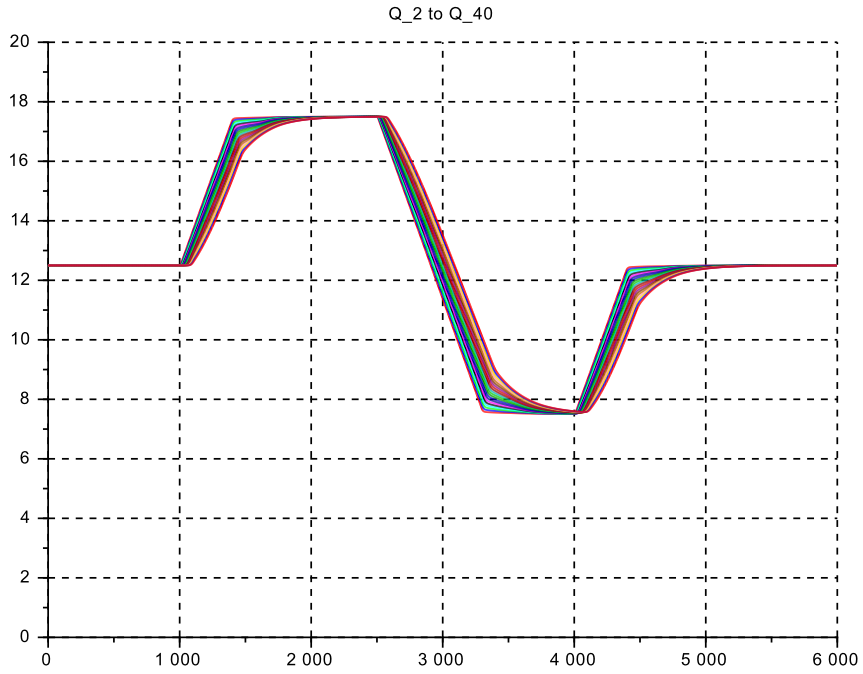
```
// s_c3_41_01_context
// Glf  06.05.14
// with 20 Segments, in 5 SuBlo4's
// no vertical dynamics

g = 10.;
L = 20.; kap = 1.0;

N= 20;      // no Volume and Momentum-segments

// flow set; choose by 'comment'
   Q_I = 400.0; B_I = 40.0; S_I = 1*(-1.0); tau_st = 4.0;
// Q_I =  12.5; B_I = 10.0; S_I = 1*(-1.0); tau_st = 80.0;

// GMS-friction coefficient
// k_s = 70.7;
   k_s = 50.0;
// k_s = 31.6;

// reference bottom slope
  Q_r = 50.0; B_r = 10.0;
  D_r =  2.5; S_r = 1*(-1.0);
  U_r =  2.0;
  R_r = (B_r*D_r)/(B_r + 2*D_r);
  I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// initialzing 'iteration loop'
D_I = D_r;
gainQ = 0.01*(D_I/Q_I);
Q = k_s*(B_I)*(D_I)*(D_I)^(2/3)*(I_r)^(1/2);
eQ  = -Q_I + Q;

// iteration loop
while (eQ<-0.001)|(eQ>0.001)  then
    R_I = (B_I*D_I)/(B_I + 2*D_I);
    Q   = (k_s*(I_r)^(1/2)*B_I)*D_I*(R_I)^(2/3);
    eQ  = -Q_I + Q;
    D_I = D_I - gainQ*eQ;
end

 D_min = +0.001; D_max = 40*D_r;
 Q_min = +0.001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb = 1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
//****************
vB = B_I*vb;
//******************

// basic layout: horizontal bottom
vs  = -1*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];

//*****************
vS0 = (-S_I)*vs;
//*************************


vd0 = ones(1,(2*N+1));
vD0 = D_I*vd0;

vq0 = ones(1,(2*N+1));
vQ0 = Q_I*vq0;

// Inflow data
    B_in = vB(1);
    S_in = vS0(1);
    D_in = vD0(1);
    Q_in = vQ0(1);

// friction slope of the bottom vS
vi0    = ones(1,(2*N+1));
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));

for kk=2:2:(2*N),
  vI0(kk)    = I_r*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end


//  outflow data
    B_o = vB(kk+1);
    S_o = vS(kk+1);
    D_o = vD0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
dQ = 0.40;
t_st_1 = 1000.0;  r_1_0 = Q_I; r_1_1=(1.0+1.*dQ)*Q_I;
t_st_2 = 2500.0;  r_2_0 = 0.;  r_2_1 = -2.*dQ*Q_I;
t_st_3 = 4000.0;  r_3_0 = 0.;  r_3_1 = +1.*dQ*Q_I;
T_fin = 6000.;
// inflow slew rate
g_st = 10.0; u_up_st = +1.0; u_dn_st = -1.0;
 Q_st_0 = Q_I;

// outflow fixed:
//Q_o = Q_0;

// outflow by GMS
g_o = k_s*((I_r)^(1/2));

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

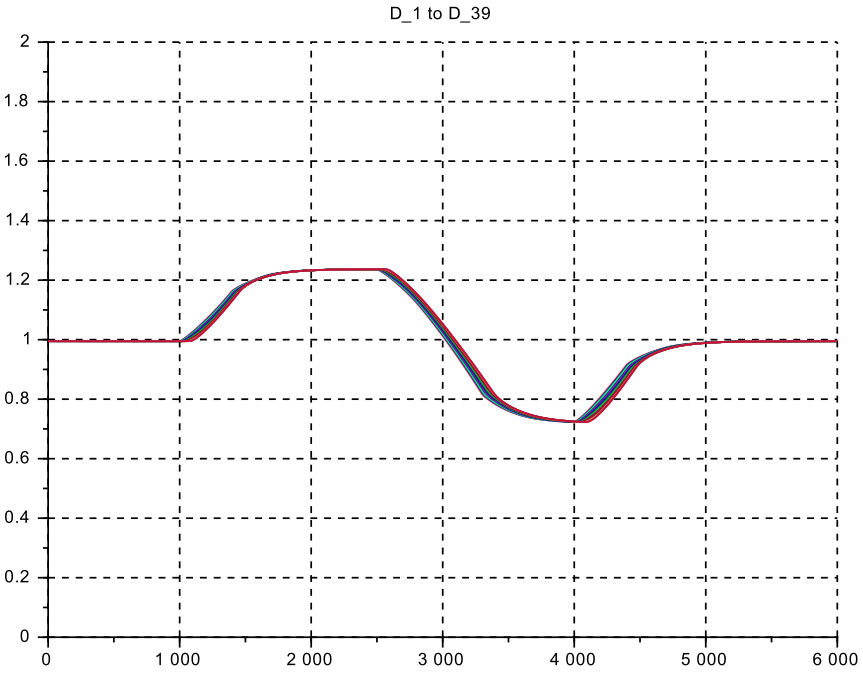
```
// s_c3_41_01_crunplot
//  Glf   26.10.2015
// no vertical dynamics

stacksize('max'); exec('s_c3_41_01_context.sce', -1);
importXcosDiagram('s_c3_41_01.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:8,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
   plot2d(Q.time,Q.values,vcolor,rect=[0,0,6000,600.]);
// plot2d(Q.time,Q.values,vcolor,rect=[0,0,6000,20.]);
xtitle("Q_2 to Q_40");
xgrid(1);

f2 = scf(2);
   plot2d(D.time,D.values,vcolor,rect=[0,1,6000,5.0]);
// plot2d(D.time,D.values,vcolor,rect=[0,0,6000,2.0]);
xtitle("D_1 to D_39");
xgrid(1);

f3 = scf(3);
   plot2d(H.time,H.values,vcolor,rect=[0,0,6000,4.0]);
// plot2d(H.time,H.values,vcolor,rect=[0,-1,6000,3.0]);
xtitle("H_1 to H_39");
xgrid(1);

f4 = scf(4);
   plot2d(F.time,F.values,vcolor,rect=[0,0.2,6000,0.6]);
xtitle("F_2 to F_40");
xgrid(1);

f5 = scf(5);
   plot2d(Q.time,Q.values,vcolor,rect=[900,300,2000,700.]);
// plot2d(Q.time,Q.values,vcolor,rect=[900,12,2000,18.]);
xtitle("Q_2 to Q_40, zoomed");
xgrid(1);

f6 = scf(6);
   plot2d(D.time,D.values,vcolor,rect=[900.,3.0,2000,4.5]);
// plot2d(D.time,D.values,vcolor,rect=[900.,0.9,2000,1.3]);
xtitle("D_1 to D_39, zoomed");
xgrid(1);

f7 = scf(7);
   plot2d(H.time,H.values,vcolor,rect=[900.,1.5,2000,3.5]);
// plot2d(H.time,H.values,vcolor,rect=[900.,-1.0,2000,1.0]);
xtitle("H_1 to H_39, zoomed");
xgrid(1);

f8 = scf(8);
   plot2d(F.time,F.values,vcolor,rect=[900.,0.2,2000,0.6]);
xtitle("F_2 to F_40, zoomed");
xgrid(1);
```

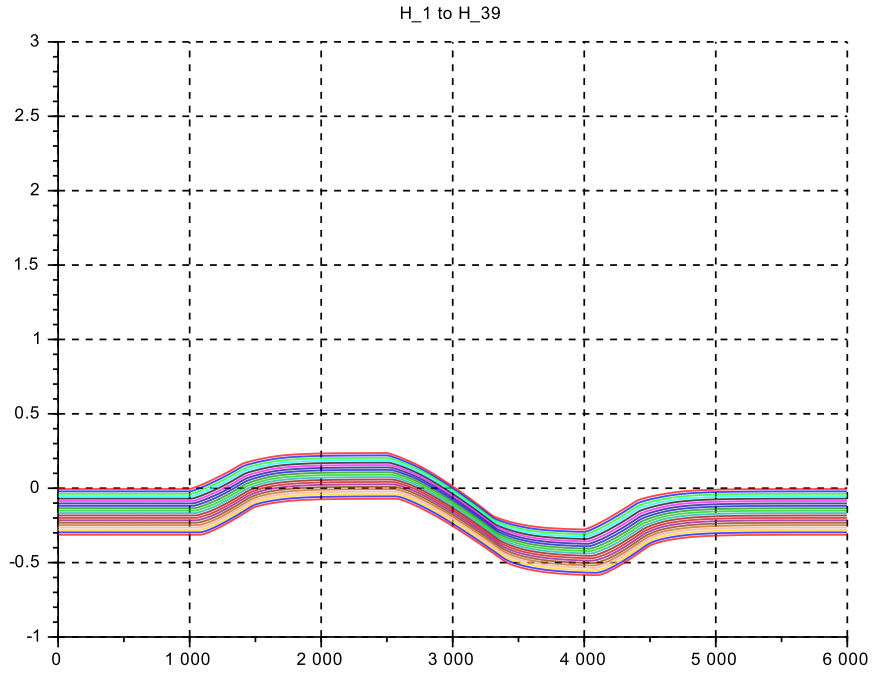
Case s_c3_41_01: Discussion

- For both sub-cases the transients are shown in full and as zoom-in of the first ramp-up, but reduced to $vQ(k)$ and $vD(k)$ only.
- The time delay from $vQ(2)$ to $vQ(40)$
  for *low* flow with $D \approx 1.0\ m$, that is $U \approx 1.25\ m/s$, and $U_F \approx 3.2\ m/s$, that is $F \approx 0.39$, is calculated at $\approx 85\ s$, whereas from the simulation $\approx 82\ s$;
  and for *high* flow with $D \approx 3.5\ m$, that is $U \approx 2.9\ m/s$, and $U_F \approx 5.9\ m/s$, that is $F \approx 0.49$, is calculated at $\approx 43\ s$, whereas from the simulation $\approx 41\ s$.
- The flow ramps are set as follows:
  for *low* flow, with $dQ = \pm 0.4 \cdot Q = 5.0\ m^3/s$ at $\pm 0.0125\ m^3/s\ /s$, that is 400 s for full stroke
  for *high* flow, with $dQ = \pm 0.40 \cdot Q = 160\ m^3/s$ at $\pm 0.25\ m^3/s\ /s$, that is 640 s for full stroke.
  Then for both sub-cases the reflected wave is weak.
- Again for faster flow ramps (not shown here), the reflected wave gets much more pronounced.

Low flow conditions, $Q = 0.25 * Q\_r$ and $B = 1 * B_r$







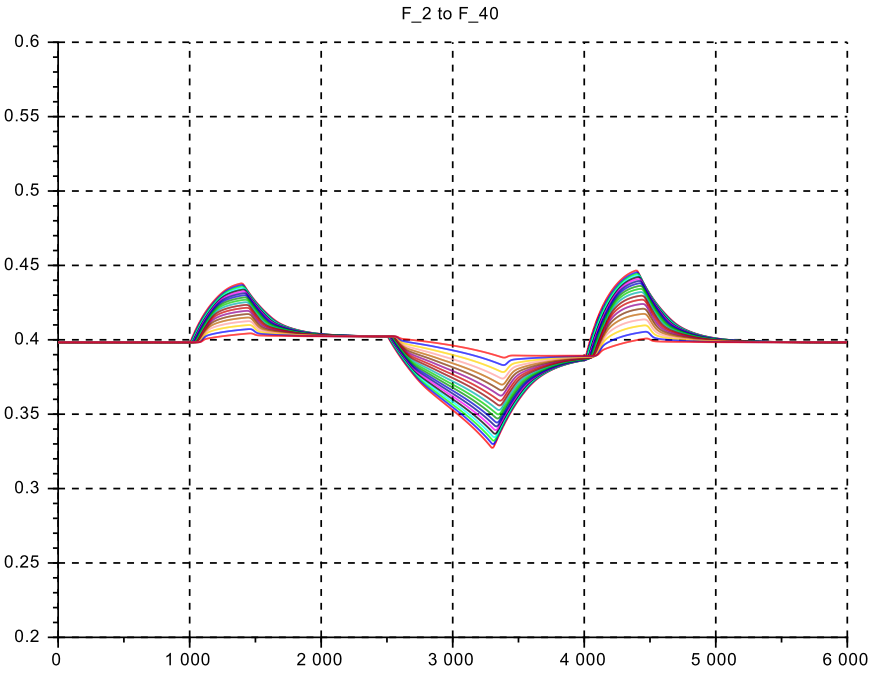
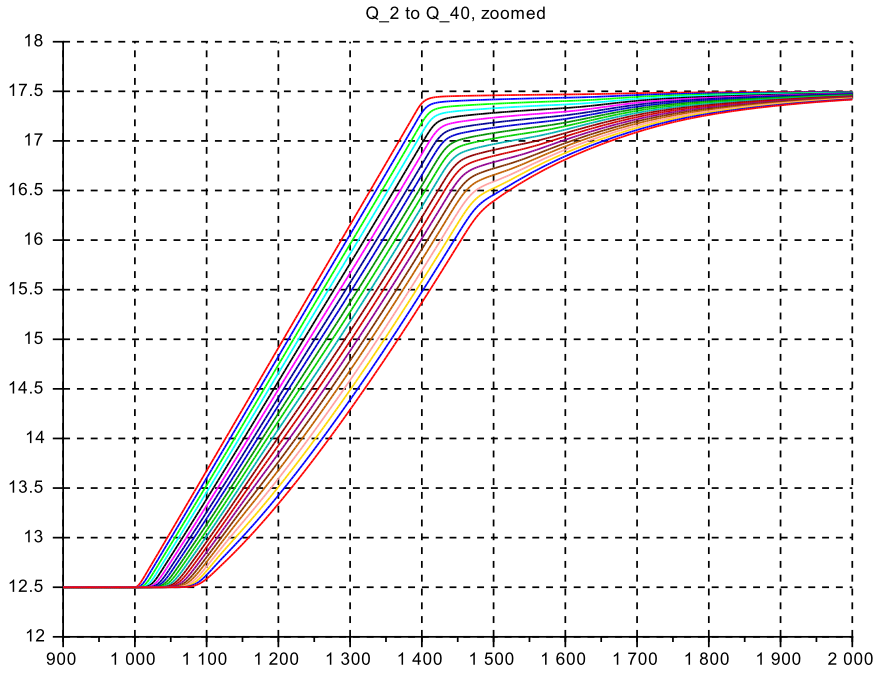
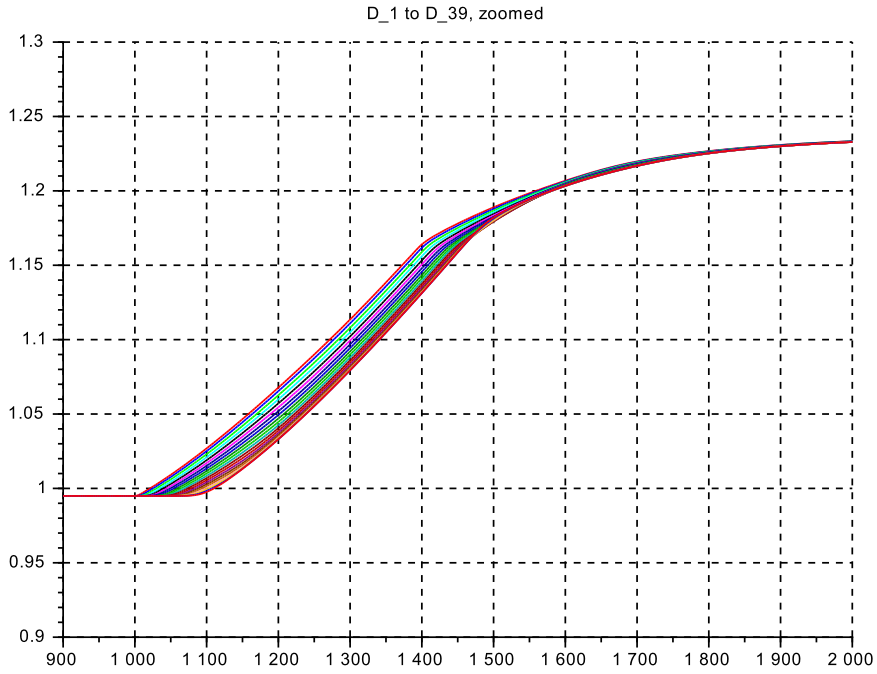


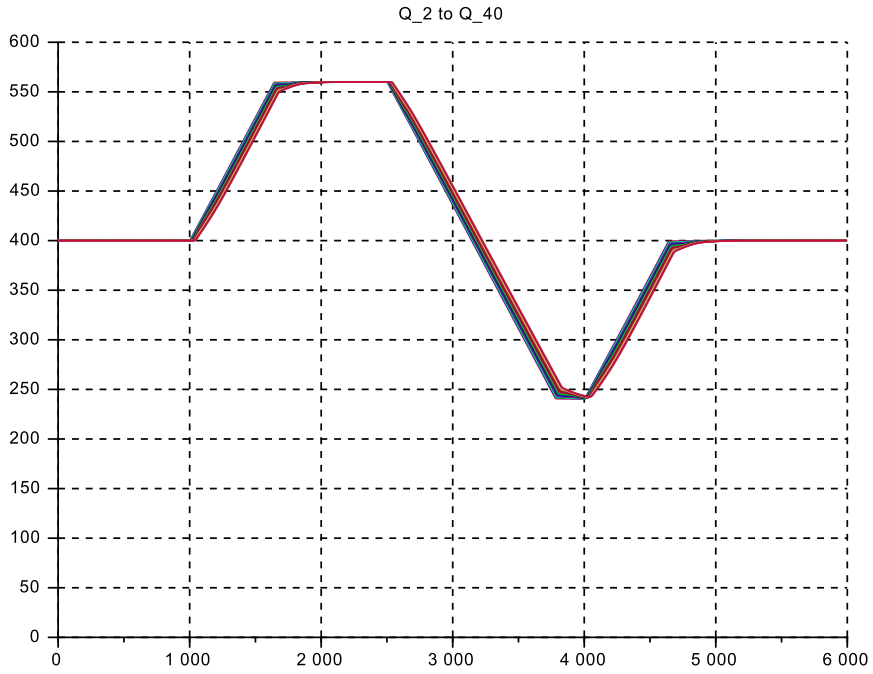
Figure 3.14: *s_ce_41_01_* for flow variations at LOW flow, $Q = 0.25 * Q\_r$

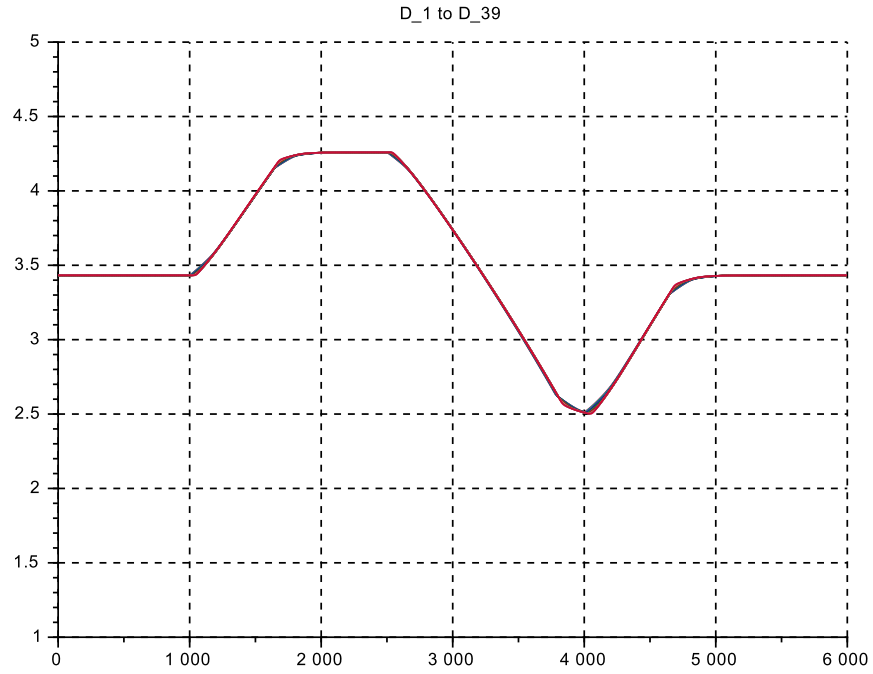
Figure 3.15: $s\_c3\_41\_01\_$ at $Q = 0.25 * Q\_r$, zoom-in on first step up

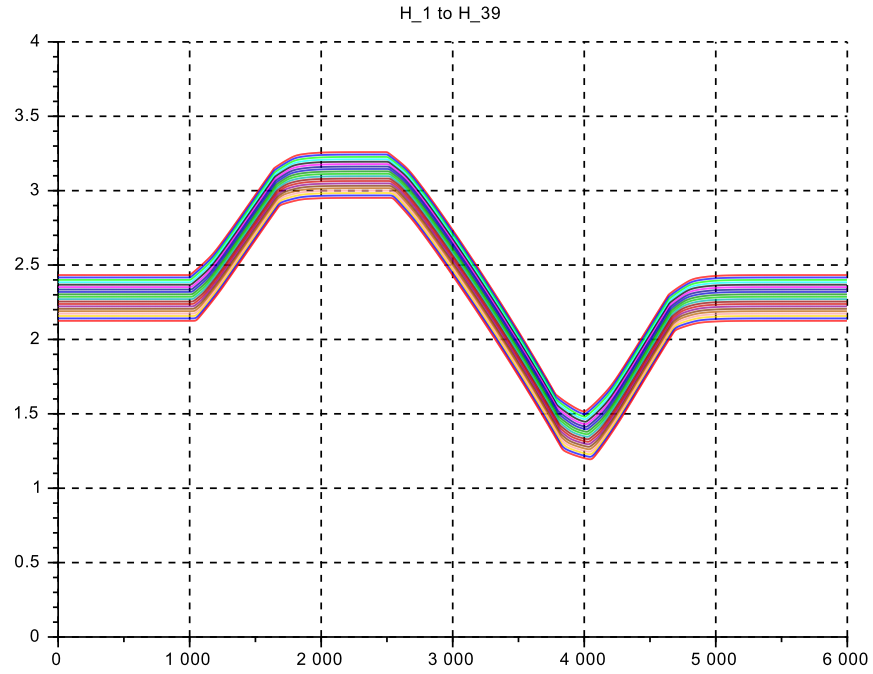




High flow conditions, $Q = 8.00 * Q\_r$ and $B = 4.0 * B_r$







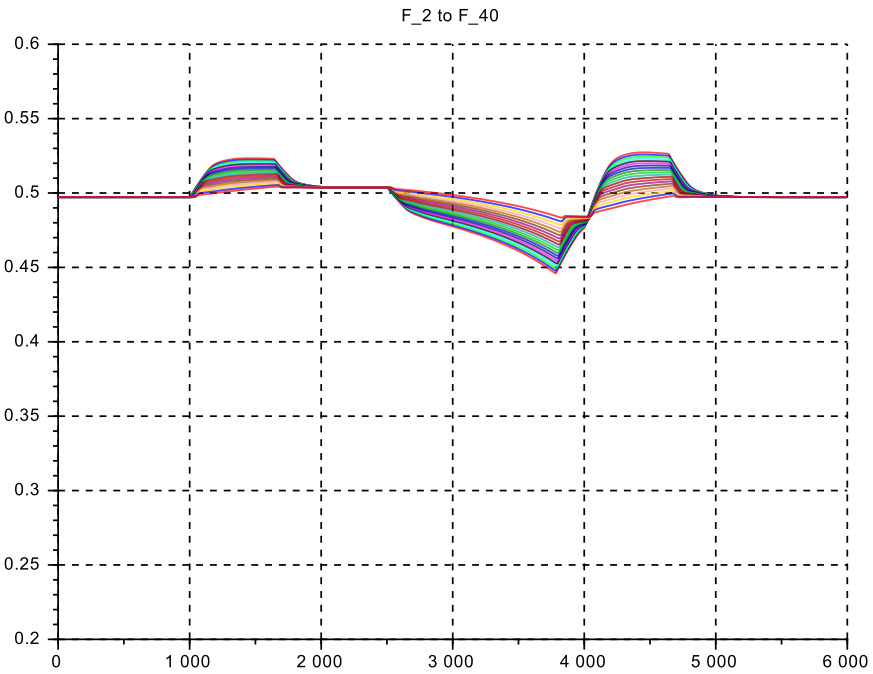
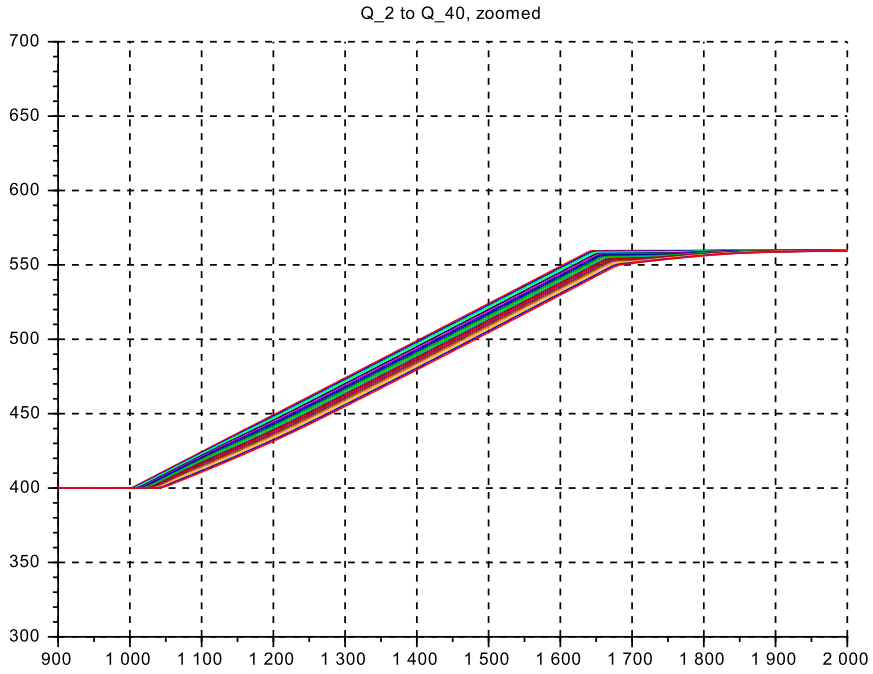
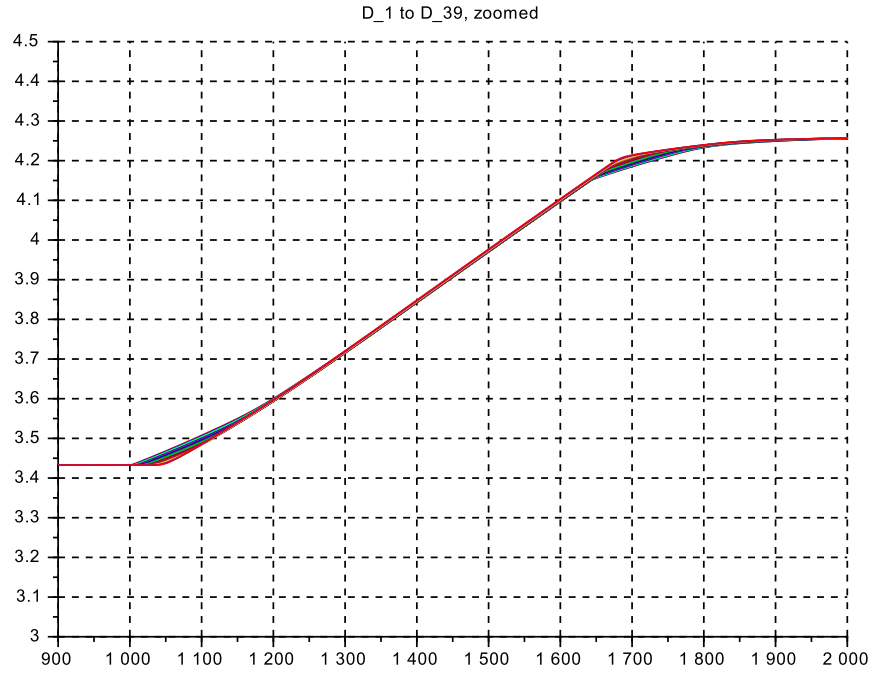


Figure 3.16: $s\_c3\_41\_01\_$ for flow variations at HIGH flow $Q = 8.00 * Q\_r$ and $B = 4 * B_r$

Figure 3.17: 41_01_ for flow variations at HIGH flow $Q = 8.00 * Q\_r$ and $B = 4 * B_r$, zoom-in





3.6 Case s_c3_41_02_ : Set of Froude numbers

In the basic case 41_00 the Froude number follows from the reference condition set as $F_r = 0.40$ which is a common value in river and channel flows. Here the Froude number $F := F_0$ will be a given, $\neq 0.40$, while width $B = B_r = 10\ m$ and flow $Q = Q_r = 50\ m^3/s$ are kept at their reference values. Therefore water depth $D := D_0$ and velocity $U := U_0$ are now a function of F, and have to be computed in the '_context'-file first, see below.
In order to keep flow $vQ(k+1)$ and depth $vD(k)$ at steady state conditions along k as in the previous cases, while the Strickler coefficient is kept at its realistic value $k_s = 50.$, then the bottom slope must be inclined according to the actual flow velocity U, and thus is to be computed in the '_context'-file as well.

.sce-files

```
// s_c3_41_02_context
// Glf  6.5.14
// with 20 Segments, in 5 SuBlo4's
// no vertical dynamics

g = 10.;

// reference op-point
Q_r = 50.0; H_r = 1.5; S_r = 1.0*(-1.0);
D_r = H_r - S_r; B_r = 10.; U_r = 2.0; F_r = 0.40;

  lambda = 8.00; L = lambda*D_r;  kap = 1.0;

N= 20;      // no Volume+momentum-segments

// Operating point, by Froude-number
//F_0 = 0.1;
  F_0 = 1.6;


// select actual flow op-point
   Q_0 = 1.00*Q_r;

 D_0 = ((1/g)*((Q_0/(B_r*F_0))^2))^(1/3);
 //*************************************
 H_0 = H_r; S_0 = S_r; B_0 = B_r; U_0 = U_r;

 D_min = +0.001; D_max = 40*D_r;
 Q_min = +0.001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb = 1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
//****************
vB = B_0*vb;
//******************

// basic layout: horizontal bottom
vs  = -1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
//*****************
vS0 = 1.0*vs;
//*************************

vd0 = ones(1,(2*N+1)); vD0 = D_0*vd0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in = vB(1); S_in = vS0(1); D_in = vD0(1); Q_in = vQ0(1);

// GMS-coefficient
// k_s = 100.;
// k_s = 70.7;
   k_s = 50.0;
// k_s = 31.6;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));

for kk=2:2:(2*N),
  vRtilda(kk)=((vB(kk)*vD0(kk))/(vB(kk)+2*vD0(kk)))^(2/3);
  vI0(kk) = (vQ0(kk)/(vB(kk)*vD0(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); Q_o=vQ0(kk+1);

// inflow generation
dQ = 0.40;
t_st_1 = 1000.0;  r_1_0 = Q_0; r_1_1=(1.0+1.*dQ)*Q_0;
t_st_2 = 2500.0;  r_2_0 = 0.;  r_2_1 = -2.*dQ*Q_0;
t_st_3 = 4000.0;  r_3_0 = 0.;  r_3_1 = +1.*dQ*Q_0;
T_fin = 6000.;
// inflow slew rate
g_st = 2.0; u_up_st = +1.0; u_dn_st = -1.0;
 tau_st = 20.0; Q_st_0 = Q_0;

// outflow fixed:
//Q_o = Q_0;

// outflow by GMS
g_o = k_s*(vI0(40))^(1/2);

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

```
// s_c3_41_02_crunplot
//  Glf  06.05.14 26.10.2015
// no vertical dynamics

stacksize('max'); exec('s_c3_41_02_context.sce', -1);
importXcosDiagram('s_c3_41_02.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:8,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
   plot2d(Q.time,Q.values,vcolor,rect=[0,0,6000,80.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
// plot2d(D.time,D.values,vcolor,rect=[0,3,6000,10.00]);
   plot2d(D.time,D.values,vcolor,rect=[0,0,6000,7.00]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
// plot2d(H.time, H.values,vcolor,rect=[0.,2.0,6000,9.0]);
   plot2d(H.time, H.values,vcolor,rect=[0.,-6.0,6000,1.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
// plot2d(F.time, F.values,vcolor,rect=[0.,0.,6000,0.2]);
   plot2d(F.time, F.values,vcolor,rect=[0.,0.,6000,2.0]);
xtitle("F_2 to F_40"); xgrid(1);

f5 = scf(5);
// plot2d(Q.time,Q.values,vcolor,rect=[900,30,2400,80.]);
   plot2d(Q.time,Q.values,vcolor,rect=[900,30,2400,80.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f6 = scf(6);
// plot2d(D.time,D.values,vcolor,rect=[900,6,2400,9.0]);
   plot2d(D.time,D.values,vcolor,rect=[900,0.8,2400,1.4]);
xtitle("D_1 to D_39"); xgrid(1);

f7 = scf(7);
// plot2d(H.time,H.values,vcolor,rect=[900.,5.0,2400,8]);
   plot2d(H.time,H.values,vcolor,rect=[900,-6.0,2400,1]);
xtitle("H_1 to H_39"); xgrid(1);

f8 = scf(8);
// plot2d(F.time,F.values,vcolor,rect=[900,0.0,2400,0.2]);
   plot2d(F.time,F.values,vcolor,rect=[900,1.3,2400,1.8]);
xtitle("F_2 to F_40"); xgrid(1);
```

Case s_c3_41_02: Discussion

- In the context file D_0 is calculated

$$F\_0^2 = \frac{U\_0^2}{gD\_0} \quad \rightarrow \quad D = \frac{U\_0^2}{g\ F\_0^2} \ = \ \frac{1}{gF\_0^2} \cdot \left(\frac{Q\_0}{B\_0 \cdot D\_0} \right)^2 \quad \rightarrow \quad D\_0^3 = \frac{1}{g} \cdot \left(\frac{Q\_0}{B\_0 \cdot F\_0} \right)^2$$

- For both sub-cases the transients are shown only as zoom-in's of the first ramp-up, for $vQ(k)$, $vD(k)$, $vH(k)$ and $vF(k)$.

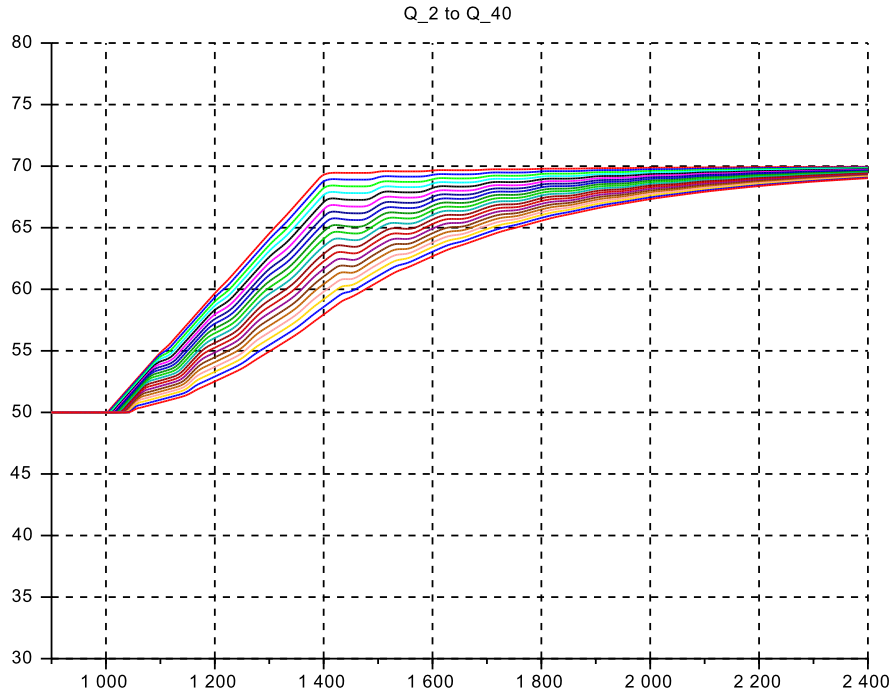
- The time delay from $vQ(2)$ to $vQ(40)$
  for the *low* Froude number $F = 0.10$ with $D \approx 6.3\ m$, that is $U \approx 0.8\ m/s$ and $U_F \approx 7.9\ m/s$, that is $F \approx 0.10$, is calculated at $\approx 44\ s$, whereas from the simulation $\approx 41\ s$;
  and for the *high* Froude number $F = 1.6$ (supercritical flow) with $D \approx 1.0\ m$, that is $U \approx 5.0\ m/s$ and $U_F \approx 3.15\ m/s$, that is $F \approx 1.6$, is calculated at $\approx 47\ s$, whereas from the simulation $\approx 44\ s$.
  Again this fits nicely.

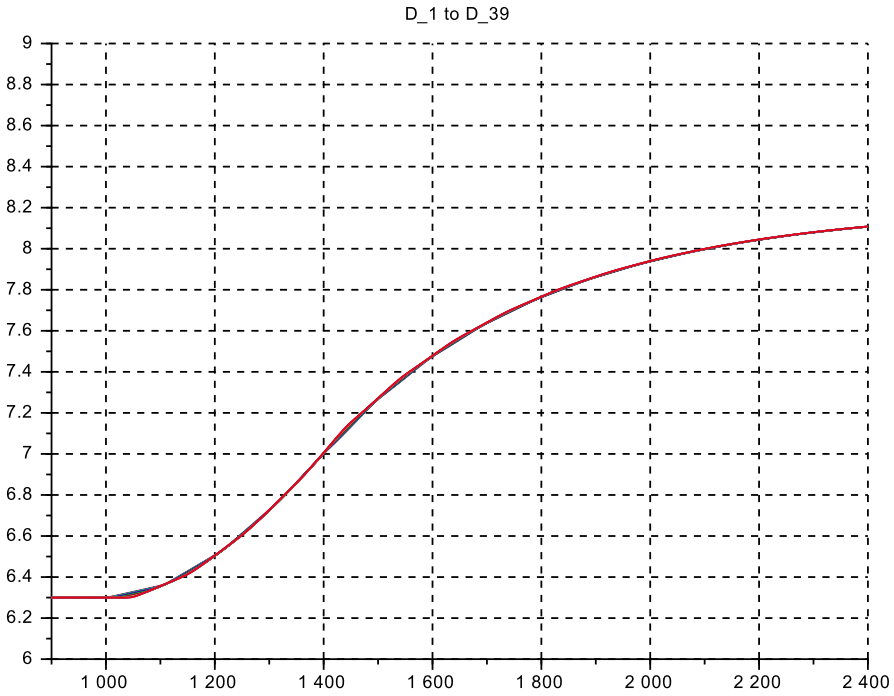
- The flow ramp is set to (with $dQ = \pm 0.4 \cdot Q = 20.0\ m^3/s$) at $\pm 0.050\ m^3/s$ per s, that is $400\ s$ for full stroke
  Then for both sub-cases the wave reflection is weak.

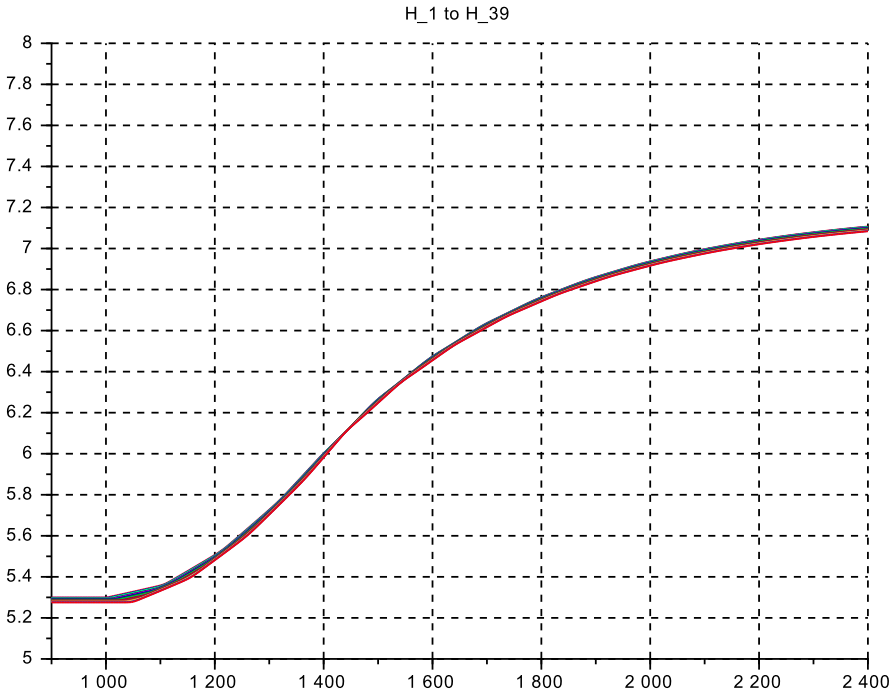
- For the low Froude number sub-case, weakly damped reflected wave oscillations are visible both after $t = 1000\ s$ and after $t = 1400\ s$
  They disappear for the high Froude number sub-case. – This is to be expected as the waves can no longer travel upstream on a supercritical flow.

- Again for faster flow ramps (simulations not shown) the reflected waves are much more pronounced.

Figure 3.18: $s\_c3\_41\_02\_$ for flow variation at $Q = 1.00 * Q\_r$ and LOW Froude number $F = 0.10$, zoom-in






H_1 to H_39
8
7.8
7.6
7.4
7.2
7
6.8
6.6
6.4
6.2
6
5.8
5.6
5.4
5.2
5
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

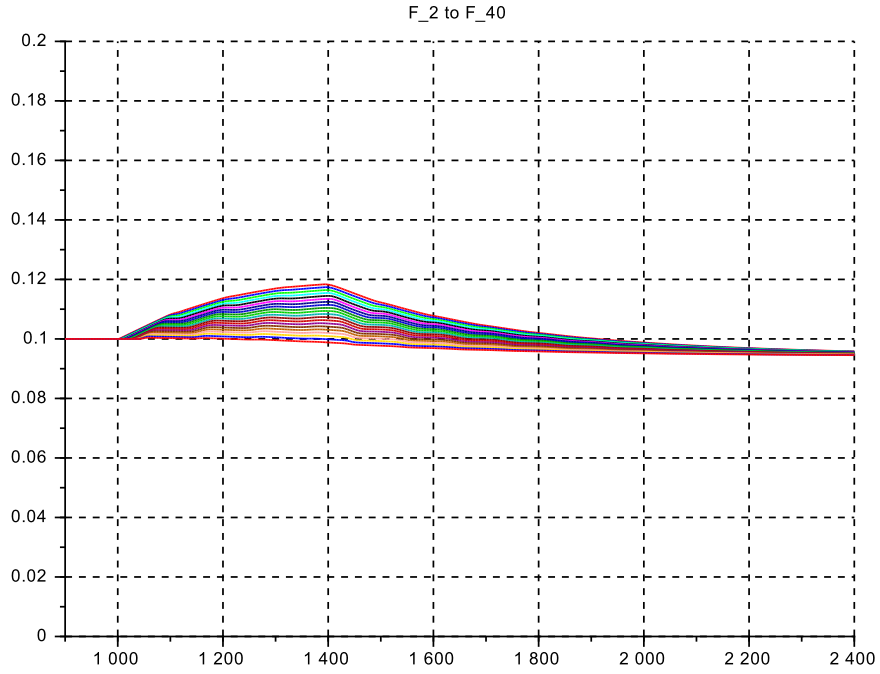
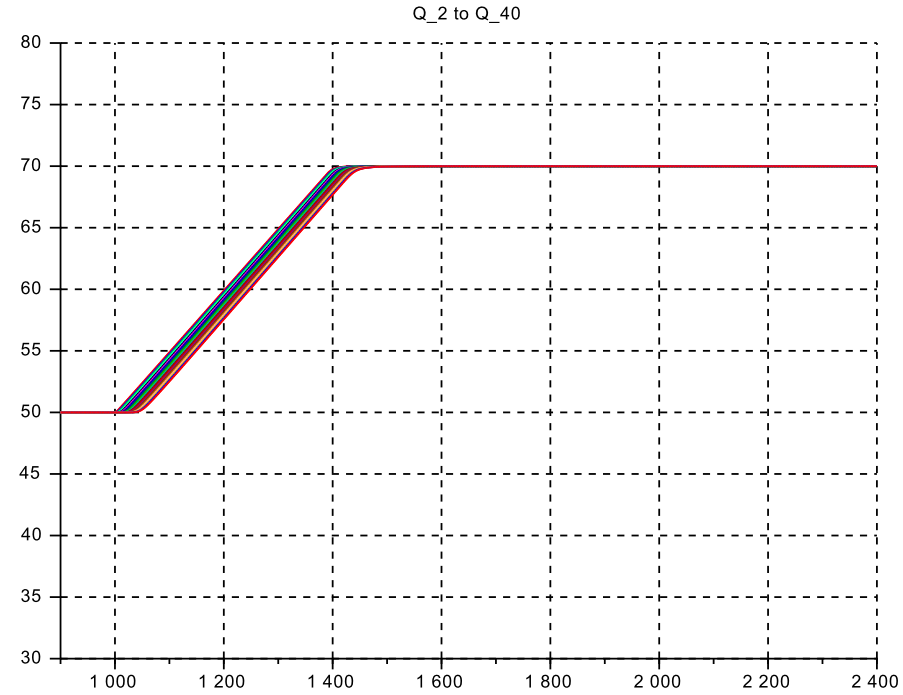
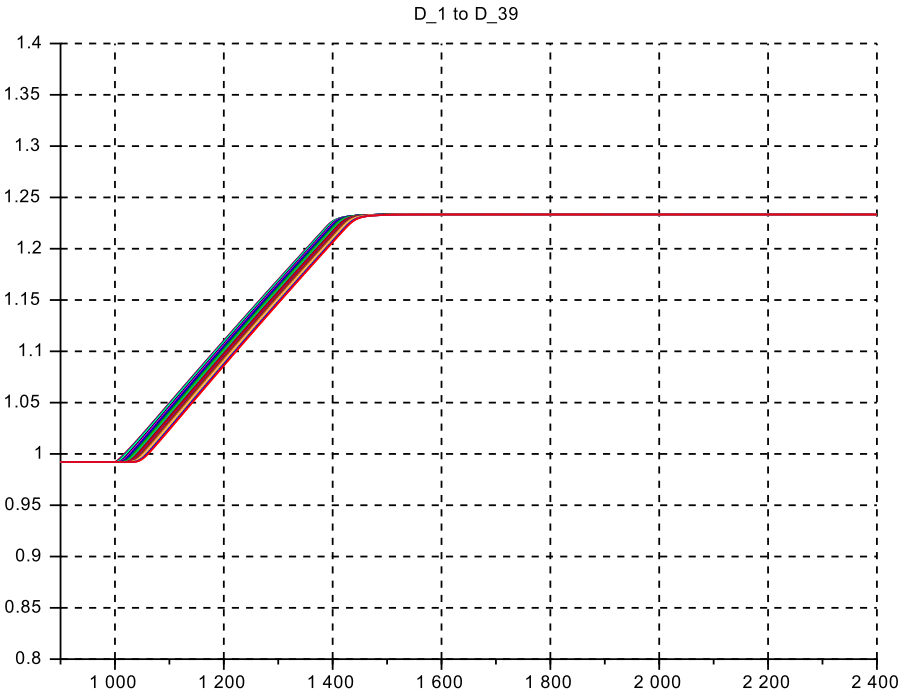
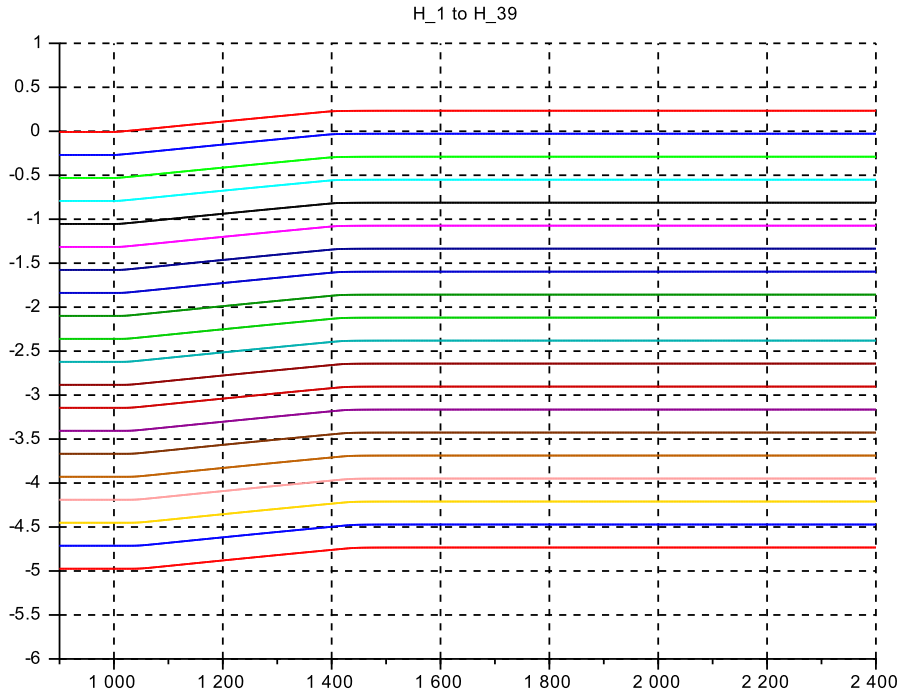
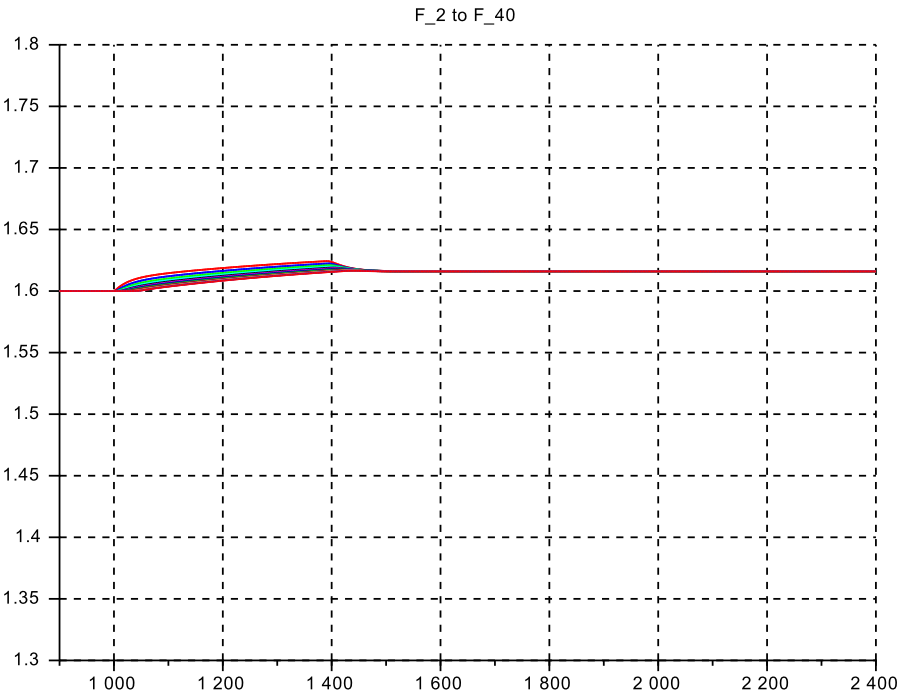

F_2 to F_40
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

Figure 3.19: $s\_c3\_41\_02\_$ for flow variation at $Q = 1.00 * Q\_r$ and HIGH Froude number $F = 1.60$, zoom-in






H_1 to H_39
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
-5
-5.5
-6
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400


F_2 to F_40
1.8
1.75
1.7
1.65
1.6
1.55
1.5
1.45
1.4
1.35
1.3
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

3.7 Case s_c3_41_03_ : Conical Cross Sections

3.7.1 Sub-case s_c3_41_03_1 : 'Confusor' Geometry

The layout with the GMS-outflow boundary is modified to a 'confusor' geometry, where the first ≈ 10 % of length are at $vB(k) = 5.0\ B_r$ and $vS(k) = -3.5\ S_r$ (that is the initial depth is $\approx 5.0\ m$). Then both $vB(k)$ and $vS(k)$ are reduced linearly down to their reference values at ≈ 60 % of length. From there the final part has constant cross section at reference values.

.sce-files

```
// s_c3_41_03_1_context
// Glf 26.10.2015
// no vertical dynamics
// confusor

g = 10.;
// reference op-point
Q_r = 50.0; H_r = 1.5; S_r = 1.0*(-1.0);
D_r = H_r - S_r; B_r = 10.; U_r = 2.0; F_r = 0.40;

lambda = 8.00; L = lambda*D_r;

kap = 1.0;

N= 20; // number of volume+momentum-segments
   F_0 = 0.40;

// select actual op-point
   Q_0 = 1.00*Q_r;

H_0=H_r; S_0=S_r; B_0=B_r; U_0=U_r; D_0=D_r;

 D_min = +0.001; D_max = 40*D_r;
 Q_min = +0.001; Q_max = 40*Q_r;

// channel geometry
//*****************
// confusor
  vb=[5.0, 5.0, 5.0, 5.0, 5.0, 4.8, 4.6, 4.4,...
      4.2, 4.0, 3.8, 3.6, 3.4, 3.2, 3.0, 2.8,...
      2.6, 2.4, 2.2, 2.0, 1.8, 1.6, 1.4, 1.2,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*vb;
//******************
//confusor
vs=-1.0*[3.5, 3.5, 3.5, 3.5, 3.5, 3.3, 3.1, 2.9, ...
         2.7, 2.5, 2.4, 2.3, 2.2, 2.1, 2.0, 1.9,...
         1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1.2, 1.1,...
         1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
         1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = 1.0*vs;
//*************************

// initial conditions on integrators
vh0 = ones(1,(2*N+1)); vH0 = H_0*vh0; vD0 = vH0 - vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// GMS-coefficient
   k_s = 50.0;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = 1*vD0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
dQ = 0.40;
t_st_1 = 1200.0;  r_1_0 = Q_0; r_1_1=(1.0+1.*dQ)*Q_0;
t_st_2 = 2700.0;  r_2_0 = 0.;  r_2_1 = -2.*dQ*Q_0;
t_st_3 = 4200.0;  r_3_0 = 0.;  r_3_1 = +1.*dQ*Q_0;
T_fin = 6000.;
// inflow slew rate
g_st = 1.0; u_up_st = +5.0; u_dn_st = -5.0;
tau_st = 10.0; Q_st_0 = Q_0;

// outflow by GMS
dS_40 = vS0(39) - vS0(41);
if dS_40 >= 0...
  then g_o = 1.00*k_s*((1*vIr(40) + 0.5*(dS_40/L))^(1/2));
  else g_o = 1.00*k_s*((vIr(40) - (dS_40/L))^(1/2));
end;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 3000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

```
// s_c3_41_03_1_crunplot
//  Glf  11.4.14, 26.10.2015
// no vertical dynamics
// confusor

stacksize('max');exec('s_c3_41_03_1_context.sce', -1);
importXcosDiagram('s_c3_41_03_1.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:10,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[1000.,20.0,6000,80.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[1000.,1.0,6000,7.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[1000.,0.0,6000,3.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[1000.,0.,6000,0.6]);
xtitle("F_2 to F_40"); xgrid(1);

//*******************************************************
f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[1000.,20.0,2500,80.]);
xtitle("Q_2 to Q_40, zoom-in"); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[1000.,1.0,2500,7.0]);
xtitle("D_1 to D_39, zoom-in"); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[1000.,0.0,2500,3.0]);
xtitle("H_1 to H_39, zoom-in"); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[1000.,0.,2500,0.6]);
xtitle("F_2 to F_40, zoom-in"); xgrid(1);
//*******************************************************
eX = L*(1:1:N); eXend = 21*L;

f9 = scf(9); yD =D.values; yH = H.values;
oD = yD(2950,:); oH = yH(2950,:);
vcolor=[2,5];
plot2d(eX',[oD',oH'],vcolor,rect=[0.,0.5,eXend,+5.5]);
xtitle("Longitudinal D (blue) and H (red) at 5900 s ");
xgrid(1);

f10 = scf(10); yF =F.values;  oF = yF(2950,:);
plot2d(eX', [oF'],  rect=[0.,0.0,eXend,+0.6]);
xtitle("Longitudinal Froude Number at 5900 s ");
xgrid(1);
```

Case s_c3_41_03_1: Discussion

- Figure 3.20 shows the full transients, Fig.3.21 zooms in on the first flow ramp-up, and Fig.3.22 shows the longitudinal profiles of $vD(k)$, $vH(k)$ and $vF(k)$ at $t = 5900\ s$.

- Note that $Q(t)$ near the inflow position rises rapidly, but quite slowly near the exit. This is due to the large surface generated by $vB(k)$, which has to be filled to the steady state level.

- Oscillations on flow and level are clearly visible. The largest amplitudes appear at $k = 9 \ldots 15$, and decrease in both the upflow and downflow directions.

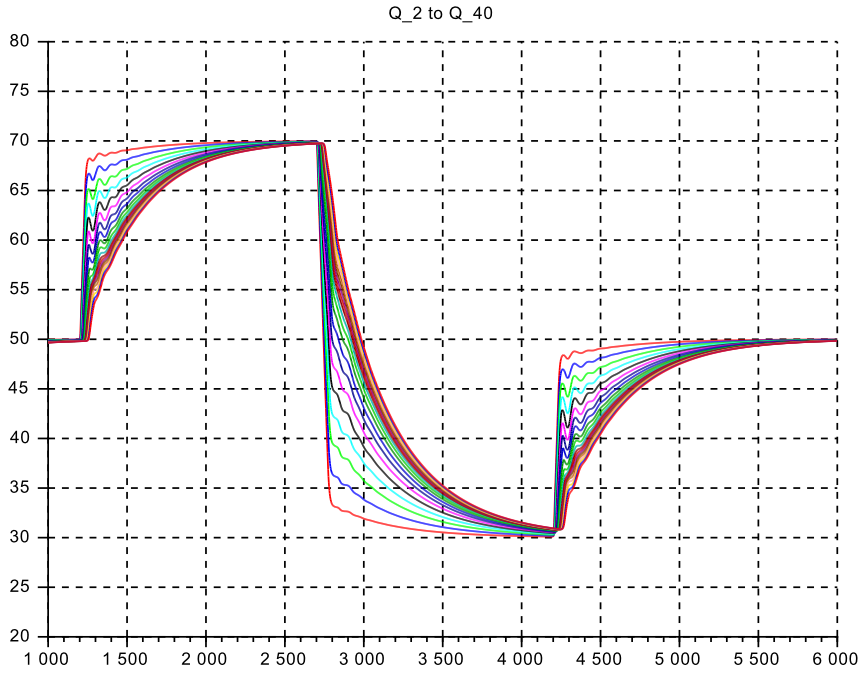
- From the longitudinal profile on $vH(k)$ a steady state level difference $\Delta H_m = vH(1) - vH(27) \approx 0.25\ m$ is measured. - Calculating the dynamic pressure loss due to the acceleration is

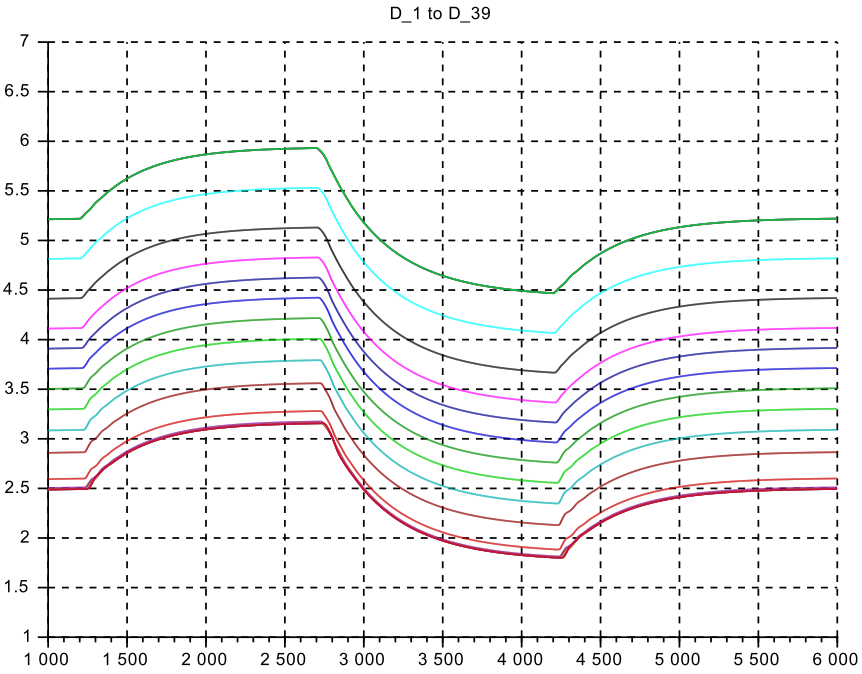
$$\Delta H_c \approx \frac{U_r^2}{2\ g} \approx \frac{4}{20} = 0.20\ m$$

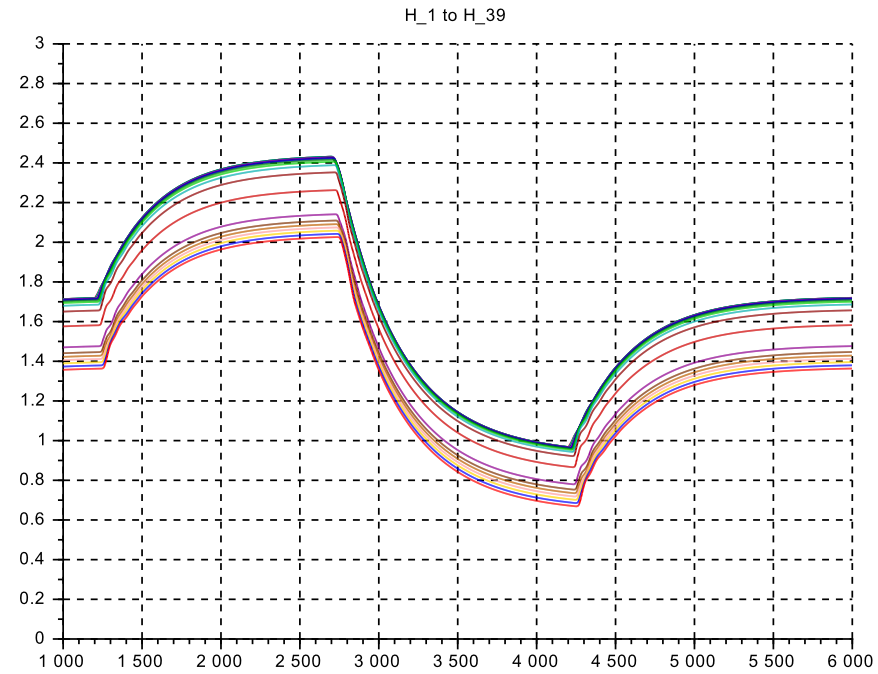
  The difference can be explained by the friction loss which is not included in ΔH_c.

- The Froude number at the inflow is measured at $vF(2)_m \approx 0.026$. - The calculation yields from $vD(1) \approx 5.2\ m$ and $vB(1) = 50\ m$ a velocity $vU(2) \approx 0.0192\ m/s$, and with $vU_F(2) = \sqrt{52} = 7.21\ m/s$ and thus $vF(2)_c \approx 0.027$. // The Froude number at the outflow reaches 0.40, as expected.

Figure 3.20: $s\_c3\_41\_03\_1\_$ confusor geometry for flow variations at $Q = 1.0 \cdot Q\_r$






H_1 to H_39
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 000
1 500
2 000
2 500
3 000
3 500
4 000
4 500
5 000
5 500
6 000

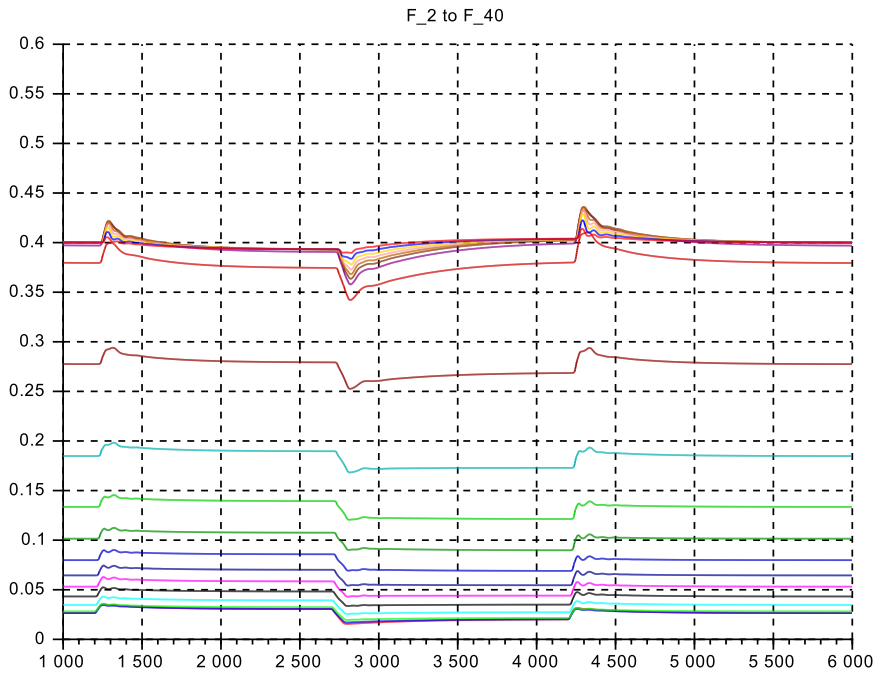
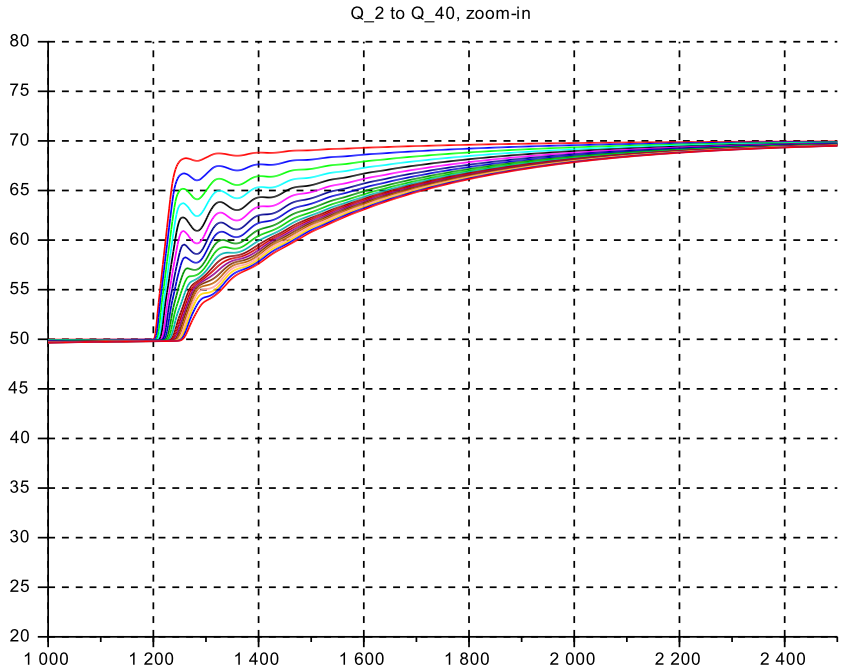
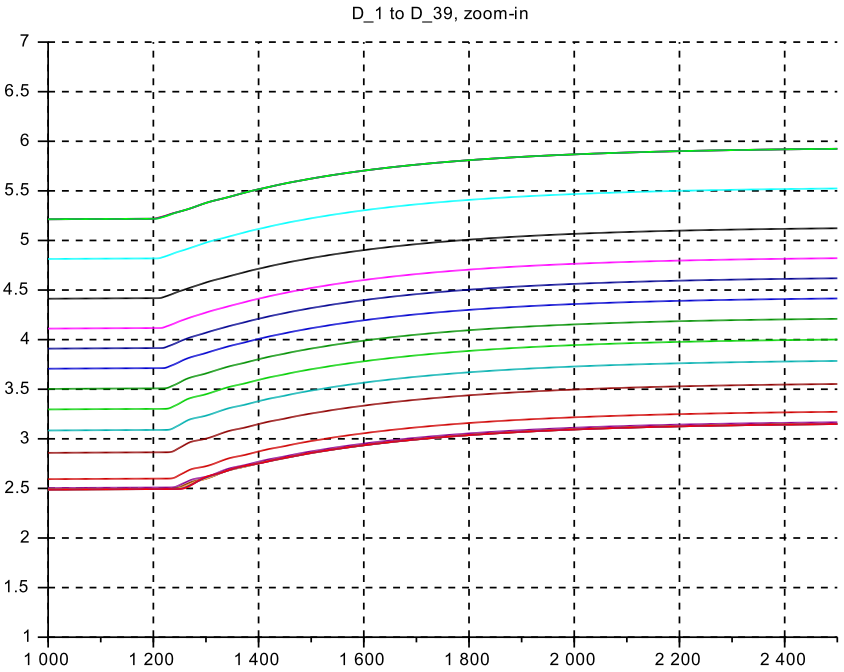
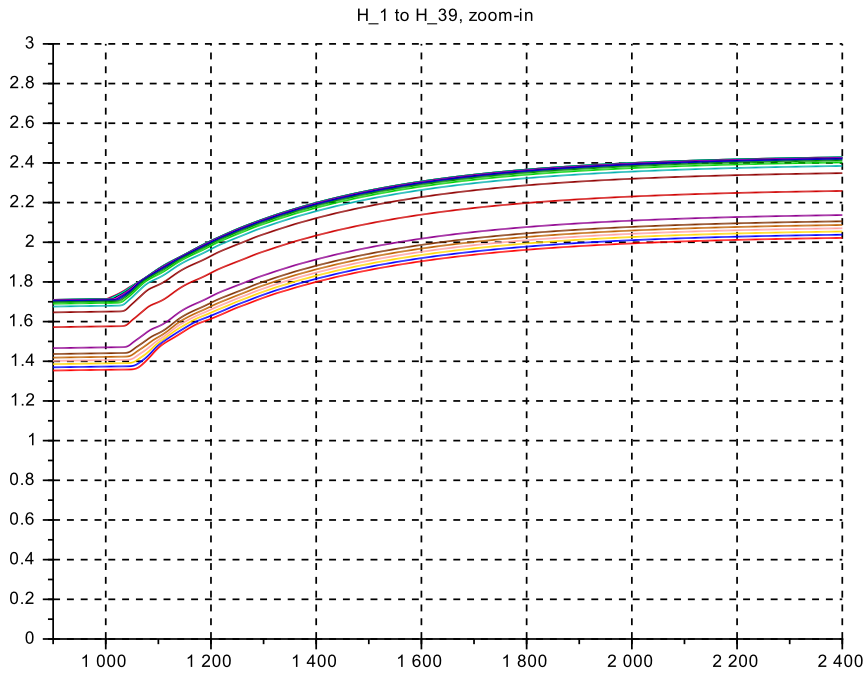

F_2 to F_40
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 000
1 500
2 000
2 500
3 000
3 500
4 000
4 500
5 000
5 500
6 000

Figure 3.21: $s\_c3\_41\_03\_1\_$ confusor geometry for flow variation, zoom-in






H_1 to H_39, zoom-in
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

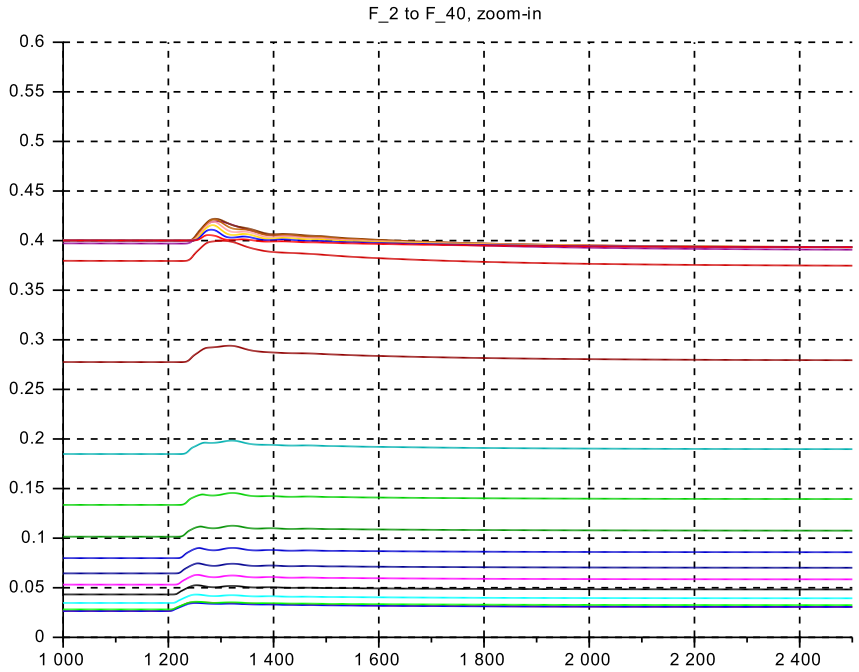
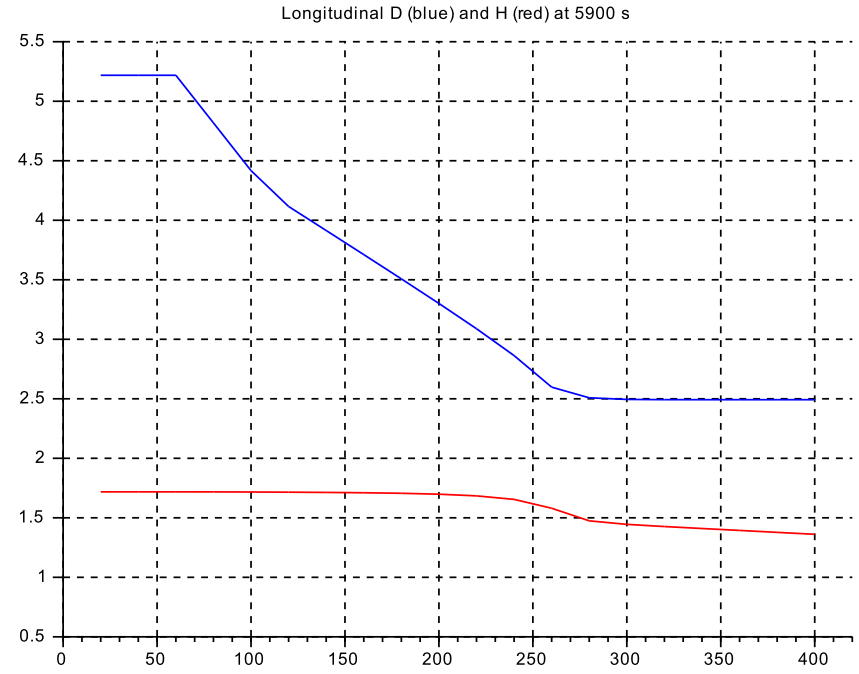
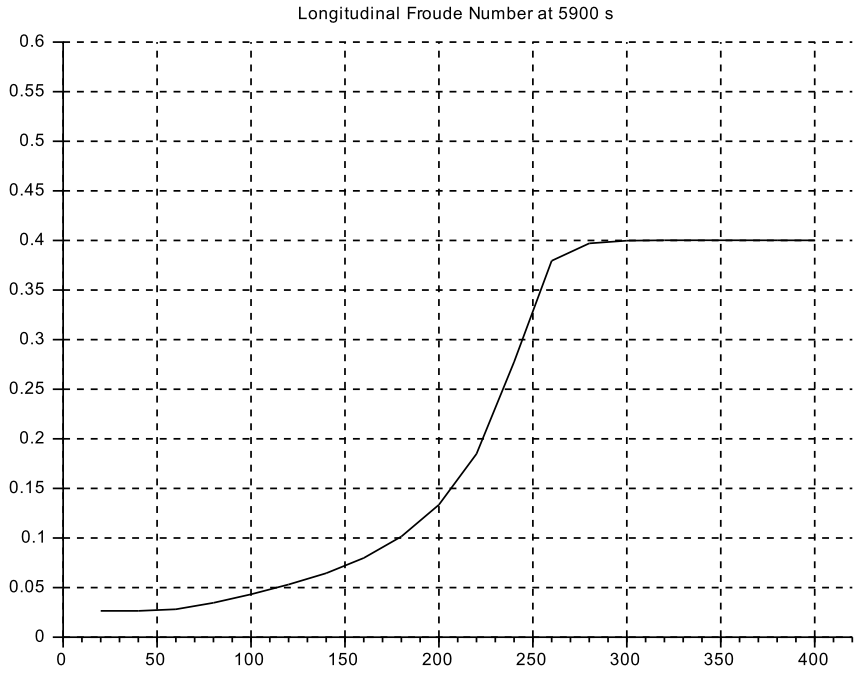

F_2 to F_40, zoom-in
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

Figure 3.22: $s\_c3\_41\_03\_1\_$ longitudinal profiles for D, H and F at $t = 5900\ s$





3.7.2 Sub-case s_c3_41_03_2 : 'Diffusor' Geometry

The geometry is set to yield a 'diffusor' shape: Initially the cross section up to $\approx$ 40% of length is constant at reference values. Then the width $vB(k)$ expands linearly up to 50 m and the bottom $vS(k)$ slopes down linearly to 3.0 m at $\approx$ 90 % of length. The final part has constant cross section down to the outflow.

.sce-files

```
// s_c3_41_03_2_context
// Glf  26.10.2015
// no vertical dynamics
// diffusor

g = 10.;
// reference op-point
Q_r = 50.0; H_r = 1.5; S_r = 1.0*(-1.0);
D_r = H_r - S_r; B_r = 10.; U_r = 2.0; F_r = 0.40;

  lambda = 8.00; L = lambda*D_r;

kap = 1.0;

N= 20;      // number of volume+momentum-segments
   F_0 = 0.40;

// select actual op-point
   Q_0 = 1.00*Q_r;

H_0 = H_r; S_0 = S_r; B_0 = B_r; U_0 = U_r; D_0 = D_r;

 D_min = +0.001; D_max = 40*D_r;
 Q_min = +0.001; Q_max = 40*Q_r;

// channel geometry
//*****************
// diffusor
vb=[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
    1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
    1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6,...
    2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0, 4.2,...
    4.4, 4.6, 4.8, 5.0, 5.0, 5.0, 5.0, 5.0,5.0];
vB = B_0*vb;
//*****************
// bottom slope
  vs=-1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8,...
           1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6,...
           2.7, 2.8, 2.9, 3.0, 3.0, 3.0, 3.0, 3.0,3.0];
vS0 = 1.0*vs;
//*************************

// initioal conditions
vd0 = ones(1,(2*N+1)); vD0 = D_0*vd0; vH0 = vD0 + vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// GMS-coefficient
   k_s = 50.0;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = D_r*vd0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
dQ = 0.40;
t_st_1 = 1200.0;  r_1_0 = Q_0; r_1_1=(1.0+1.*dQ)*Q_0;
t_st_2 = 2700.0;  r_2_0 = 0.;  r_2_1 = -2.*dQ*Q_0;
t_st_3 = 4200.0;  r_3_0 = 0.;  r_3_1 = +1.*dQ*Q_0;
T_fin = 6000.;
// inflow slew rate
g_st = 1.0; u_up_st = +5.0; u_dn_st = -5.0;
tau_st = 10.0; Q_st_0 = Q_0;

// outflow by GMS
dS_40 = vS0(39) - vS0(41);
if dS_40 >= 0 ...
  then g_o = 1.00*k_s*((1*vIr(40) + 0.5*(dS_40/L))^(1/2));
  else g_o = 1.00*k_s*((vIr(40) - (dS_40/L))^(1/2));
end;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 3000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

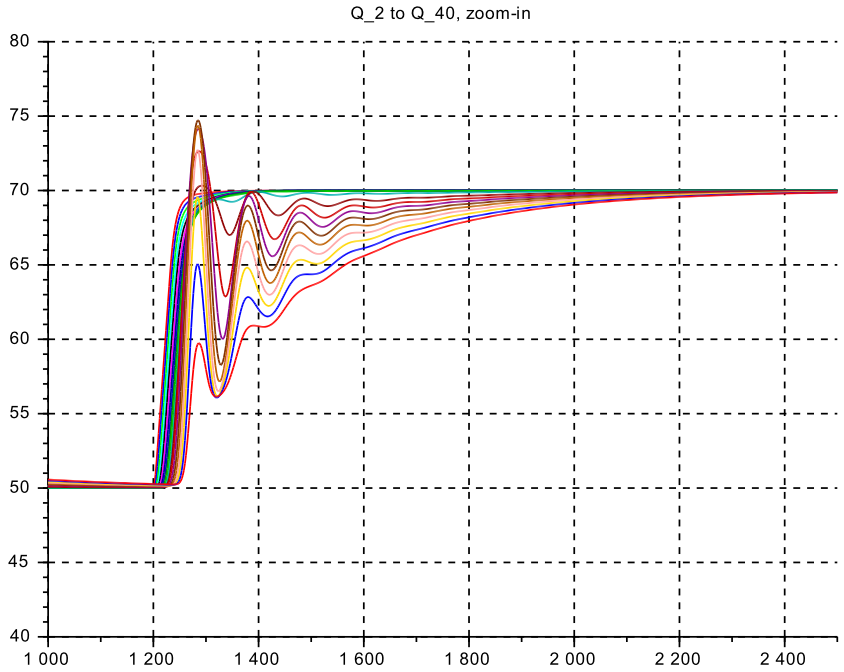
```
// s_c3_41_03_2_crunplot
//  Glf  26.10.2015
// no vertical dynamics
// diffusor
stacksize('max');exec('s_c3_41_03_2_context.sce', -1);
importXcosDiagram('s_c3_41_03_2.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:10,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[1000.,20.0,6000,80.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[1000.,0.0,6000,3.6]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[1000.,-2.0,6000,2.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[1000.,0.,6000,2.4]);
xtitle("F_2 to F_40"); xgrid(1);

//*******************************************************
f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[1000.,40.0,2500,80.]);
xtitle("Q_2 to Q_40, zoom-in"); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[1000.,0.0,2500,3.6]);
xtitle("D_1 to D_39, zoom-in"); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[1000.,-2.0,2500,2.0]);
xtitle("H_1 to H_39, zoom-in"); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[1000.,0.,2500,2.4]);
xtitle("F_2 to F_40, zoom-in"); xgrid(1);

//*******************************************************
eX = L*(1:1:N); eXend = 21*L;

for k6= 1:1:N, vSr(k6) = vS(2*k6-1); end

f9 = scf(9); yD =D.values; yH = H.values;
oD = yD(2950,:);  oH = yH(2950,:);
vcolor=[2,5,1];
plot2d(eX',[oD',oH',vSr],vcolor,rect=[0.,-3.5,eXend,+3.0]);
xtitle("Longitudinal D (blue) and H (red) at 5900 s,...
       bottom (black)");
xgrid(1);

f10 = scf(10); yF =F.values;  oF = yF(2950,:);
plot2d(eX', [oF'],  rect=[0.,0.0,eXend,+2.0]);
xtitle("Longitudinal Froude Number at 5900 s ");
xgrid(1);
```

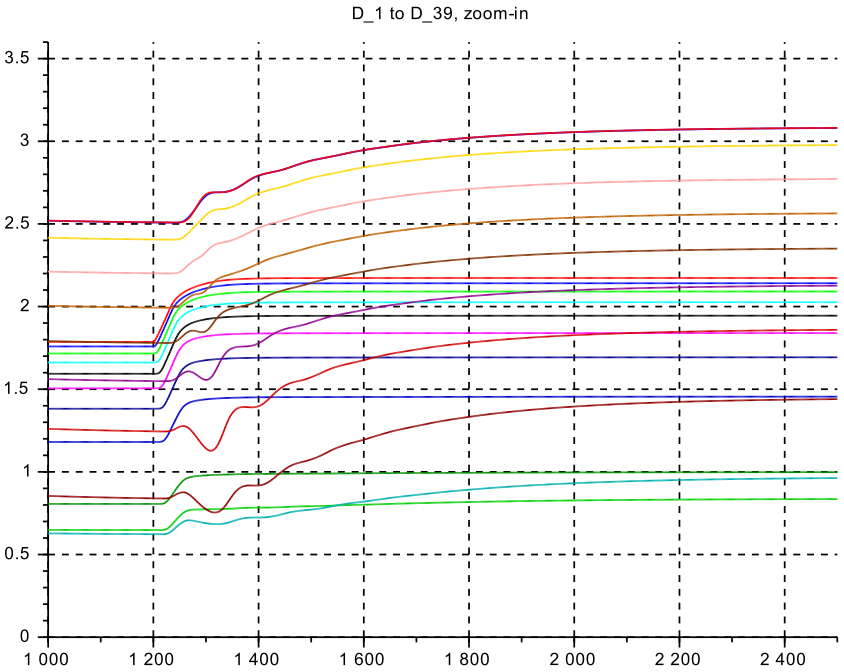
Case s_c3_41_03_2: Discussion

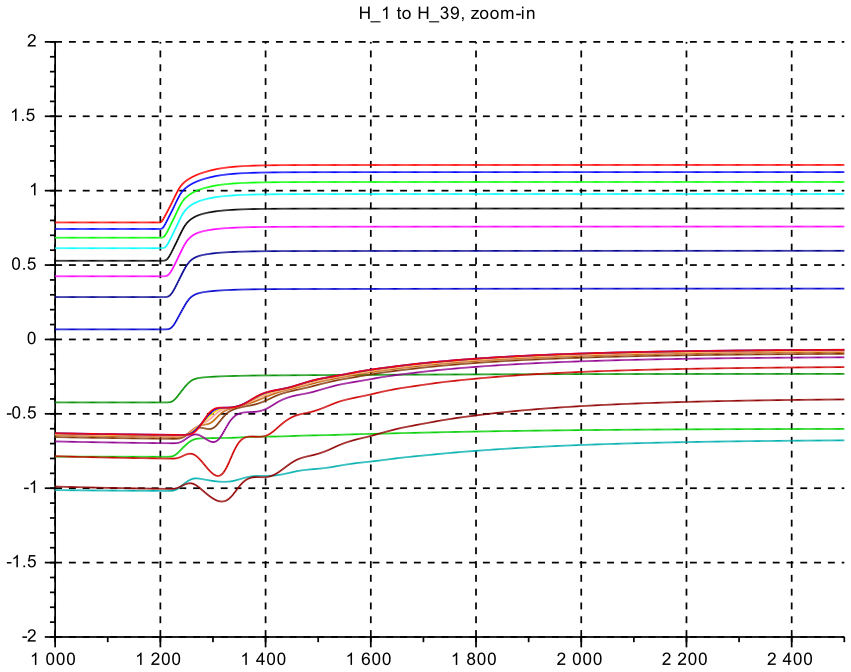
- Fig.3.23 shows the zoom-in to the first flow ramp-up, and Fig.3.24 the longitudinal profiles of $vD(k)$, $vH(k)$ and $vF(k)$ at $t = 5900\ s$.
- In the first part (up to 90 m) , the flow is accelerating but still *subcritical*. There are no oscillations here. The transients are very well damped.
- Then the flow crosses $F = 1.0$ and reaches a maximum of $F \approx 1.70$ at 220 m and drops back below $F = 1.0$ at 230 m to the final value of $F_m = 0.080$. - The calculation confirms this value by inserting $D = 2.5\ m$, that is $U_f = 5\ m/s$, and $U = 50m^3/s/(50\ m \cdot 2.5\ m) = 0.40\ m/s$, and thus $F = 0.40/5.0 = 0.080$.
- At $k = 28 \ldots 32$, in the region of strong deceleration of the subcritical flow, there are strong oscillations on $vQ(k,t)$ with a period of $\approx 80\ s$.
- This area is also clearly visible in the longitudinal profile of vH, where the level increases by $\approx 0.30\ m$ over a length of $240 \ldots 270\ m$. This is in fact a 'hydraulic jump'[5].

[5]This phenomenon will be investigated further in chapter 5

Figure 3.23: $s\_c3\_41\_03\_2\_$ diffusor geometry for flow variations at $Q = 1.0 \cdot Q\_r$






H_1 to H_39, zoom-in
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

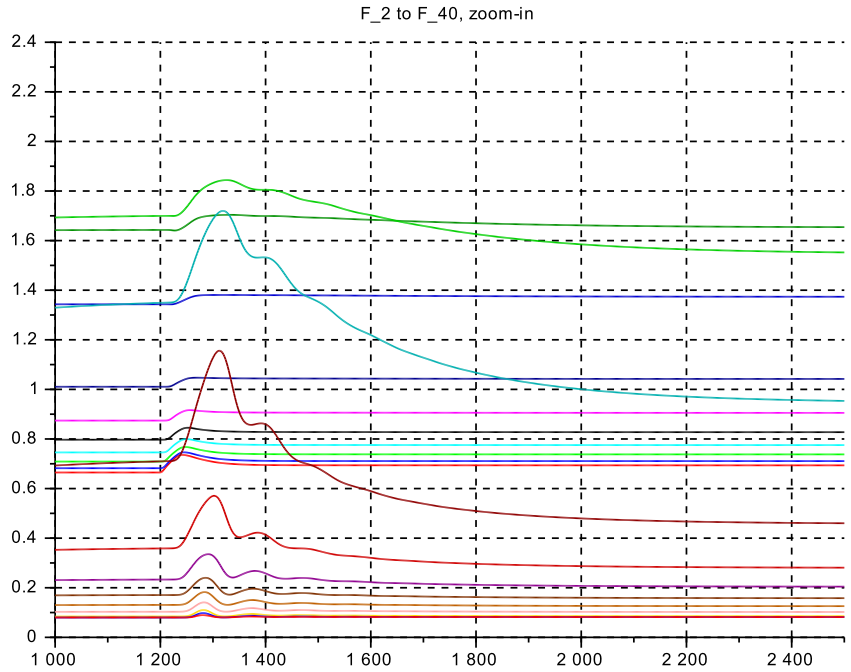
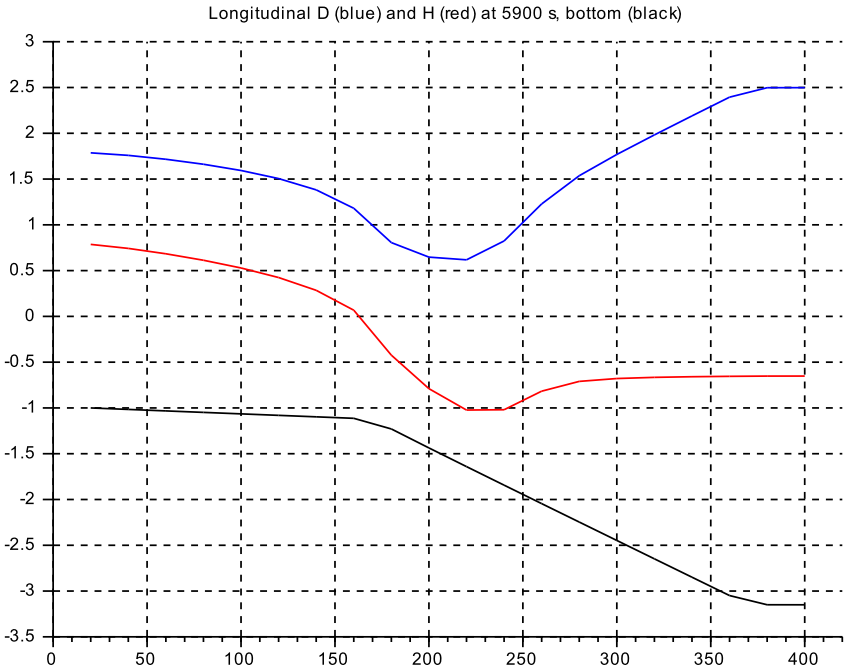
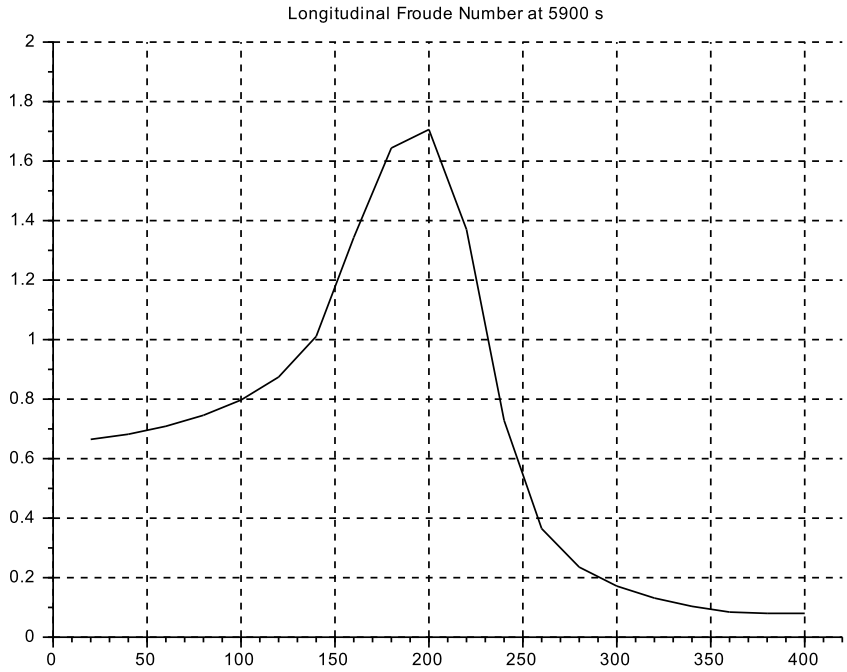

F_2 to F_40, zoom-in
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

Figure 3.24: $s\_c3\_41\_03\_2\_$ longitudinal profiles for D, H and F at $t = 5900\ s$





3.7.3 Sub-case s_c3_41_03_3: Bottom Slope Geometry

The geometry layout of the previous sub-case is modified by extending the downward slope of the bottom until the outflow end. Further the width is not opened but kept constant at its reference value up to the end.
The aim is to avoid the strong deceleration part of the hydraulic jump and focus on the flow acceleration.

.sce-files

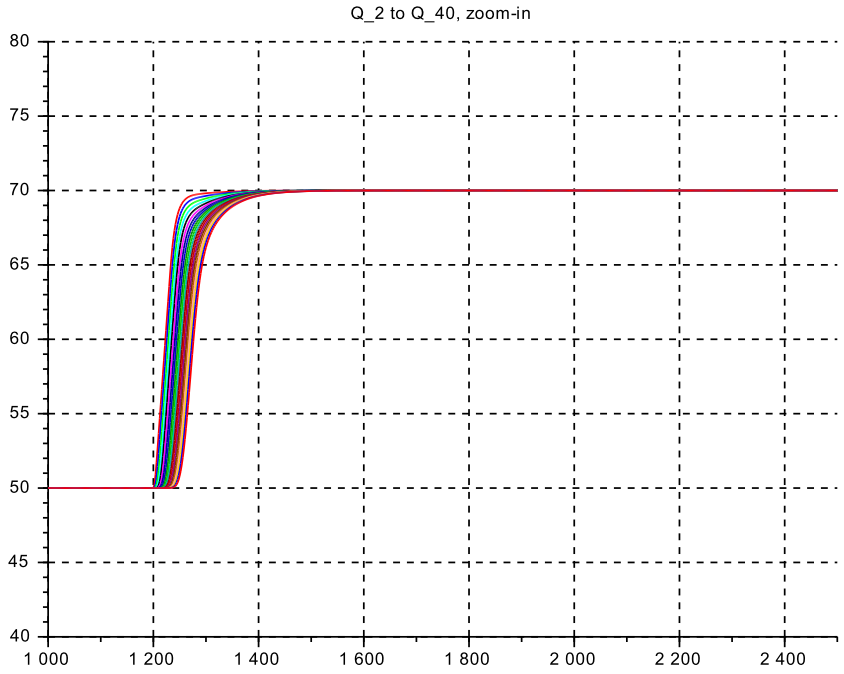
```
// s_c3_41_03_3_context
// Glf  14.4.14, 26.10.2015
// no vertical dynamics
// bottom slope

g = 10.;
// reference op-point
Q_r=50.0; H_r=1.5; S_r=1.0*(-1.0); D_r=H_r - S_r;
B_r = 10.; U_r = 2.0; F_r = 0.40;

  lambda = 8.00; L = lambda*D_r;

kap = 1.0;

N= 20;      // number of volume+momentum-segments

   F_0 = 0.40;

// select actual op-point
   Q_0 = 1.00*50.0;

H_0 = H_r; S_0 = S_r; B_0 = B_r; U_0 = U_r; D_0 = D_r;

 D_min = +0.001; D_max = 40*D_r;
 Q_min = +0.001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb = 1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*vb;
//*****************

//  horizontal to slope until outflow
vs  = -1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8,...
            1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6,...
            2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5];
vS0 = 1.0*vs;
//*************************

// initial conditions
vd0 = ones(1,(2*N+1)); vD0 = D_0*vd0; vH0 = vD0 + vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in = vB(1); S_in = vS0(1); D_in = vD0(1); Q_in = vQ0(1);

// GMS-coefficient
   k_s = 50.0;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = D_r*vd0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
dQ = 0.40;
t_st_1 = 1200.0;  r_1_0 = Q_0; r_1_1=(1.0+1.*dQ)*Q_0;
t_st_2 = 2700.0;  r_2_0 = 0.;  r_2_1 = -2.*dQ*Q_0;
t_st_3 = 4200.0;  r_3_0 = 0.;  r_3_1 = +1.*dQ*Q_0;
T_fin = 6000.;
// inflow slew rate
g_st = 1.0; u_up_st = +5.0; u_dn_st = -5.0;
tau_st = 10.0; Q_st_0 = Q_0;

// outflow by GMS
dS_40 = vS0(39) - vS0(41);
if dS_40 >= 0 then...
     g_o = 1.39*k_s*((1*vIr(40) + 0.5*(dS_40/L))^(1/2));
else g_o = 1.00*k_s*((vIr(40) - (dS_40/L))^(1/2));
end;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 3000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

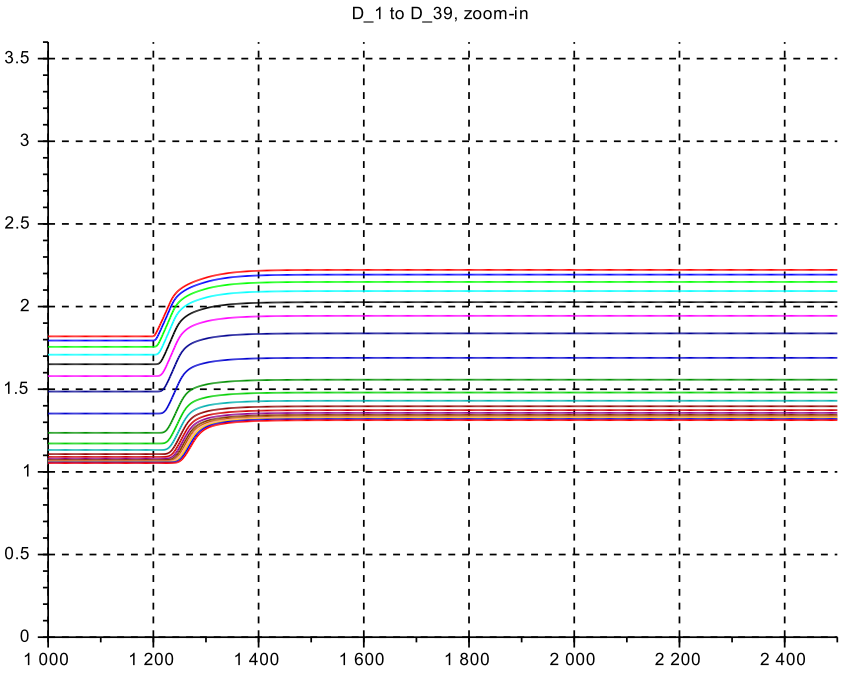
```
// s_c3_41_03_3_crunplot
//  Glf  26.10.2015
// no vertical dynamics
// bottom slope
stacksize('max');exec('s_c3_41_03_3_context.sce', -1);
importXcosDiagram('s_c3_41_03_3.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:6,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[1000,20.0,6000,80.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[1000,0.4,6000,2.8]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[1000,-3.,6000,+2.]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[1000,0.,6000,2.4]);
xtitle("F_2 to F_40"); xgrid(1);

//*******************************************************
f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[1000.,40.0,2500,80.]);
xtitle("Q_2 to Q_40, zoom-in"); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[1000.,0.0,2500,3.6]);
xtitle("D_1 to D_39, zoom-in"); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[1000.,-3.0,2500,2.0]);
xtitle("H_1 to H_39, zoom-in"); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[1000.,0.,2500,2.4]);
xtitle("F_2 to F_40, zoom-in"); xgrid(1);

//*******************************************************
eX = L*(1:1:N); eXend = 21*L;

for k6= 1:1:N,  vSr(k6) = vS(2*k6-1); end

f9 = scf(9); yD =D.values; yH = H.values;
oD = yD(2950,:); oH = yH(2950,:);
vcolor=[2,5,1];
plot2d(eX',[oD',oH',vSr],vcolor,rect=[0.,-3.5,eXend,+3.0]);
xtitle("Longitudinal D (blue) and H (red) at 5900 s, ...
       bottom (black)");
xgrid(1);

f10 = scf(10); yF =F.values; oF = yF(2950,:);
plot2d(eX', [oF'],  rect=[0.,0.0,eXend,+2.0]);
xtitle("Longitudinal Froude Number at 5900 s ");
xgrid(1);
```

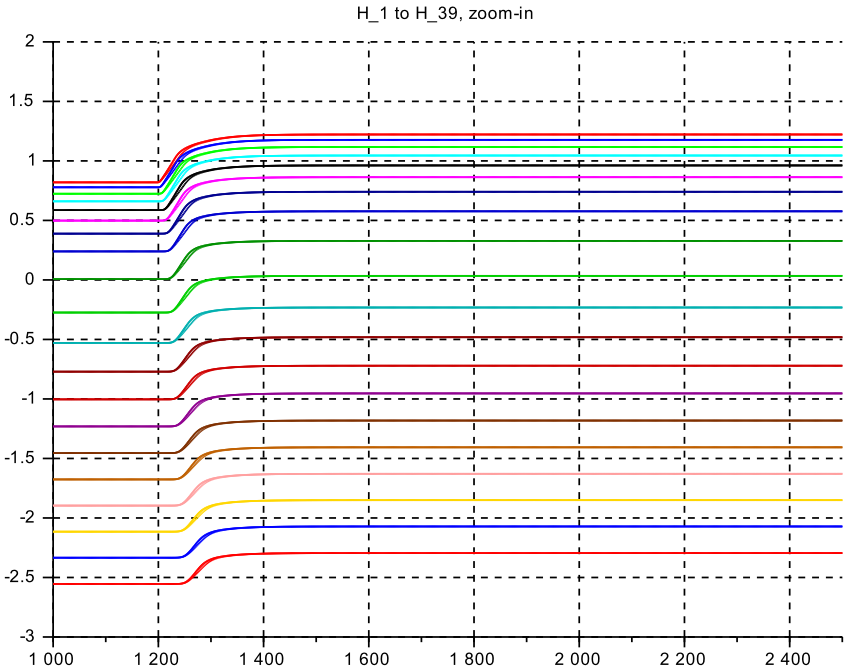
Case s_c3_41_03_3: Discussion

- Fig.3.25 shows the zoom-in to the first flow ramp-up, and Fig.3.26 the longitudinal profiles of $vD(k)$, $vH(k)$ and $vF(k)$ at $t = 5900\ s$.
- In the initial (nominally horizontal) part the flow accelerates continuously over the length. Starting at $F \approx 0.64$ it crosses over $F = 1.0$ at 160 m and continues to accelerate up to $F \approx 1.47$ near the outflow end.
- However the gradient of F on length is much smaller beyond 280 m. The flow is in 'steady state' there. It continues to slowly accelerate, as the gravity component on the inclined plane is still a bit larger than the friction component.
- No oscillations are visible anywhere along the length, as expected.

Figure 3.25: $s\_c3\_41\_03\_3\_$ Bottom slope geometry for flow variations at $Q = 1.0 \cdot Q\_r$






H_1 to H_39, zoom-in
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

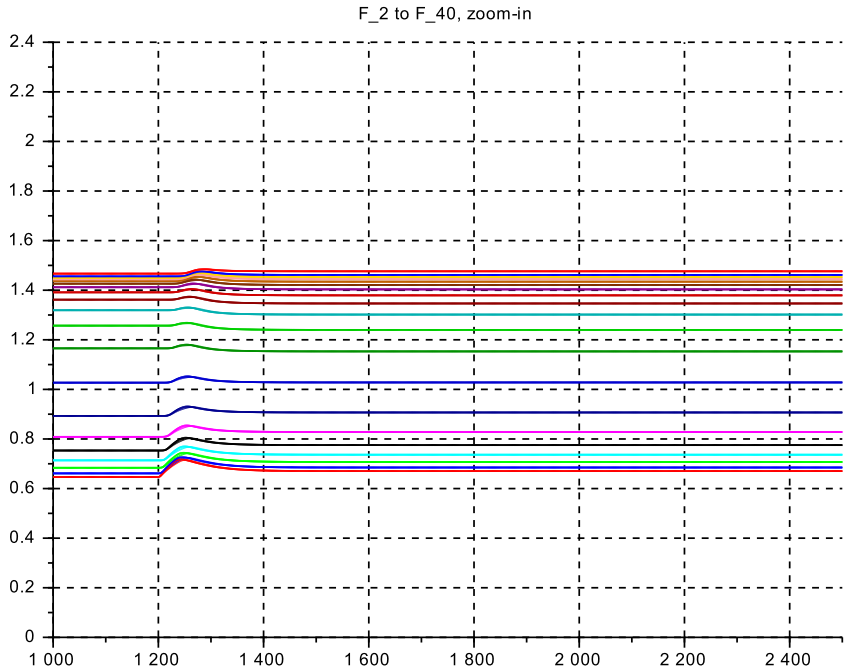
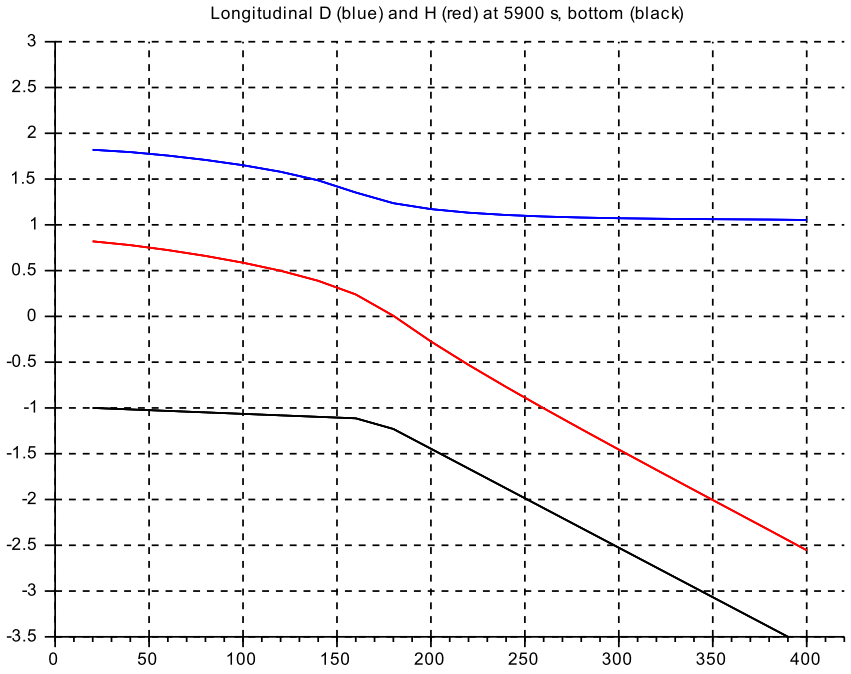
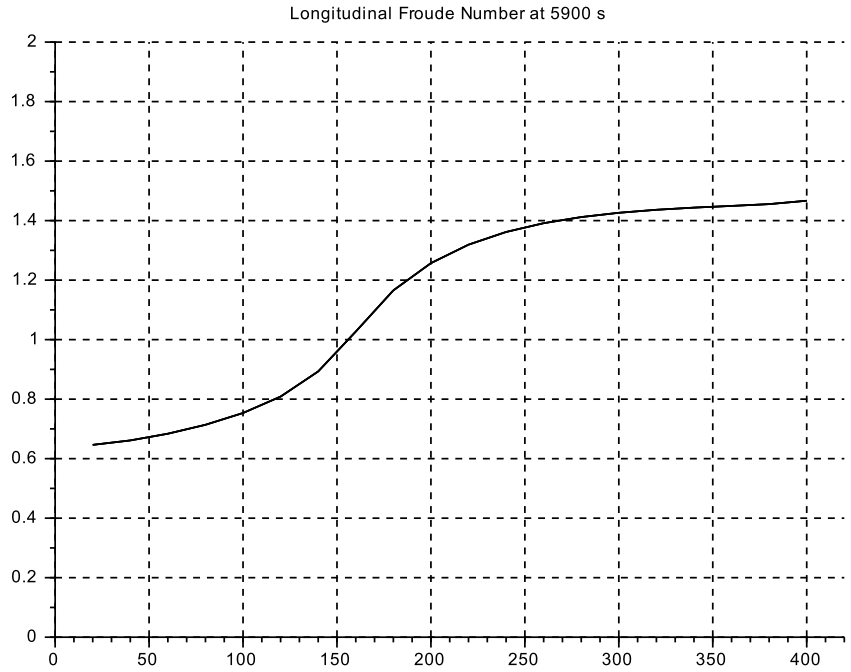

F_2 to F_40, zoom-in
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1 000
1 200
1 400
1 600
1 800
2 000
2 200
2 400

Figure 3.26: $s\_c3\_41\_03\_3\_$ longitudinal profiles for D, H and F at $t = 5900\ s$





3.8 case s_c3_41_04_ power station inlet channel

This case models the typical inlet channel to a hydropower station, where the inflow is fixed and the outflow is manipulated in a sequence of small and fast ramped steps modelling the action of a turbine/generator power controller. The cross section is set constant, steady state width and depth being constant at reference values. But flow is set either at 40 % of reference for low power conditions, or at 120 % for maximum power conditions. Thus the Froude number shall stay well below $F = 1.0$ for all conditions.
The size of the flow variations is set much smaller here (to $\mp 0.04 \cdot Q_0$) in order to limit the level excursions within realistic bounds, say $\Delta D \leq +0.5\ m$.
The friction bottom slope for the channel is calculated for the reference flow and kept on for the two other flow conditions, as in reality.
The symmetric sub-case of constant outflow and variations in inflow has been investigated, but is skipped here for brevity.

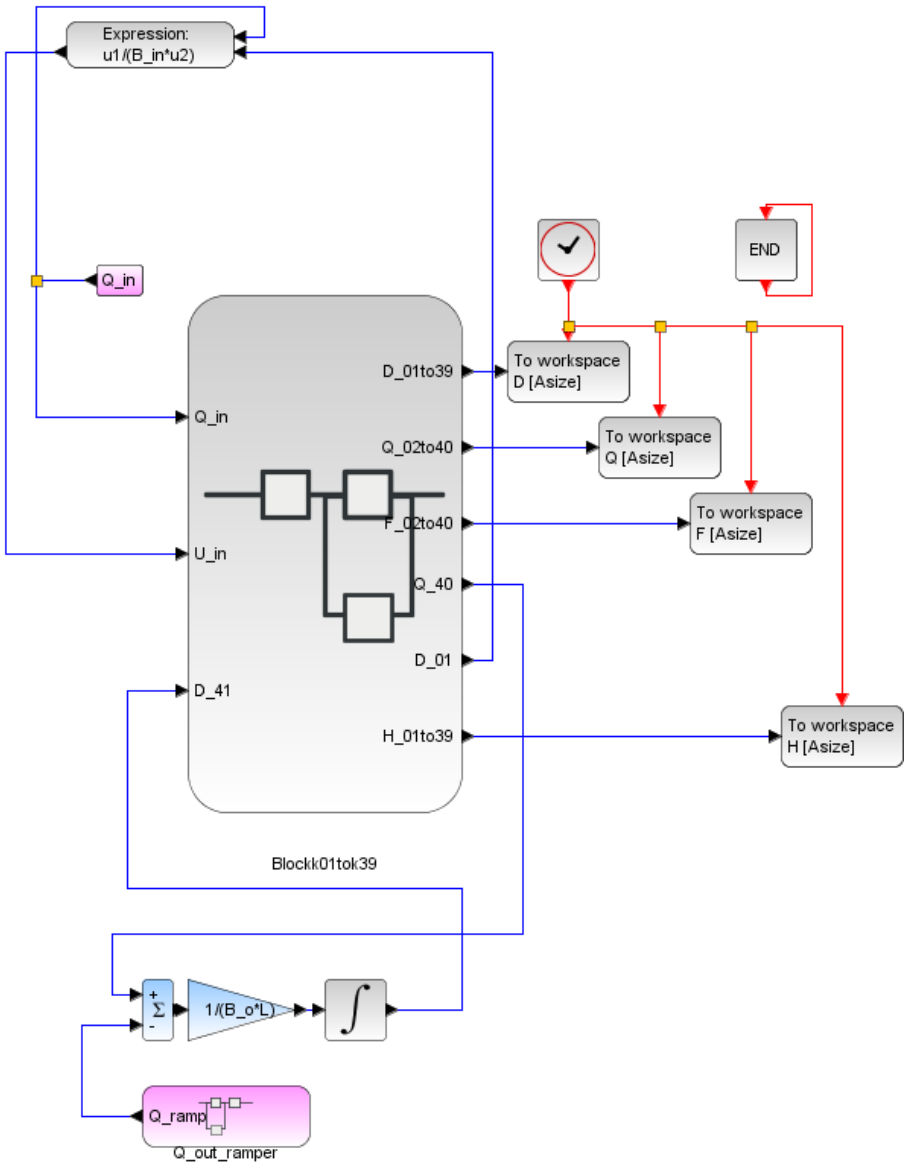


Figure 3.27: top-level diagram for case s_c3_41_04_2, with no overfall

.sce-files

```
// s_c3_41_04_2_context
// Glf   26.10.2015
// no vertical dynamics

g = 10.;

// reference op-point
Q_r=50.0; H_r=1.5; S_r=1.0*(-1.0); D_r=H_r - S_r;
B_r = 10.; U_r = 2.0; F_r = 0.40;

  lambda = 8.00; L = lambda*D_r;

kap = 1.0;

N= 20;      // number Volume+momentum-segments

   F_0 = 0.40;

// select actual op-point
   Q_0 = 0.40*50.0;
// Q_0 = 1.20*50.0;

H_0=H_r; S_0=S_r; B_0=B_r; U_0=U_r; D_0=D_r;

 D_min = +0.0001; D_max = 40*D_r;
 Q_min = +0.0001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb =   [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*vb;
//*****************

// basic layout: horizontal bottom
vs  = -1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = 1.0*vs;
//*************************

vd0 = ones(1,(2*N+1)); vD0 = D_0*vd0; vH0 = vD0 + vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);
```

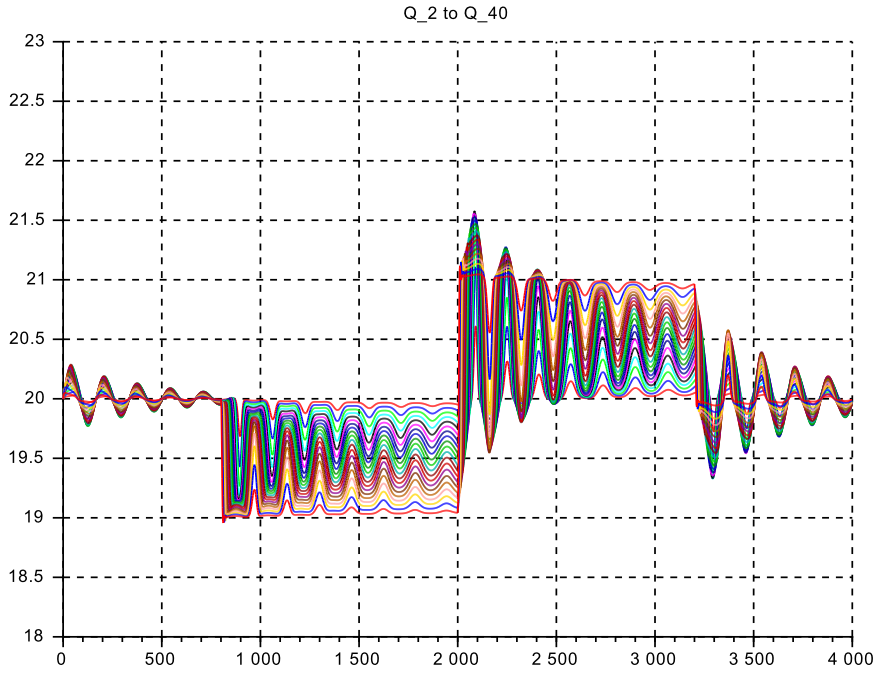
```
// GMS-coefficient
//k_s = 100.;
//k_s = 70.7;
  k_s = 50.0;
//k_s = 31.6;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = D_r*vd0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// outflow generation
dQ = 0.05;
t_st_1 = 800.0;   r_1_0 = Q_0; r_1_1=(1.0-1.*dQ)*Q_0;
t_st_2 = 2000.0;  r_2_0 = 0.;  r_2_1 = +2.*dQ*Q_0;
t_st_3 = 3200.0;  r_3_0 = 0.;  r_3_1 = -1.*dQ*Q_0;
T_fin = 4000.;
// outflow slew rate
g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st = 10.0; Q_st_0 = Q_0;

// inflow fixed:
Q_in = Q_0;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

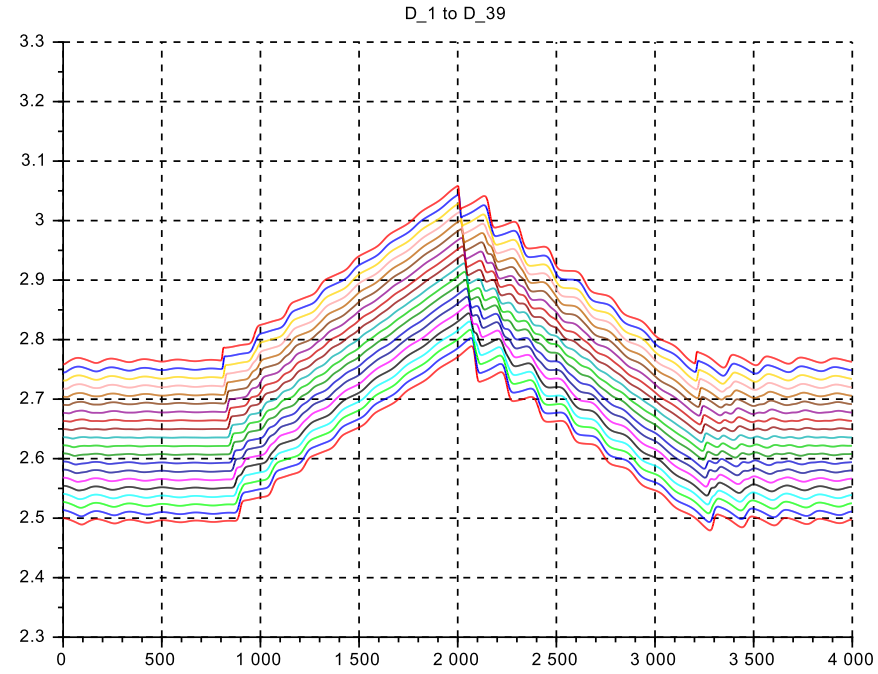
```
// s_c3_41_04_2_crunplot
//  Glf  26.10.2015
// no vertical dynamics

stacksize('max');exec('s_c3_41_04_2_context.sce', -1);
importXcosDiagram('s_c3_41_04_2.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:4,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];


f1 = scf(1);
   plot2d(Q.time,Q.values,vcolor,rect=[0.,18.,4000,23.]);
// plot2d(Q.time,Q.values,vcolor,rect=[0.,54.,4000,66.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
   plot2d(D.time,D.values,vcolor,rect=[0.,2.3,4000,3.3]);
// plot2d(D.time,D.values,vcolor,rect=[0.,2.2,4000,3.8]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
   plot2d(H.time,H.values,vcolor,rect=[0.,1.2,4000,1.8]);
// plot2d(H.time,H.values,vcolor,rect=[0.,0.8,4000,2.4]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
   plot2d(F.time,F.values,vcolor,rect=[0.,0.1,4000,0.2]);
// plot2d(F.time,F.values,vcolor,rect=[0.,0.,4000,0.8]);
xtitle("F_2 to F_40"); xgrid(1);
```

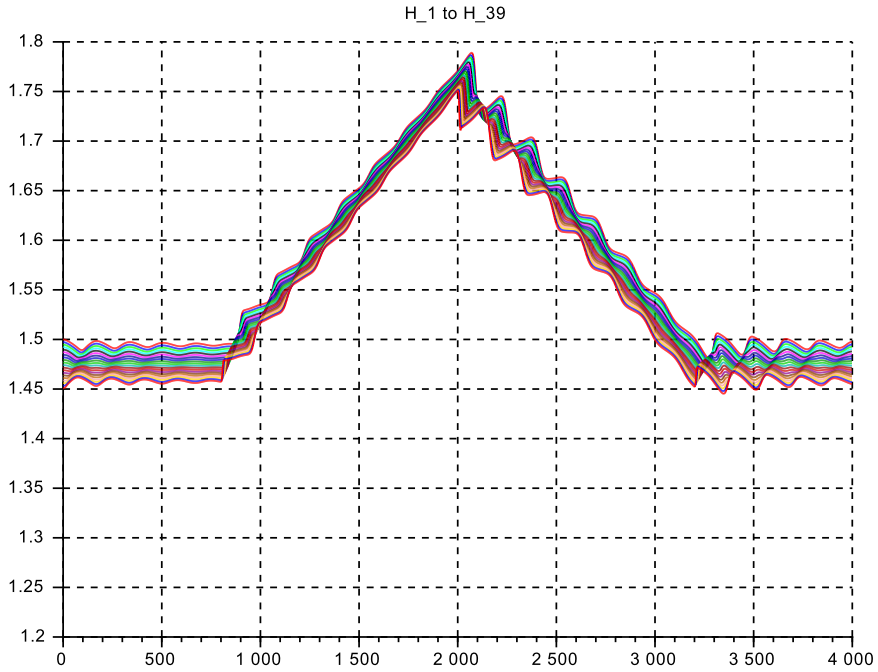
Case s_c3_41_04_2: Discussion

- Fig.3.28 shows the total transients for the low power conditions and Fig.3.29 for the high power conditions
- As expected, the transients are much less damped than in the GMS-law outflow cases above: The impedance at the outflow end is much stiffer, in fact it is infinite.
- At the low flow condition, the damping is extremely weak
- Also the travelling wave fronts are neatly visible on $vD(k,t)$.

Figure 3.28: $s\_c3\_41\_04\_2$ for outflow variations at $Q = 0.40 \cdot Q\_r$, constant inflow






H_1 to H_39
1.8
1.75
1.7
1.65
1.6
1.55
1.5
1.45
1.4
1.35
1.3
1.25
1.2
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000

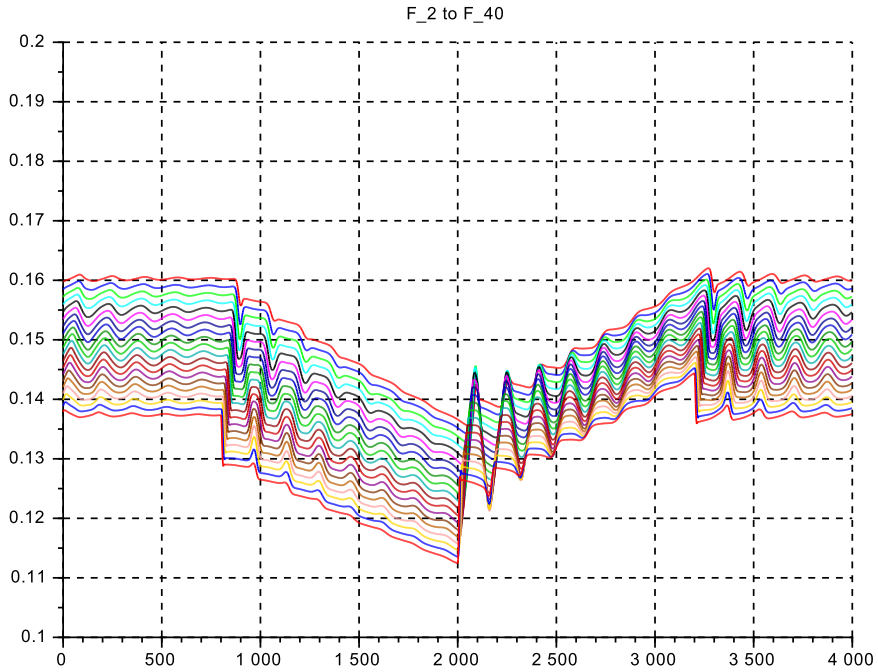
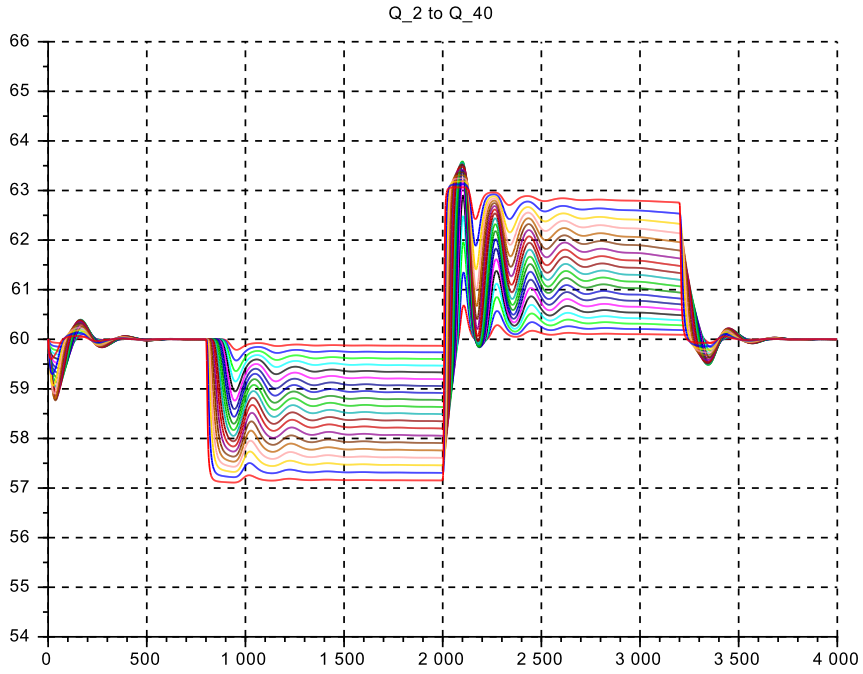
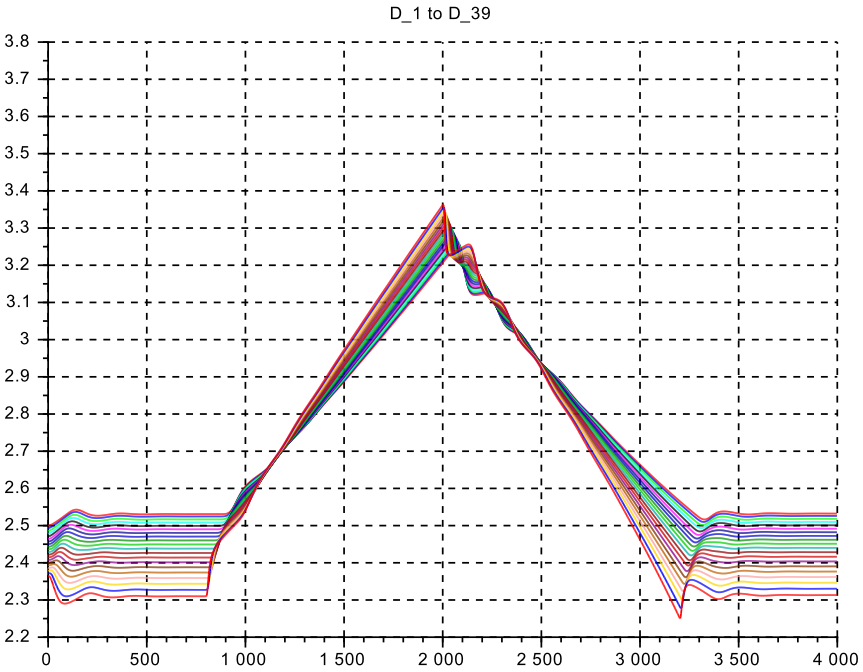
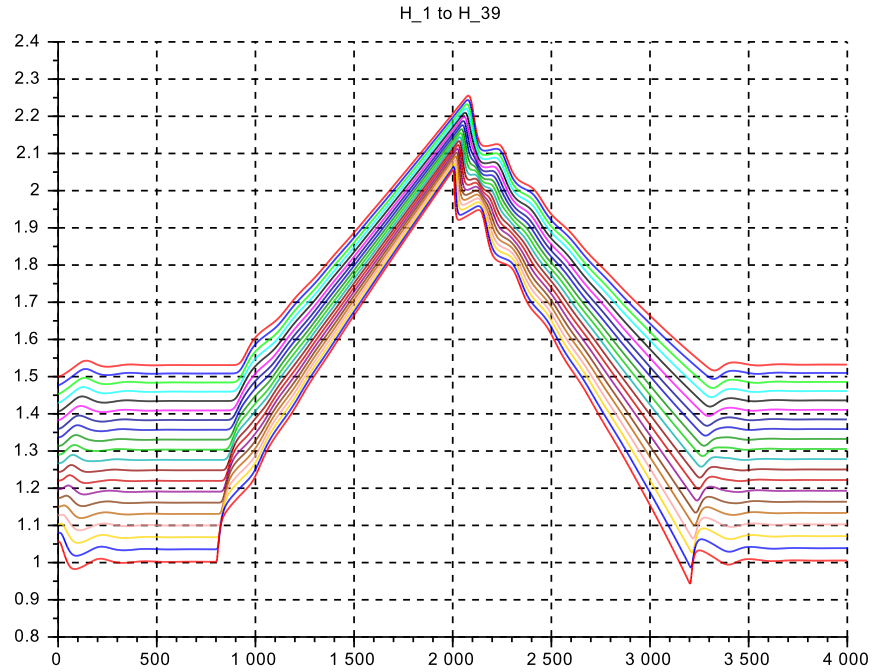
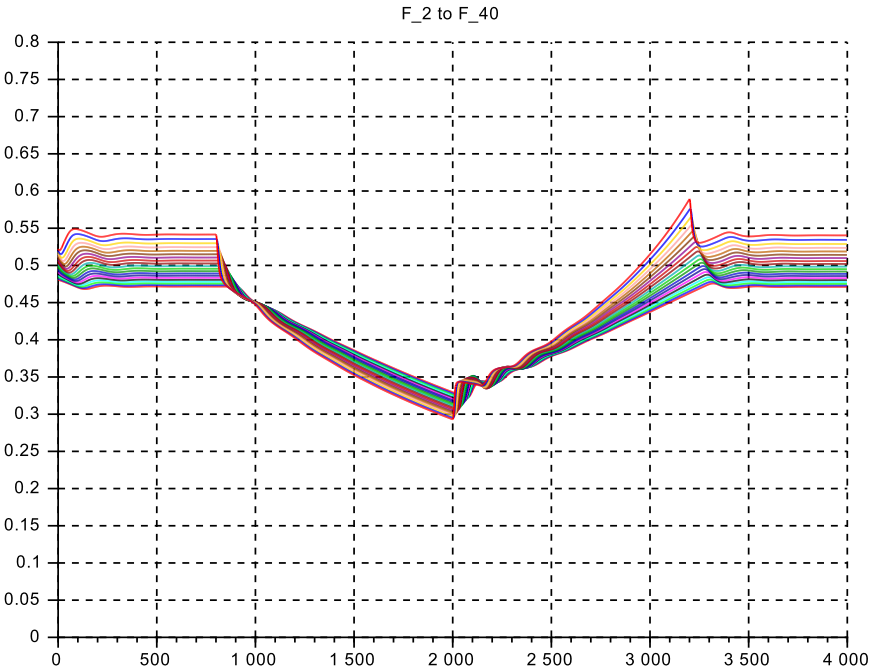

F_2 to F_40
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000

Figure 3.29: $s\_c3\_41\_04\_$ for outflow variations at $Q = 1.20 \cdot Q\_r$, constant inflow






H_1 to H_39
2.4
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000


F_2 to F_40
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000

Turbine inlet channel with overfall on first volume compartment

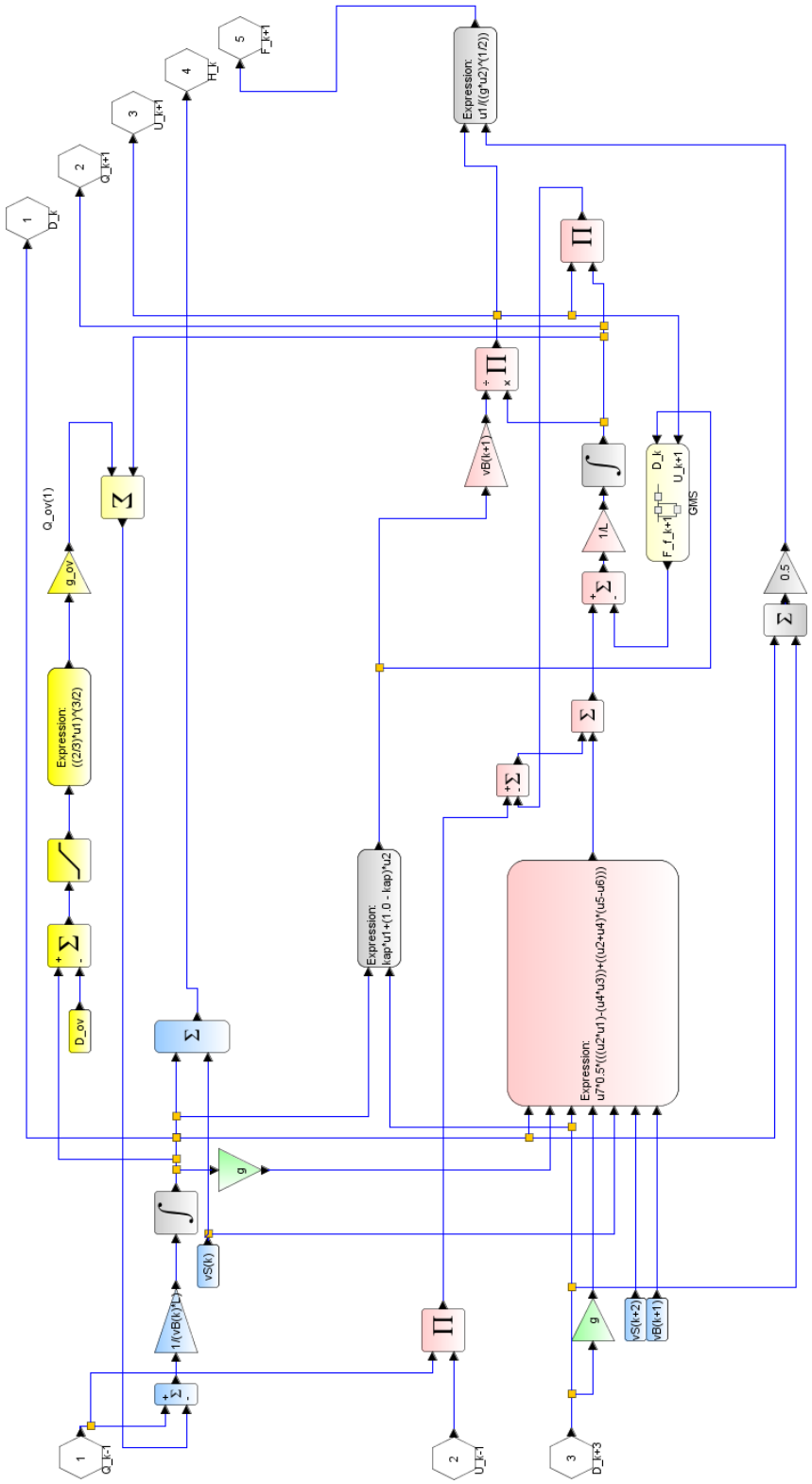


Figure 3.30: lowest-level diagram for element $k = 1$ with overfall at inflow, s_c3_41_04_3

.sce-files

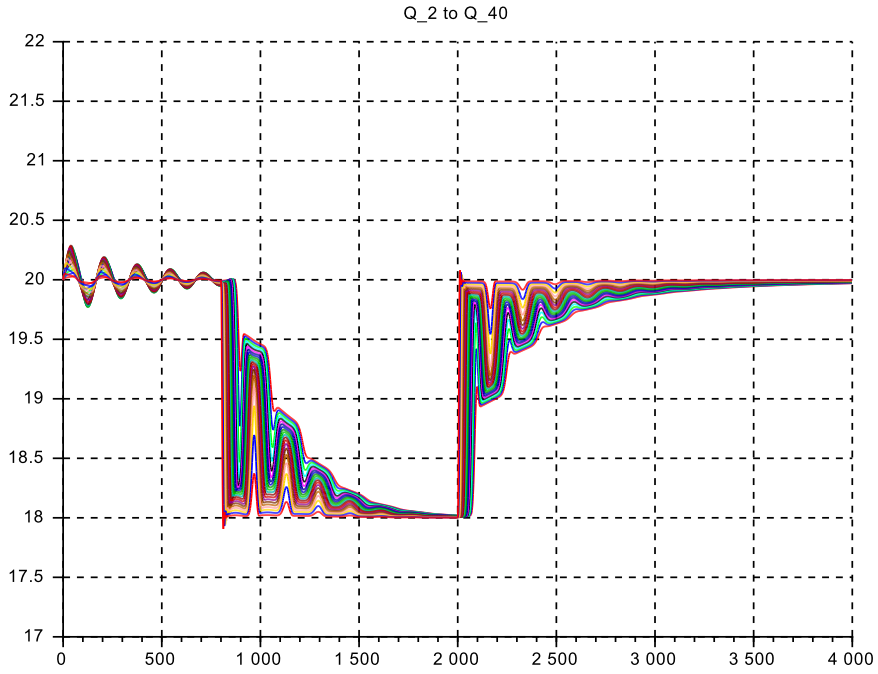
```
// s_c3_41_04_3_context
// Glf  26.10.2015
// no vertical dynamics
// with overfall at inlet

g = 10.;
// reference op-point
Q_r=50.0; H_r=1.5; S_r=1.0*(-1.0); D_r=H_r - S_r;
B_r = 10.; U_r = 2.0; F_r = 0.40;

  lambda = 8.00; L = lambda*D_r;

// overfall at inflow compartment
D_ov = D_r - 0.0; g_ov = L*(g^(0.5));

kap = 1.0;

N= 20;      // number Volume+momentum-segments

   F_0 = 0.40;

// select actual op-point
   Q_0 = 0.40*50.0;
// Q_0 = 1.20*50.0;

H_0=H_r; S_0=S_r; B_0=B_r; U_0=U_r; D_0=D_r;

 D_min = +0.0001; D_max = 40*D_r;
 Q_min = +0.0001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb =   [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*vb;
//******************

// basic layout: horizontal bottom
vs  = -1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
           1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = 1.0*vs;
//*************************

vd0 = ones(1,(2*N+1)); vD0 = D_0*vd0; vH0 = vD0 + vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// GMS-coefficient
//k_s = 100.;
//k_s = 70.7;
  k_s = 50.0;
//k_s = 31.6;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = D_r*vd0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// outflow generation
dQ = 0.10;
t_st_1 = 800.0;   r_1_0 = Q_0; r_1_1=(1.0-1.*dQ)*Q_0;
t_st_2 = 2000.0;  r_2_0 = 0.;  r_2_1 = +1.*dQ*Q_0;
t_st_3 = 3200.0;  r_3_0 = 0.;  r_3_1 = -0.*dQ*Q_0;
T_fin = 4000.;
// outflow slew rate
g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st = 10.0; Q_st_0 = Q_0;

// inflow fixed:
Q_in = Q_0;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

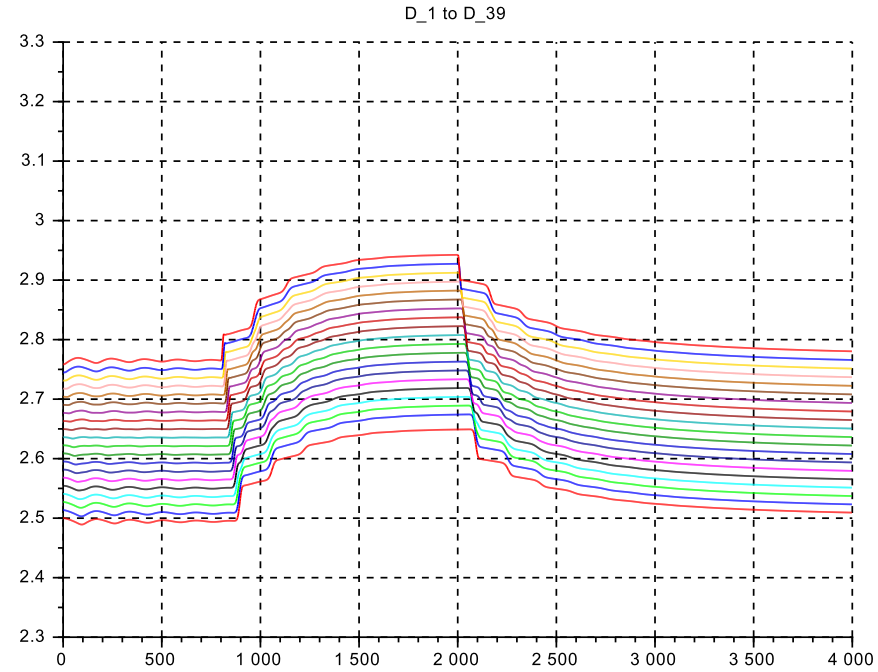
```
// s_c3_41_04_3_crunplot
//  Glf  26.10.2015
//  no vertical dynamics
// with overfall at inlet
stacksize('max');exec('s_c3_41_04_3_context.sce', -1);
importXcosDiagram('s_c3_41_04_3.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:4,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
   plot2d(Q.time,Q.values,vcolor,rect=[0.,17.,4000,22.]);
// plot2d(Q.time,Q.values,vcolor,rect=[0.,52.,4000,64.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
   plot2d(D.time,D.values,vcolor,rect=[0.,2.3,4000,3.3]);
// plot2d(D.time,D.values,vcolor,rect=[0.,2.2,4000,3.4]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
   plot2d(H.time,H.values,vcolor,rect=[0.,1.2,4000,1.8]);
// plot2d(H.time,H.values,vcolor,rect=[0.,0.8,4000,2.4]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
   plot2d(F.time,F.values,vcolor,rect=[0.,0.1,4000,0.2]);
// plot2d(F.time,F.values,vcolor,rect=[0.,0.0,4000,0.8]);
xtitle("F_2 to F_40"); xgrid(1);
```

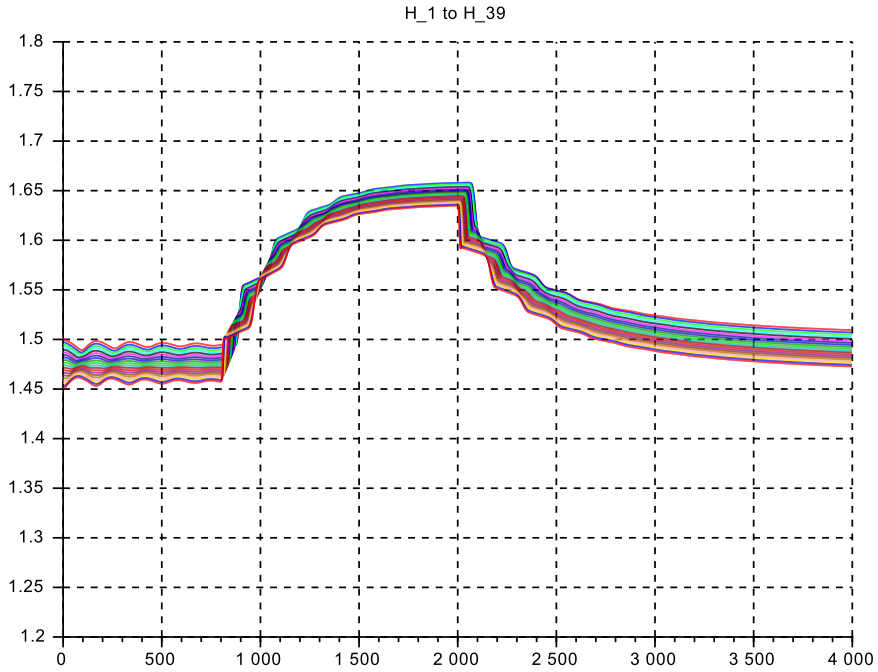
Case s_c3_41_04_3: Discussion

- Fig.3.31 shows the total transients for the low power conditions and Fig.3.32 for the high power conditions
- The overfall threshold is set to $D\_ov = D\_r = 2.5\ m$,
- and the flow variation is downward only $\Delta Q = -0.10 \cdot Q$ to go back to initial Q, (for having an increasing water level and thus overfall flow)
- At the low flow condition, the damping is still weak, but significantly improved on the end part of the transient, where $vD(1,t) > D\_ov\ \forall t$, than in the initial part of the transient where $vD(1,t)$ stays near to D_ov.
- Also the travelling wave fronts are neatly visible on $vQ(k,t), vD(k,t), vF(k,t)$.

Figure 3.31: $s\_c3\_41\_04\_3$ constant inflow, $\Delta Q_0 = 0.10 \cdot Q\_r$






H_1 to H_39
1.8
1.75
1.7
1.65
1.6
1.55
1.5
1.45
1.4
1.35
1.3
1.25
1.2
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000

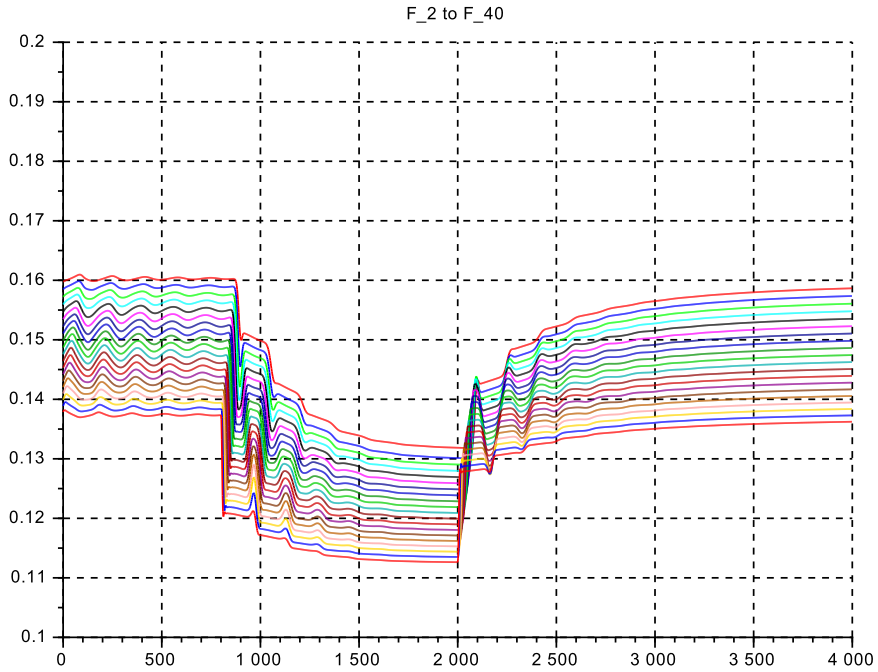
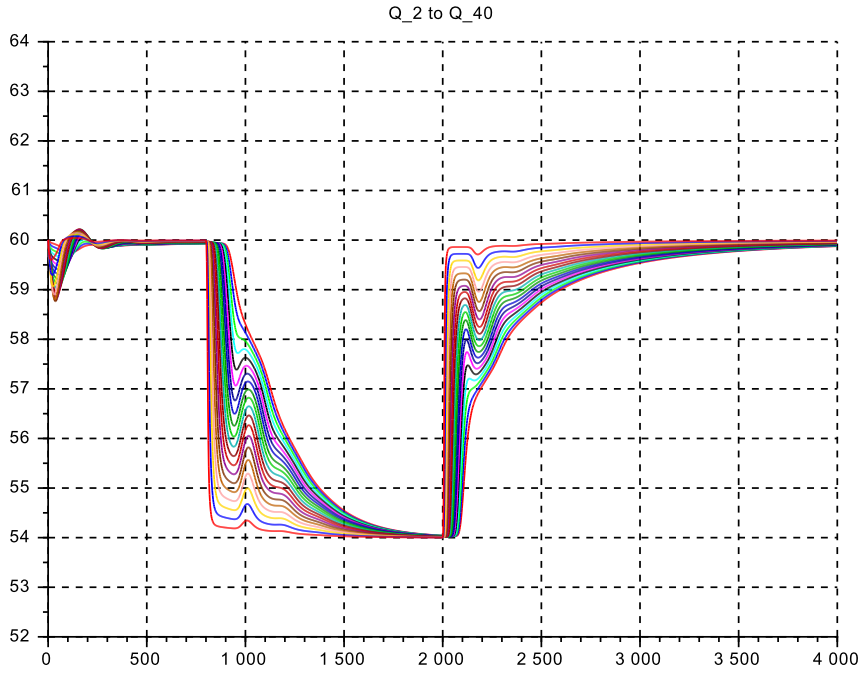
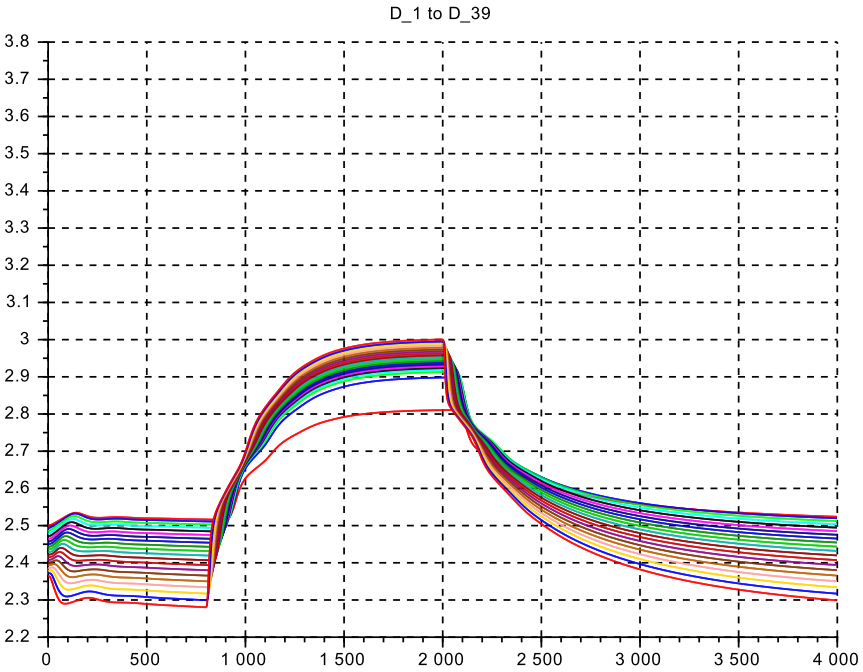
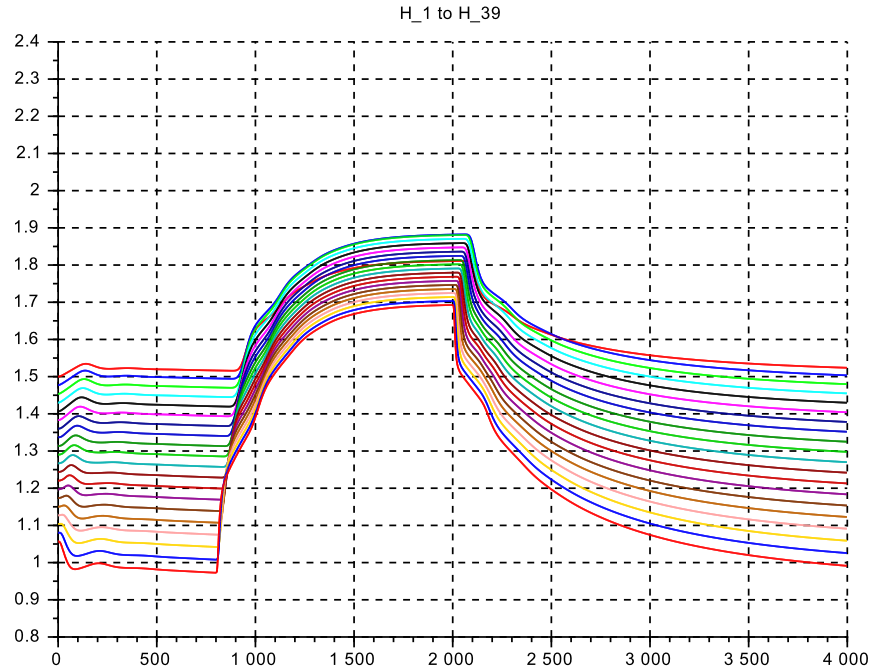
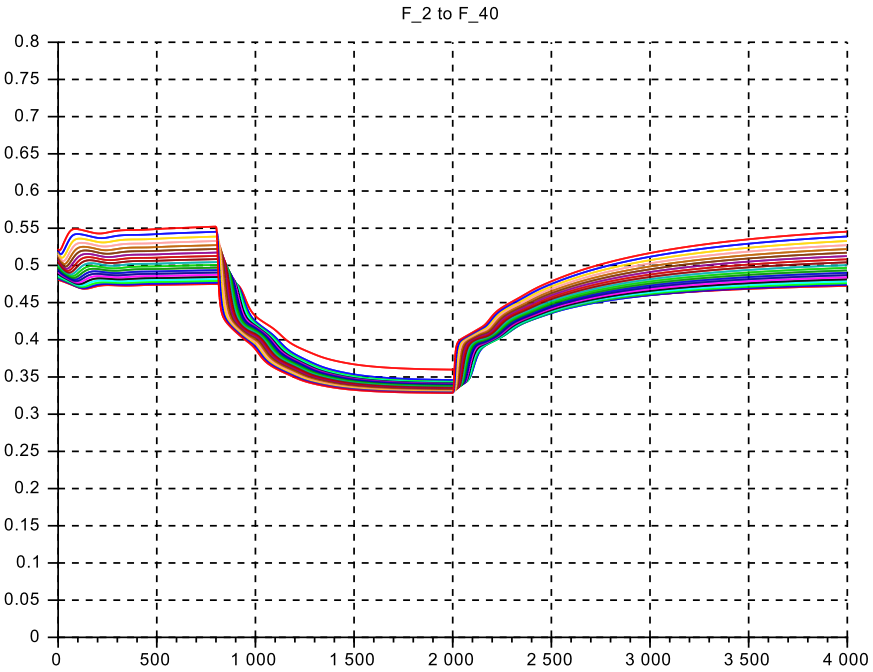

F_2 to F_40
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000

Figure 3.32: $s\_c3\_41\_04\_3$ for outflow variations at $Q = 1.20 \cdot Q\_r$, constant inflow






H_1 to H_39
2.4
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000


F_2 to F_40
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
500
1 000
1 500
2 000
2 500
3 000
3 500
4 000

3.9 Case Study 'Birsfelden'

This is about the upstream basin dynamics of the hydro power station at 'Birsfelden' on the Rhine (WGS 84 location: 47.56019, 7.63047), where the outflow on the lower end is manipulated and the inflow at the upper end is the outflow of the upstream power station. There is no stretch of free river flow on the upstream end of the reservoir. The basin width is nearly constant along the length of the basin, and the borders are steep. Thus the constant rectangular cross section model can be used as a reasonable approximation.

In this case experimental results on level response are available from two level sensors, one on the upstream pool near the power station and one situated near the upper end of the basin at $\approx 7'000m$.

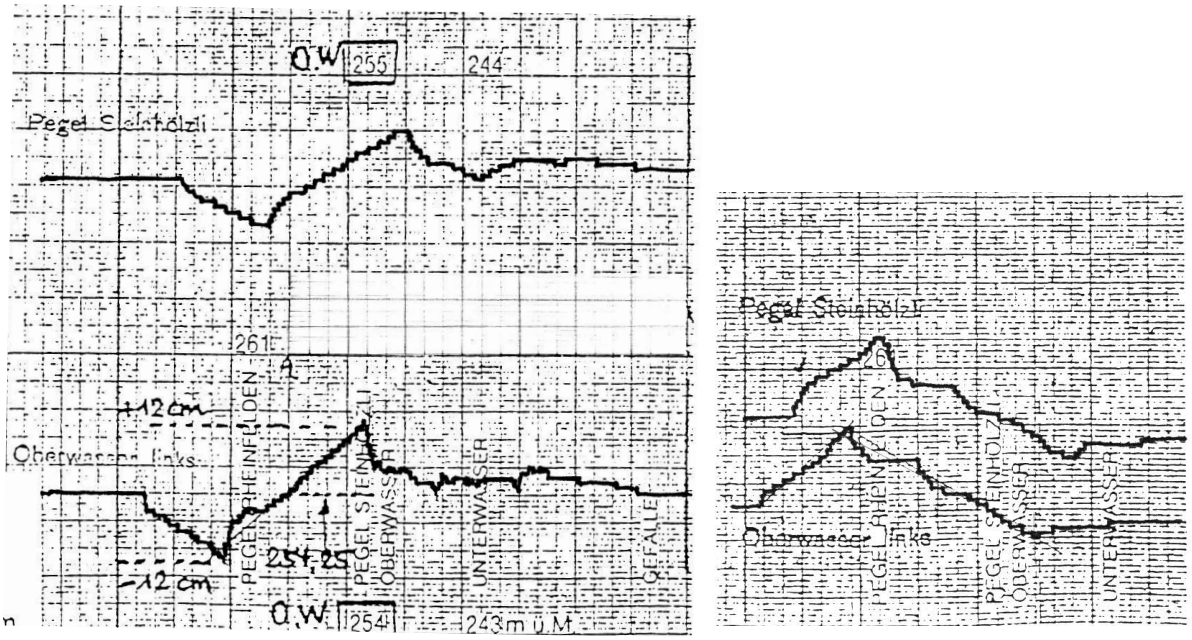


Figure 3.33: step responses on both levels: left at $\approx 1'000m^3/s$, right at $\approx 500m^3/s$
Lower track for sensor near power plant, upper track for sensor $7'000m$ upstream

The experiments dating from 1986 were performed because poor damping was observed on the existing level control loop at very low river flow conditions. For the tests, the level control loop was put on 'manual' and the turbine opening was manipulated in a sequence of rectangular moves. The water level on the upstream side of the power station was restricted to $\approx \pm 0.12\ m$ in order to respect the level tolerances imposed by the Swiss federal agency.
The results were not followed up at that time, but they motivated the author's ongoing interest in river basin dynamics and ended up with the modelling, simulation and control design presented in the current report.

The **aim** here is to compare the simulation results to the measured ones, and thus to validate the models built in the previous sections.

3.9.1 Data assembly

The first step will be to collect the data on the river basin and on the experiments. Unfortunately the data given on the web sites are not complete. So the missing data must be 'reconstructed' to give a consistent framework for the simulation.

The characteristic *river flow* values are:

High flow: $3'000\ m^3/s$ (weirs to be opened)
Full turbine flow: $4x400\ m^3/s\ =\ 1'600\ m^3/s$ Mean flow: $1'000\ m^3/s$ Low flow: $500\ m^3/s$

The upstream *basin dimensions* are (from Swiss topo maps):
Total length: $L\_tot = 7'850m$ that is with 21 elements $L = 375m$ Mean width: $\overline{B} = 157m$.

The *water levels* are (given in m above sea level):
Nominal upstream level at power station: $254.25m$
downstream levels: at $500m^3/s$: $245.25m$, at $1'000m^3/s$: $246.26m$ at $3'000m^3/s$: $249.52m$.
The location of this sensor is not given, it is assumed near to the power station. Also water levels at the upstream power station ('Augst-Wyhlen') are not available.

The *bottom levels* are not given in a consistent manner. So they have to be 'reconstructed'. This is done by considering the free river flow before the power station was built:
There is an indication on the inclination: $150m$ drop over a distance of $150km$, that is a mean inclination value of $\overline{I} = 10 \cdot 10^{-4}$.
Assume $I := 10.37 \cdot 10^{-4}$ and for the GMS-friction coefficient $k_s := 32.5$ (fine pebble ground).
Then the water depths are (by iteration):
At $500m^3/s$ $\overline{D} = 2.0m$, at $1'000m^3/s$ $\overline{D} = 3.0m$, at $1'600m^3/s$ $\overline{D} = 4.0m$.
Using the downstream water levels given above this produces a bottom level at the power station position of $243.25m$, and an upstream water depth at the power station position of $\approx 11.0m$. And the bottom level rises to the upstream end of the basin by $S(41) - S(1) = \Delta S = -7.8m$.
Then set the reference level "0" such that $S\_r = S(1) := -0.80m$ and $H\_r = H(41) := 2.20m$. Thus at *zero flow* the water depth will go from $3.0m$ at the 'inflow' end up to $10.8m$ at the 'outflow' end.

3.9.2 Modelling

The second step is to model the experiment on the plant.

The exact values of the actual river flow [6] are not available precisely but have been estimated to be $\approx 1'000m^3/s$ and $\approx 500m^3/s$. So they are set to $1'000m^3/s$ and to $500m^3/s$, as this produces good fits with the measured data.

The simulation shall be initialised from the free flow situation as specified above. Then the level buildup to its target value at $254.25m$ above sea level or $2.20m$ above the defined 'zero'-level will be done by a level controller of PI-type. Its settings k_p, k_i are tuned experimentally for a settling within approx. $10'000s$. When the control loop has attained its steady state (at $12'000s$), the controller input is switched off (the control error forced to zero), thus opening the level control loop. Next the sequence of turbine opening changes is applied. The time instants of the sequence are read out from the traces of the upstream level sensor close to the power station.
At $1'000m^3/s$:

> Outflow up at $00:12$, down at $00:54$; that is $\Delta t = 42min = 2'520s$;
>
> up again at $02:06$; that is $\Delta t = 72min = 4'320s$
>
> and down to near steady state flow at $02:45$, that is $\Delta t = 36min = 2'340s$;

And at $500m^3/s$:

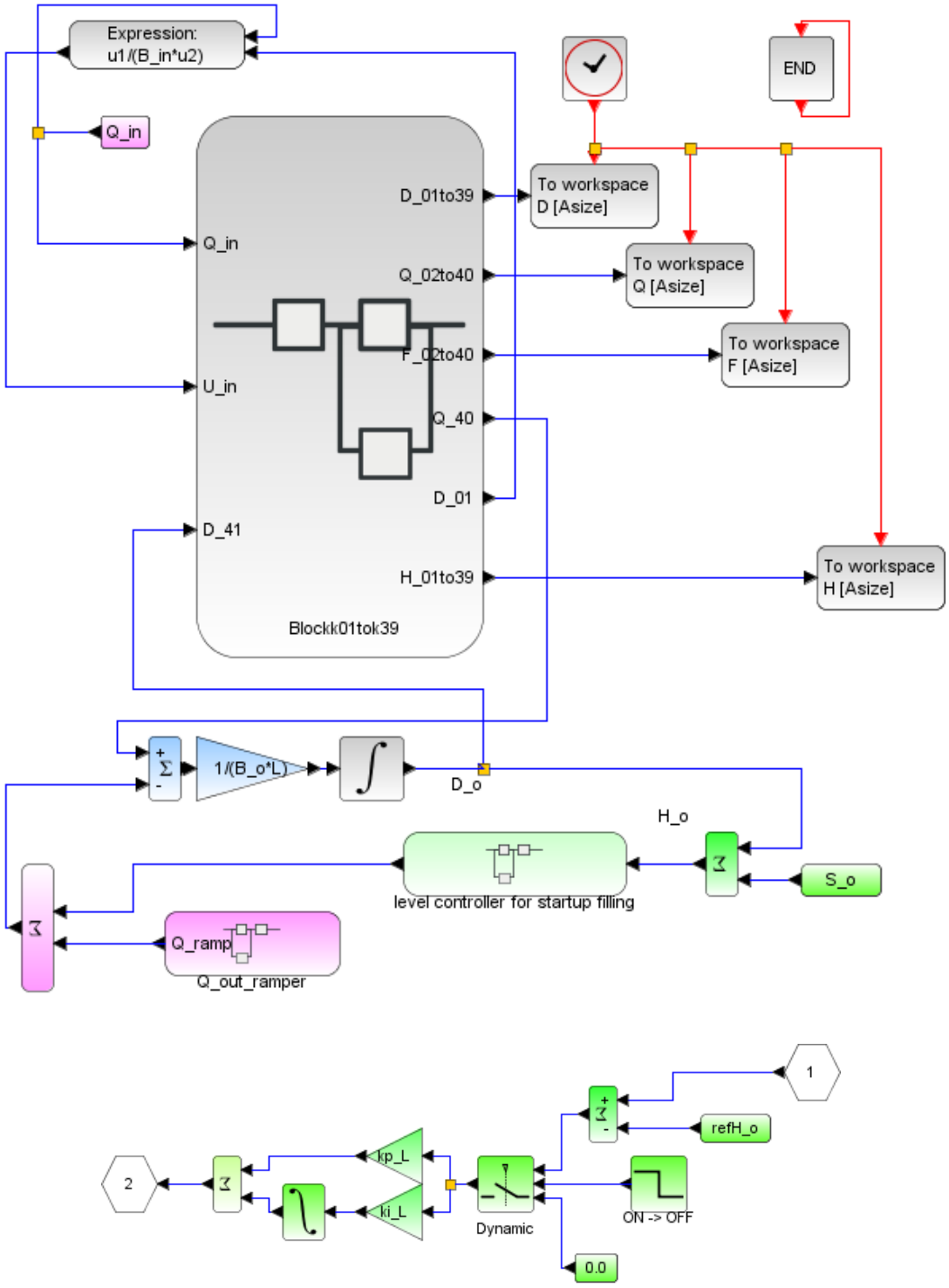
> Outflow down at $00:08$, up at $00:54$; that is $\Delta t = 46min = 2'760s$;
>
> down to near steady state flow at $02:32$, that is $\Delta t = 98min = 5'880s$;

The step sizes on the turbine flow steps are not available precisely. They have been estimated around $\pm 80m^3/s$. And the applied ΔQ-sizes have again been 'reconstructed' by the measured level deviations just prior to the next step, see the respective `...context.sce`-file for details.

[6] Note that in contrast to levels, flows are much more difficult and expensive to measure on large hydro plants

3.9.3 Implementation in scilab

.zcos diagram for s_c3_41_05_10.zcos (for $1'000m^3/s$) and s_c3_41_05_05.zcos (for $500m^3/s$)



context.sce-file for s_c3_41_05_10.zcos

```
// s_c3_41_05_10_context
// Glf  26.10.2015
// Birsfelden 1000m^3/s
// open loop responses

g = 10.;
// reference op-point
Q_r=1000.0; H_r=2.2; S_r=0.80*(-1.0); D_r=H_r - S_r;
B_r = 157.; U_r = 2.02; F_r = 0.368;

L = 375.;

kap = 1.0;

N= 20;      // number Volume+momentum-segments

// select actual op-point
 Q_0 = 1.00*Q_r;

 H_0 = H_r; S_0 = S_r; B_0 = B_r;
 U_0 = Q_r/(B_r*D_r); D_0 = D_r;

 D_min = +0.0001; D_max = 40*D_r;
 Q_min = +0.0001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb =   [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*vb;
//******************

// basic layout: horizontal bottom
vs  = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = S_0*vs;
//*************************

vd0=ones(1,(2*N+1)); vD0=D_0*vd0; vH0=vD0 + vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// GMS-coefficient
//k_s = 100.;
//k_s = 70.7;
// k_s = 50.0;
   k_s = 32.5;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = D_r*vd0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
vH00   = ones(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
//      vH0(k4) = 0.45*vH00(k4);
      vH0(k4) = 2.7*vH00(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// outflow generation
dQ = 0.0625;
t_st_1 = 12000.0;   r_1_0 = Q_0; r_1_1=(1.0+0.8*dQ)*Q_0;
t_st_2 = 14520.0;  r_2_0 = 0.;  r_2_1 = -1.8*dQ*Q_0;
t_st_3 = 18840.0;  r_3_0 = 0.;  r_3_1 = +1.8*dQ*Q_0;
t_st_4 = 21180.0;  r_4_0 = 0.;  r_4_1 = -0.8*dQ*Q_0;
T_fin  = 24000.0;
// outflow slew rate
g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st = 20.0; Q_st_0 = Q_0;

// Outflow Level control
kp_L = 1600.0;
ki_L = 400/1000;
refH_o = 2.2;

T_LevConOFF = 12000.; ulevcon0 = +1.; ulevcon1 = -1.0;

// inflow fixed:
Q_in = Q_0;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

context.sce-file for s_c3_41_05_05.zcos

```
// s_c3_41_05_05_context
// Glf  16.6.14
// Birsfelden 500m^3/s
// open loop

g = 10.;
// reference op-point
Q_r=1000.0; H_r=2.2; S_r=0.8*(-1.0); D_r=H_r - S_r;
B_r = 157.; U_r = Q_r/(B_r*D_r); F_r = 0.387;

L = 375.;

kap = 1.0;

N= 20;      // number Volume+momentum-segments

// select actual op-point
   Q_0 = 0.50*Q_r;

H_0=H_r; S_0=S_r; B_0=B_r; U_0=U_r; D_0=D_r;

 D_min = +0.0001; D_max = 40*D_r;
 Q_min = +0.0001; Q_max = 40*Q_r;

// channel geometry
//*****************
// Basic element: constant (nominal) width
vb =   [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*vb;
//*****************

// basic layout: horizontal bottom
vs  = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
       1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = S_0*vs;
//*************************

vd0=ones(1,(2*N+1)); vD0=D_0*vd0; vH0=vD0 + vS0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1);D_in=vD0(1); Q_in=vQ0(1);

// GMS-coefficient
//k_s = 100.;
//k_s = 70.7;
// k_s = 50.0;
   k_s = 32.5;

// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
vDr = D_r*vd0;
vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));
vHf    = zeros(1,(2*N+1));
vH00   = ones(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
  vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
    vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vIr(kk);
    vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
    vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
    vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
        vH(k4)  = vH0(k4)  + vHf(k4);
        vH0(k4) = vH(k4) ;
      vH0(k4) = 2.625*vH00(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1); H_o=vH0(kk+1);
    Q_o = vQ0(kk+1);

// outflow generation
dQ = 0.12;
t_st_1 = 12000.0;   r_1_0 = Q_0; r_1_1=(1.0-1.0*dQ)*Q_0;
t_st_2 = 14460.0;  r_2_0 = 0.;  r_2_1 = +1.6*dQ*Q_0;
t_st_3 = 20340.0;  r_3_0 = 0.;  r_3_1 = -0.6*dQ*Q_0;
t_st_4 = 24000.0;  r_4_0 = 0.;  r_4_1 = 0.;
T_fin  = 24000.0;
// outflow slew rate
g_st = 100.0; u_up_st = +10.0; u_dn_st = -10.0;
tau_st =20.0; Q_st_0 = Q_0;

// Outflow Level control
kp_L = 1600.0;
ki_L = 100/100;
refH_o = 2.20;

T_LevConOFF = 12000.; ulevcon0 = +1.; ulevcon1 = -1.0;

// inflow fixed:
Q_in = Q_0;

// Data transfer to Plots
CC = 21;            // no of channels 20 + 1 for time
CN = 2000;          // no of  clockticks up to Tfin
delT = T_fin/CN;    // intervall for clock ticks
Asize = 1.01*CC*CN; // size of  Data arrays
```

crunplot.sce-files for s_c3_41_05_10.zcos and s_c3_41_05_05.zcos

```
// s_c3_41_05_10_crunplot
//  Glf  26.10.2015
//  Birsfelden 1000 m^3/s
// open loop
stacksize('max');exec('s_c3_41_05_10_context.sce', -1);
importXcosDiagram('s_c3_41_05_10.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:5,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

T_0 = 0.0; T_1 = 10000.; T_2 = T_fin;
f1 = scf(1);
// plot2d(Q.time,Q.values,vcolor,rect=[T_0,800,T_2,1200]);
   plot2d(Q.time,Q.values,vcolor,rect=[T_1,800,T_2,1200]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
// plot2d(D.time,D.values,vcolor,rect=[T_0,2.0,T_2,12.0]);
   plot2d(D.time,D.values,vcolor,rect=[T_1,2.0,T_2,12.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
// plot2d(H.time,H.values,vcolor,rect=[T_0,+1.8,T_2,3.2]);
   plot2d(H.time,H.values,vcolor,rect=[T_1,+1.8,T_2,3.2]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
// plot2d(F.time,F.values,vcolor,rect=[T_0,0.0,T_2,0.4]);
   plot2d(F.time,F.values,vcolor,rect=[T_1,0.0,T_2,0.4]);
xtitle("F_2 to F_40"); xgrid(1);

//*****************************************************
eX = L*(1:1:N); eXend = 21*L;

for k6= 1:1:N,  vSr(k6) = vS(2*k6-1); end

f5 = scf(5); yH = H.values; oH = yH(1000,:);
vcolor=[5,1];
plot2d(eX', [oH', vSr], vcolor, rect=[0.,-9.0,eXend,+4.0]);
xtitle("Longitudinal  H (red) at 12000 s, bottom (black)");
xgrid(1);
```

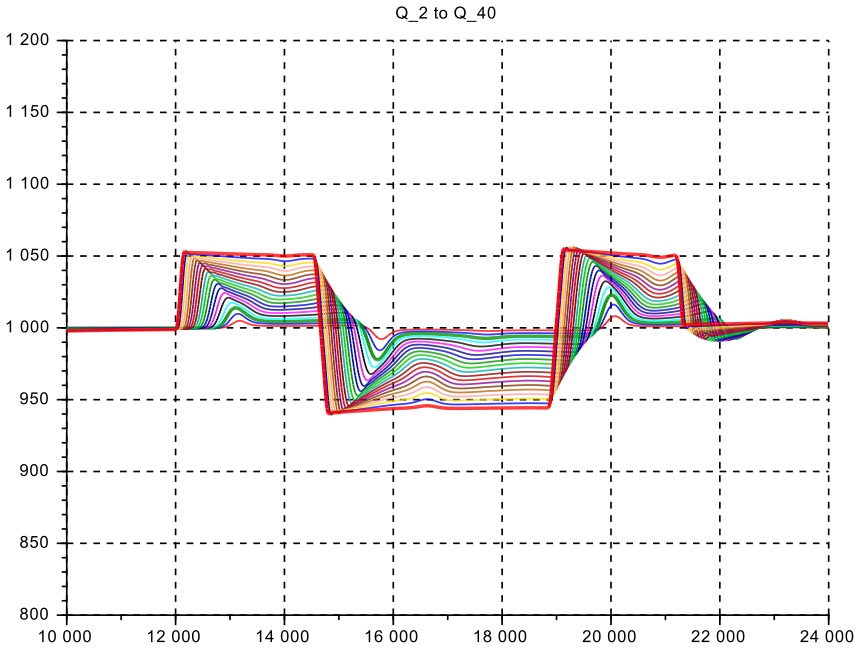
```
// s_c3_41_05_05_crunplot
//  Glf  26.10.2015
//  Birsfelden 600 m^3/s
// open loop
stacksize('max');exec('s_c3_41_05_05_context.sce', -1);
importXcosDiagram('s_c3_41_05_05.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:5,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

T_0 = 0.; T_1 = 10000; T_2 = T_fin;
f1 = scf(1);
// plot2d(Q.time,Q.values,vcolor,rect=[T_0,400.,T_2,600.]);
   plot2d(Q.time,Q.values,vcolor,rect=[T_1,400.,T_2,600.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
// plot2d(D.time,D.values,vcolor,rect=[T_0,2.0,T_2,12.0]);
   plot2d(D.time,D.values,vcolor,rect=[T_1,2.0,T_2,12.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
// plot2d(H.time,H.values,vcolor,rect=[T_0,1.8,T_2,2.8]);
   plot2d(H.time,H.values,vcolor,rect=[T_1,1.8,T_2,2.8]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
// plot2d(F.time,F.values,vcolor,rect=[T_0,0.0,T_2,0.2]);
   plot2d(F.time,F.values,vcolor,rect=[T_1,0.0,T_2,0.2]);
xtitle("F_2 to F_40"); xgrid(1);

//*****************************************************
eX = L*(1:1:N); eXend = 21*L;

for k6= 1:1:N, vSr(k6) = vS(2*k6-1); end

f5 = scf(5); yH = H.values; oH = yH(1000,:);
vcolor=[5,1];
plot2d(eX', [oH', vSr], vcolor, rect=[0.,-9.0,eXend,+4.0]);
xtitle("Longitudinal  H (red) at 12000 s, bottom (black)");
xgrid(1);
```

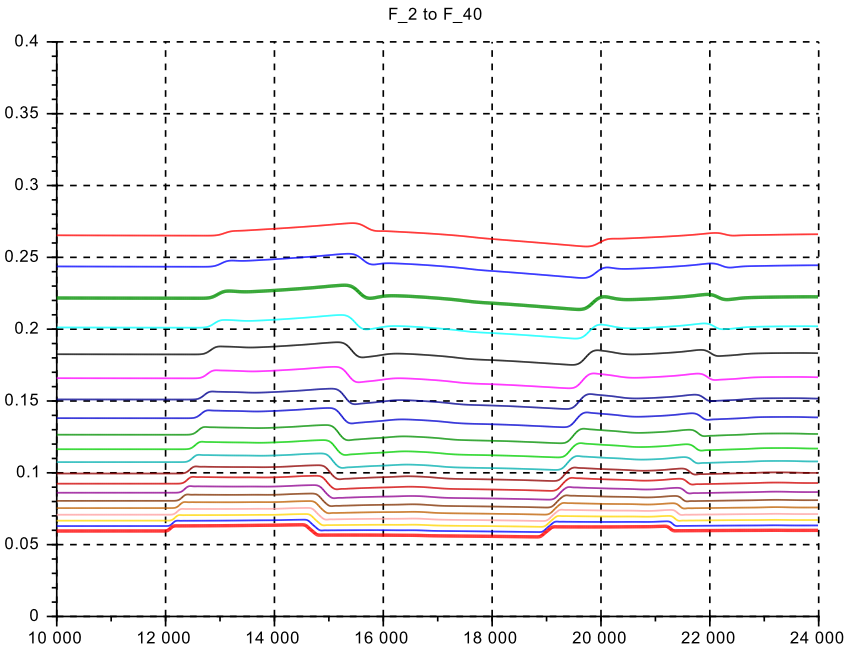
Remarks

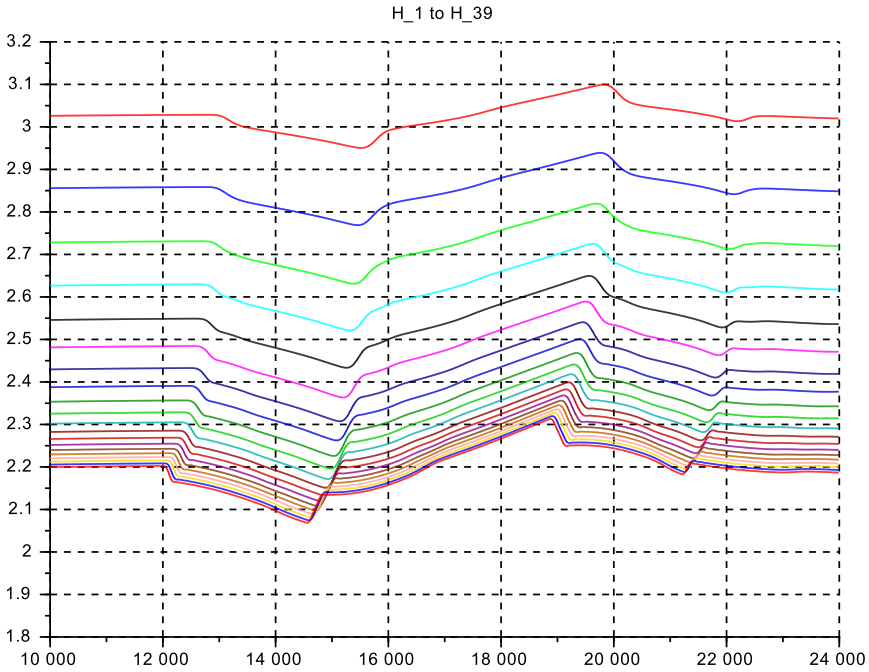
- The settling time of the level control loop has been suppressed in the following plots. They focus on the interesting part of the open-loop responses to the outflow sequence while the level controller is switched off.
- The plot of $D(k,t)$ is not shown. The longitudinal profile of $H(k)$ is shown instead to indicate the 'retention volume' and the depths when approaching the inflow end of the basin.
- The thick dark green trace shows the variables near the location of the upstream sensor at $7'000m$
- And the thick red trace shows the variables at location $k = 39$ to be consistent with previous plots. Thus $H(39)$ is shown and not $H(41)$ which is the actual measured value input to the level controller.

3.9.4 Simulation Results

Figure 3.34: $s\_c3\_41\_05\_10$ for outflow variations at $1'000 m^3/s$, constant inflow






H_1 to H_39
3.2
3.1
3
2.9
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2
1.9
1.8
10 000
12 000
14 000
16 000
18 000
20 000
22 000
24 000

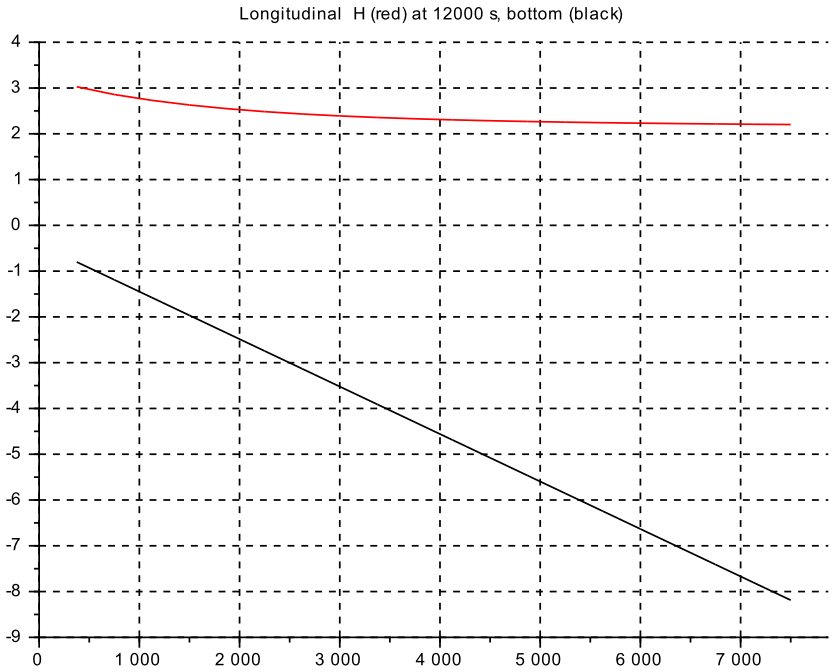
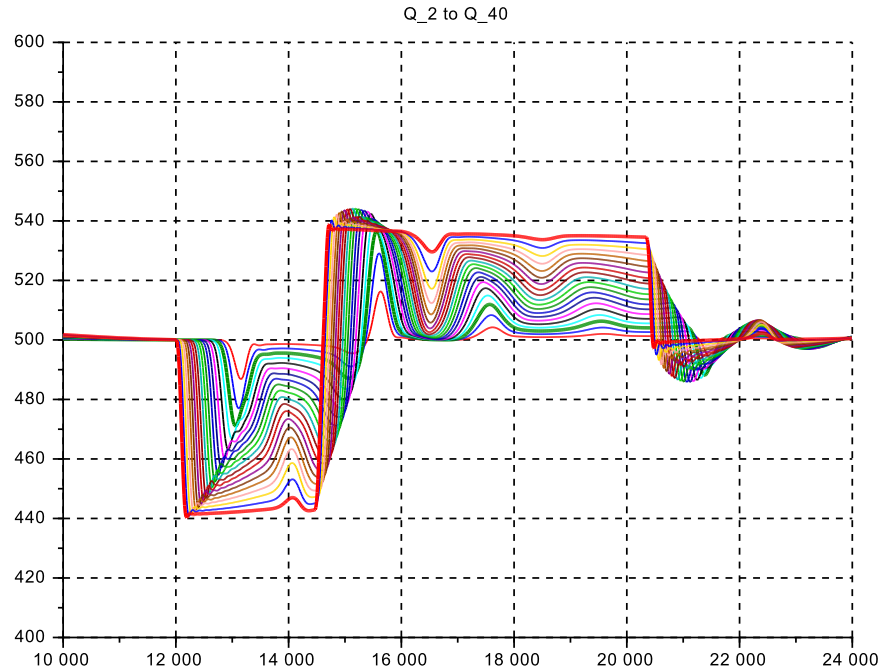
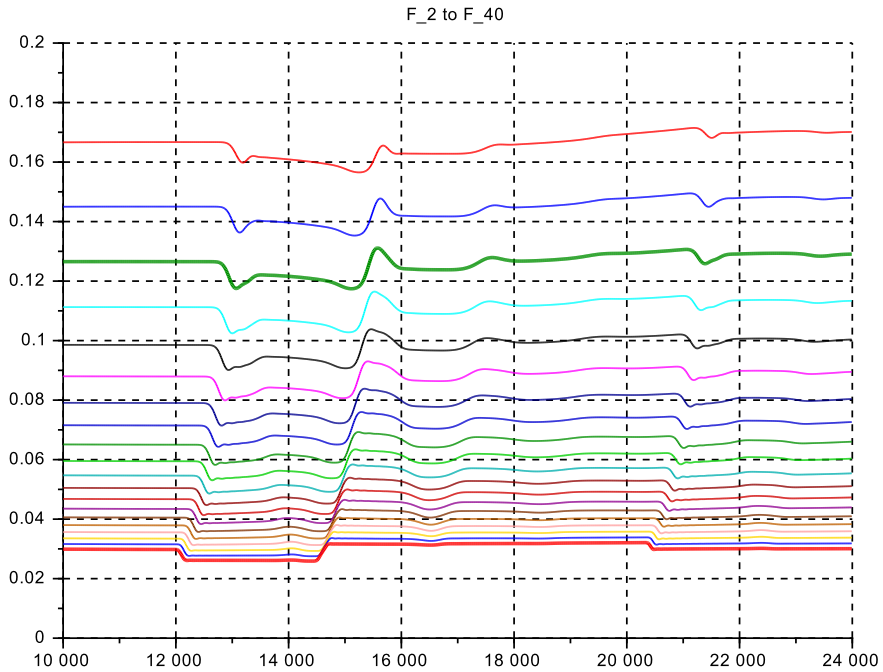
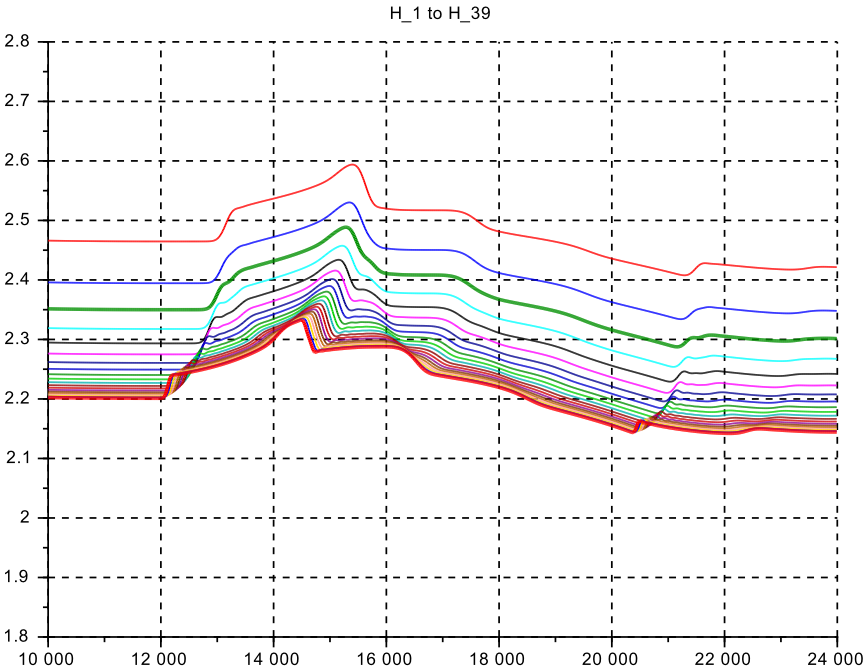
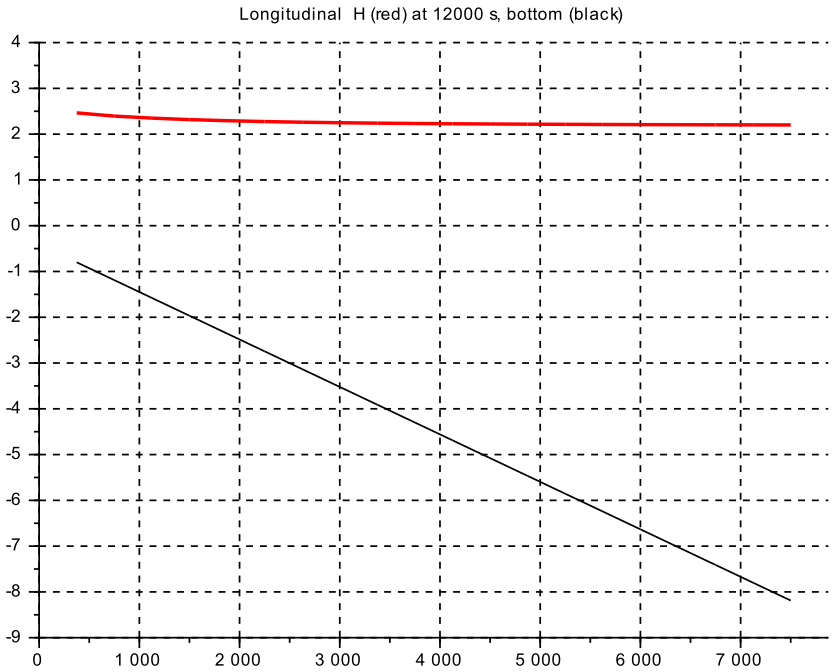

Longitudinal H (red) at 12000 s, bottom (black)
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000

Figure 3.35: $s\_c3\_41\_05\_05$ for outflow variations at $500m^3/s$, constant inflow






H_1 to H_39
2.8
2.7
2.6
2.5
2.4
2.3
2.2
2.1
2
1.9
1.8
10 000
12 000
14 000
16 000
18 000
20 000
22 000
24 000


Longitudinal H (red) at 12000 s, bottom (black)
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000

3.9.5 Discussion

- The travelling time of the wave front from the power station sensor to the second sensor (approx. $7'000m$) is from the measurement Fig.3.33 $\approx 17min = 1'020s$, and from the simulation $\approx 980s$, which agrees well considering the uncertainty on the bottom shape.

- The difference between the sensor levels in steady state at $1'000m^3/s$ is from the measurement $\approx 0.55m$ compared to the value from the simulation $\approx 0.53m$, whereas at $500m^3/s$ from the measurement $\approx 0.16m$ and from the simulation $\approx 0.15m$. Again this agrees well.

- The retention effect of the Birsfelden level control on the upriver power station is that it reduces the available head for the turbines there and thus its power output. This effect is mentioned in a vague manner at the website www.kw-birsfelden.ch. The following numbers are from the simulation

Q	D_nat	H_nat	H(1) LevCon	ΔH_nat	ΔH_LevCon
200	1.14	0.34	2.25	-1.91	-0.05
500	2.00	1.20	2.50	-1.30	-0.30
1000	3.00	2.20	3.00	-0.80	-0.80
1500	3.90	3.10	3.70	-0.60	-1.50

  The second to last column shows the head loss while the level control loop is active to the natural/free flow level and the last column shows the head loss referenced to the zero flow level at $2.20m$. This quantifies a bit better what is happening.

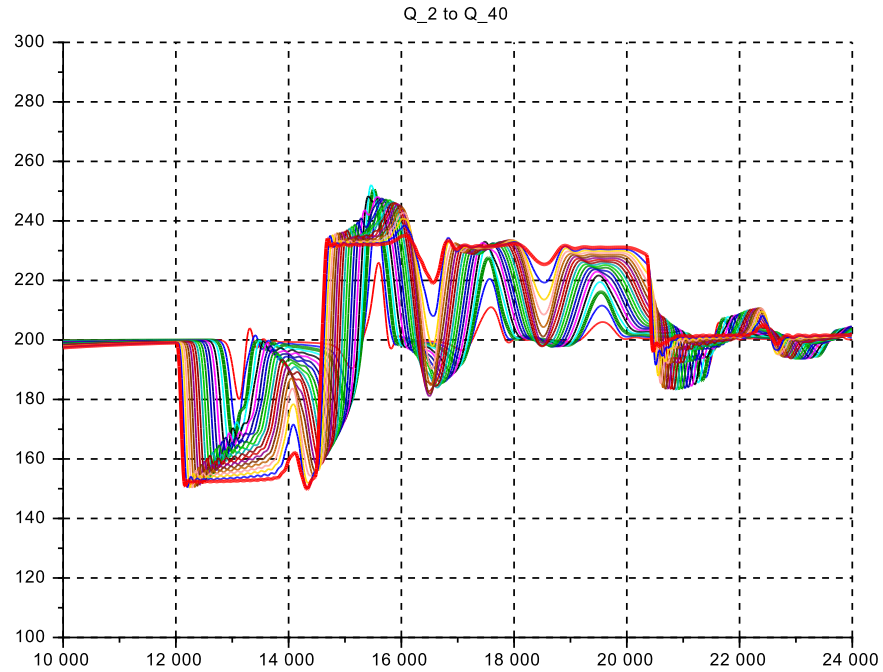
- The shape of the transients agree well for both flow conditions. There are no oscillations visible on both measured and simulated responses at $1'000m^3/s$, whereas three peaks are discernible on both sets of responses at $500m^3/s$, and there is also (after the second step up on turbine flow) a typical horizontal phase of length $\approx 1'500s$ on both sets of responses. – Thus the simulation also demonstrates a significant reduction in damping from mean flow to low flow conditions.

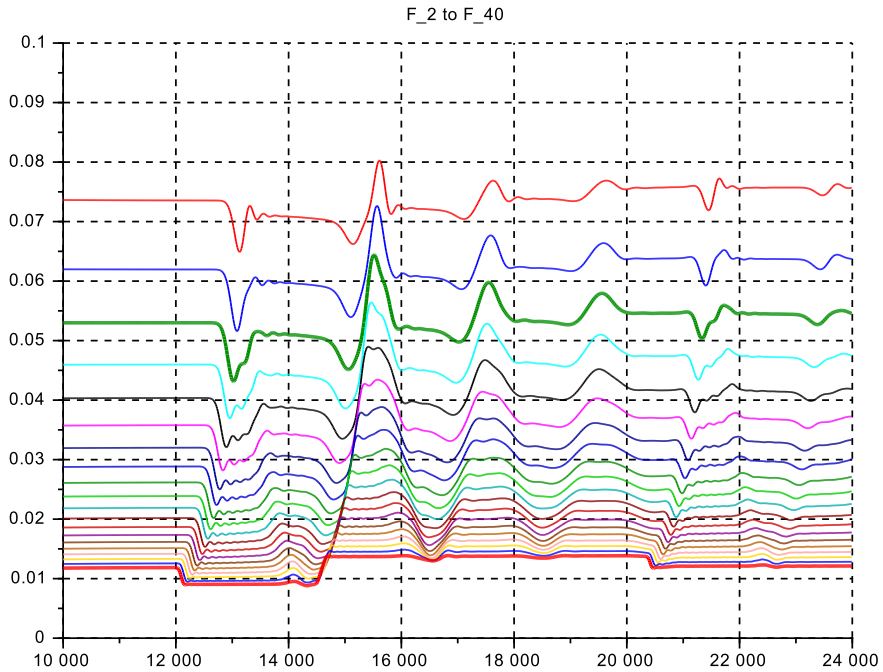
- The Froude numbers are at the lower half of the basin for $1'000m^3/s$ at 0.06...0.15 and for $500m^3/s$ at 0.03...0.08. This would indicate a much smaller damping ratio from the momentum flow effect for the situation where the impedance on the inflow end would be near infinite (meaning a hard barrier at fixed inflow).

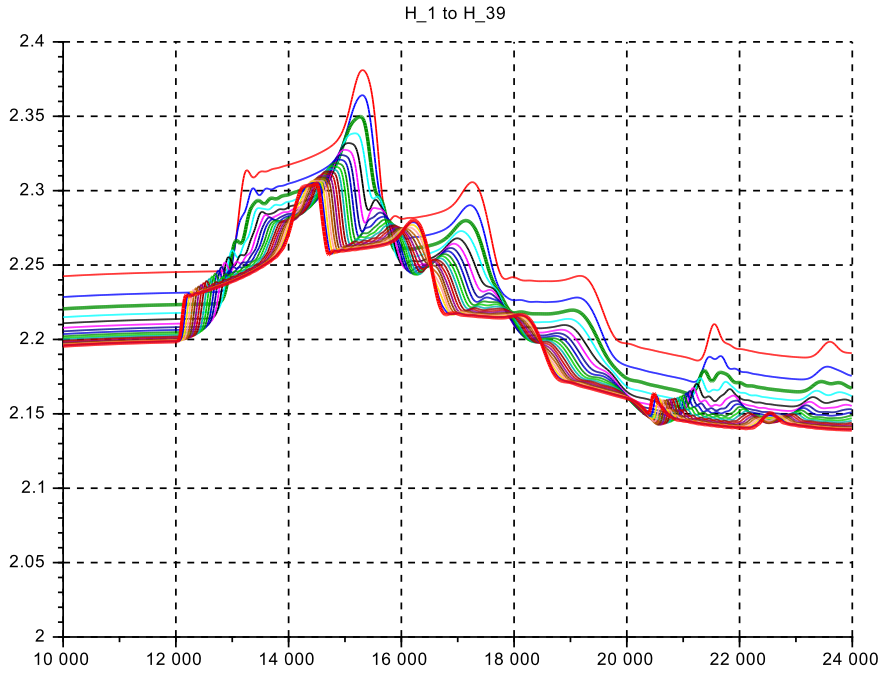
- The longitudinal profiles in $H(k)$ at $1'000m^3/s$ show a marked rise of level over the upper half of the basin, indicating a sizable 'retention volume' over the horizontal surface for zero flow. This is due to the GMS-friction, which is low at the deeper low end of the basin, but higher at the less deep upper part of the basin. In other words the wave reflection on the upper end is far from perfect, there is a finite impedance.
  Further at $500m^3/s$ this rise in level is much smaller as the friction is much smaller (Note that the slope on level $\overline{H}$ due to the GMS- friction is proportional to the square of mean flow $\overline{Q}$). And the impedance at the upper end increases. – In short the shrinking of the 'retention volume' at low flow is associated with the reduction of the damping ratio.

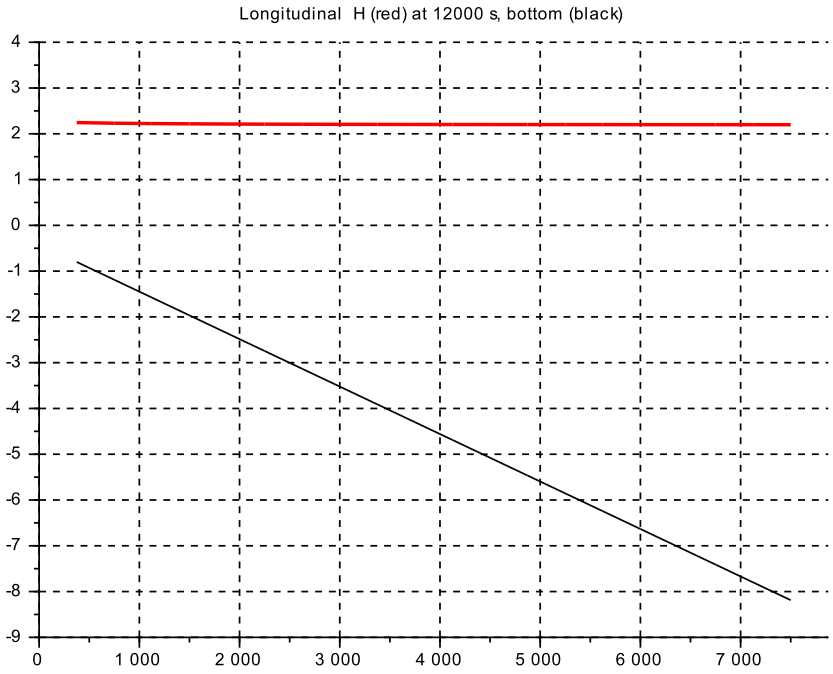
This hypothesis is tested by simulation at very low flow $200m^3/s$. The travelling wave is clearly visible in the plot of the Froude numbers F and of the levels H. The plot of F further shows how small the decrease of successive peaks of the echo waves are. The longitudinal level profile is now practically horizontal up to the upper end, the 'retention volume' approaches zero, and the impedance at the upper end approaches that of 'a hard wall'. All these findings agree well with what the above discussion of causes would predict.

Figure 3.36: $s\_c3\_41\_05\_02$ for outflow variations at $200m^3/s$, constant inflow






H_1 to H_39
2.4
2.35
2.3
2.25
2.2
2.15
2.1
2.05
2
10 000
12 000
14 000
16 000
18 000
20 000
22 000
24 000


Longitudinal H (red) at 12000 s, bottom (black)
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000

Chapter 4

The Wide Channel Model

4.1 Introduction

So far the channel geometry has been approximated to be rectangular with constant cross section along the longitudinal coordinate x. Therefore cross sectional planes stay plane along the water flow. Or flow lines are neither converging or diverging along x. This corresponds to the basic assumptions for the de Saint Venant equations.
In this chapter the river/channel geometry is assumed such that the flow components along the lateral coordinate y are no longer negligible, but become of interest as well.
Usually the rise and fall of water flow is significantly slower than the filling time constants of the system, especially if there is an upstream lake to act as a low pass filter. In other words the flow regimes are quasi stationary and a standard finite difference solver is adequate for calculating the flow pattern. However within the framework of this study the focus shall be on faster flow changes, such as a " flash flood" due to a sudden very high rain fall in the upstream area and with no upstream reservoir / low pass filter.

4.1.1 System Geometries

Two cases often arise in practice.

case A
The cross section (Fig.4.1) of the original channel is rectangular and sized for normal river flows. Then a lateral "foreland" of rectangular cross section is added, which is shallower, as the bottom rises from

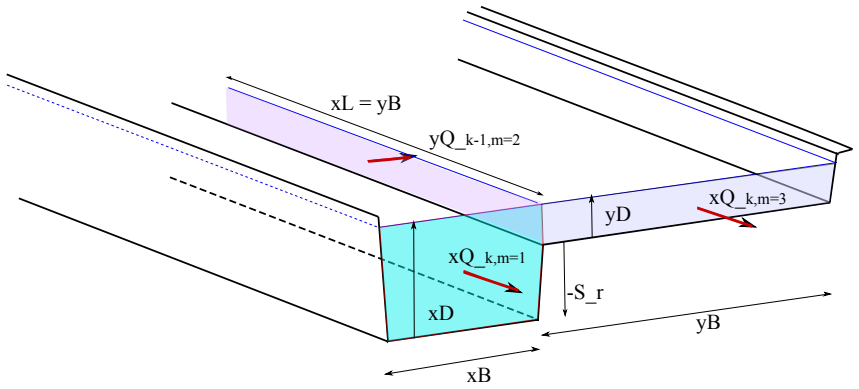


Figure 4.1: Isometric view of the geometry for case A

channel to pool. Its size determines at which flow the foreland starts to get flooded. The foreland is much

wider than the original channel bed and extends up to a dam, which runs parallel to the channel and is of sufficient height to contain an extremely high water flow. Thus the foreland creates more cross section in order to absorb very high water flows without very high water levels, and thus protects the land outside the dam from flooding.
The system considered is to be a relatively short length section from a very long river stretch, such that at equilibrium the longitudinal flows on the river bed Q_c and on the foreland Q_f are at the same water levels, $H_f := H_c$.
A rapid increase in flow will propagate faster along the channel than along the foreland until the foreland flow will catch up and the new equilibrium of longitudinal flows is reached. Thus there will be a transient lateral flow from the channel to the foreland, but will decay back to zero. Its size can be expected to be relatively small for a realistic increase of overall flow. That is the flow pattern is still essentially longitudinal.

case B
See Fig.4.2 below. Again the cross section of the original channel is rectangular and sized for normal river flows. There is a foreland as well with a given bottom rise, but it extends longitudinally only for a small part of the overall system length. Thus this foreland geometry creates a "pool" which fills only for flows above a given value. It acts as a small size storage element and thus as a low pass filter, although not a very effective one.

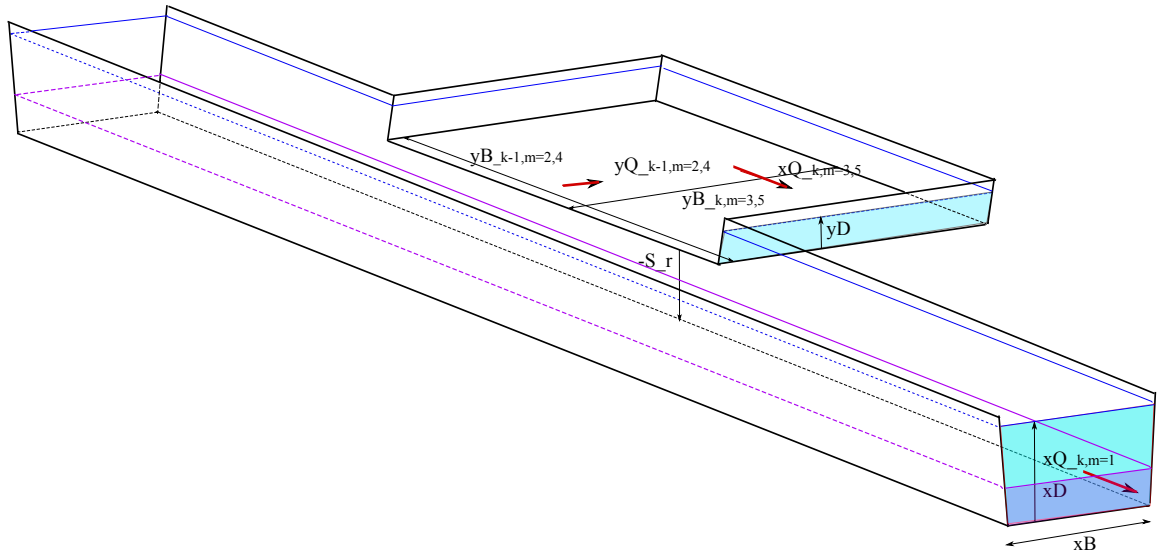


Figure 4.2: Isometric view of the geometry for case B
lines in blue: level for high flow, lines in pink: level for low flow

For this geometry the flow lines in the pool are obviously no longer parallel, but highly divergent at the inflow to the pool and convergent at the outflow. Typically large areas with 'backflow' (with reverse flow direction) are to be expected. In other words the flow pattern in the pool is no longer longitudinally dominant. This must be accounted for in the modelling and makes it more complex.
In this report however only the longitudinally dominant case A shall be considered, where lateral flows stay small compared to the longitudinal flows.

4.2 Modelling

The starting point are (as in Chap.2) the 'Reynolds Averaged Navier Stokes Equations'. The independent variables are t, x (longitudinal) and y (lateral), (with z (vertical) omitted here).
The dependent variables are u (longit. velocity), v (lateral velocity) and p (local pressure). Fluid density is ρ and ... indicate the eddy viscosity terms.

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + ... \\ \frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (vv)}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + ... \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \end{aligned}$$

Step 1

is to apply the product rule to the derivatives to x, y in the first two equations

$$\begin{aligned} \frac{\partial u}{\partial t} + 2\cdot u\cdot\frac{\partial u}{\partial x} + u\cdot\frac{\partial v}{\partial y} + v\cdot\frac{\partial u}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + ... \\ \frac{\partial v}{\partial t} + u\cdot\frac{\partial v}{\partial x} + v\cdot\frac{\partial u}{\partial x} + 2\cdot v\cdot\frac{\partial v}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + ... \end{aligned}$$

and from the third equation above for insertion in the first two equations

$$\begin{aligned} \text{either} \quad u\cdot\frac{\partial v}{\partial y} &= -u\cdot\frac{\partial u}{\partial x} \\ \text{or} \quad v\cdot\frac{\partial u}{\partial x} &= -v\cdot\frac{\partial v}{\partial y} \end{aligned}$$

That is

$$\begin{aligned} \frac{\partial u}{\partial t} + u\cdot\frac{\partial u}{\partial x} + v\cdot\frac{\partial u}{\partial y} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + ... \\ \frac{\partial v}{\partial t} + v\cdot\frac{\partial v}{\partial y} + u\cdot\frac{\partial v}{\partial x} &= -\frac{1}{\rho}\frac{\partial p}{\partial y} + ... \end{aligned}$$

Step 2

is to move to spacial discretization, see Fig.4.3.

Each cell shall be identified by index k in the longitudinal (x-) direction and by index m in the lateral (y-) direction. Each cell shall extend from bottom to surface, without any further discretization in the vertical (z-) direction. Thus there is no third index n required, as it is trivial to one for all cells.

The cell size is
seen in the x-direction: width $:= xB(k,m)$; length $:= xL(k,m)$
and in the y-direction: width $:= yB(k,m)$; length $:= yL(k,m)$

where from Fig.4.3

$$yB(k-1,m+1) = xL(k,m) \quad \text{and} \quad yL(k-1,m+1) = xB(k,m).$$

Then the three dimensional velocity functions $u(x,y,z), v(x,y,z)$ are replaced by single representative values for each cell:

$$\begin{aligned} u(x,y,z)\Big|_{k,m} &\rightarrow xU(k,m) \quad \text{for} \quad k = 0,2,4,6,..., \text{ and for} \quad m = 1,3,5,7,... \\ v(x,y,z)\Big|_{k,m} &\rightarrow yU(k,m) \quad \text{for} \quad k = 1,3,5,7,..., \text{ and for} \quad m = 0,2,4,6,... \end{aligned}$$

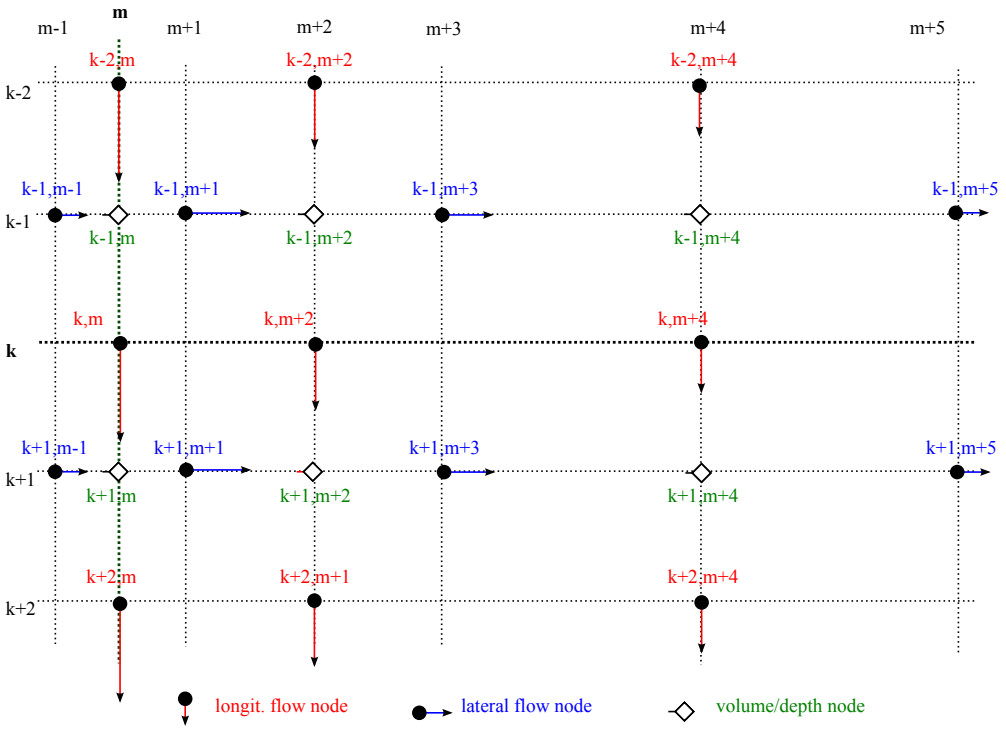


Figure 4.3: The finite element grid

As there is no vertical curvature of the flow lines and thus the associated vertical acceleration is zero, p is the local hydrostatic pressure for depth $-z$,

$$p = \rho \cdot g \cdot z$$

and the representative value is

$$P(k,m) = \rho \cdot g \cdot D(k,m)$$

Note that this will connect the momentum balance equations for x, y-directions with the volume balance equation which will yield D.

For strictly positive xU (no reverse flow is allowed here !)[1] the momentum balance equation in the positive x-direction transforms into (see also Fig.4.4)

$$\begin{aligned} \frac{d}{dt} xU(k,m+2) \quad & + \quad xU(k,m+2) \cdot \frac{xU(k,m+2) - xU(k-2,m+2)}{xL(k,m+2)} \\ & + \quad yU(k-1,m+1) \cdot \frac{xU(k-2,m+2) - xU(k-2,m)}{yL(k-1,m-1)} \\ = \quad & - \quad \frac{1}{\rho} \cdot \rho \cdot g \cdot \frac{D(k+1,m+2) - D(k-1,m+2)}{xL(k,m+2)} \end{aligned}$$

rewritten into

$$\begin{aligned} \frac{d}{dt} xU(k,m+2) \quad = \quad & + \quad g \cdot \frac{D(k-1,m+2) - D(k+1,m+2)}{xL(k,m+2)} \\ & + \quad xU(k,m+2) \cdot \frac{xU(k-2,m+2) - xU(k,m+2)}{xL(k,m+2)} \\ & + \quad yU(k-1,m+1) \cdot \frac{xU(k-2,m) - xU(k-2,m+2)}{yL(k-2,m+1)} \end{aligned}$$

[1] see [1], p.25, about expanding to model local backflow in $x-$ and $y-$ directions as well, (required for case B, Fig.4.2)

The part of the formula highlighted in red is the term which transfers momentum ('momentum cross over') from the y-axis into the x-axis. And for small values yU this term stays small compared to the second term.

And similarly for the y-axis for strictly positive xU

$$\begin{aligned}\frac{d}{dt}yU(k-1,m+1) &= + g\cdot\frac{D(k-1,m)-D(k-1,m+2)}{yL(k-1,m)} \\ &+ yU(k-1,m+1)\cdot\frac{yU(k-1,m-1)-yU(k-1,m+1)}{yL(k,m+1)} \\ &+ xU(k-2,m)\cdot\frac{yU(k-3,m-1)-yU(k-1,m-1)}{xL(k-2,m-1)}\end{aligned}$$

again with the 'momentum crossover' part in red.

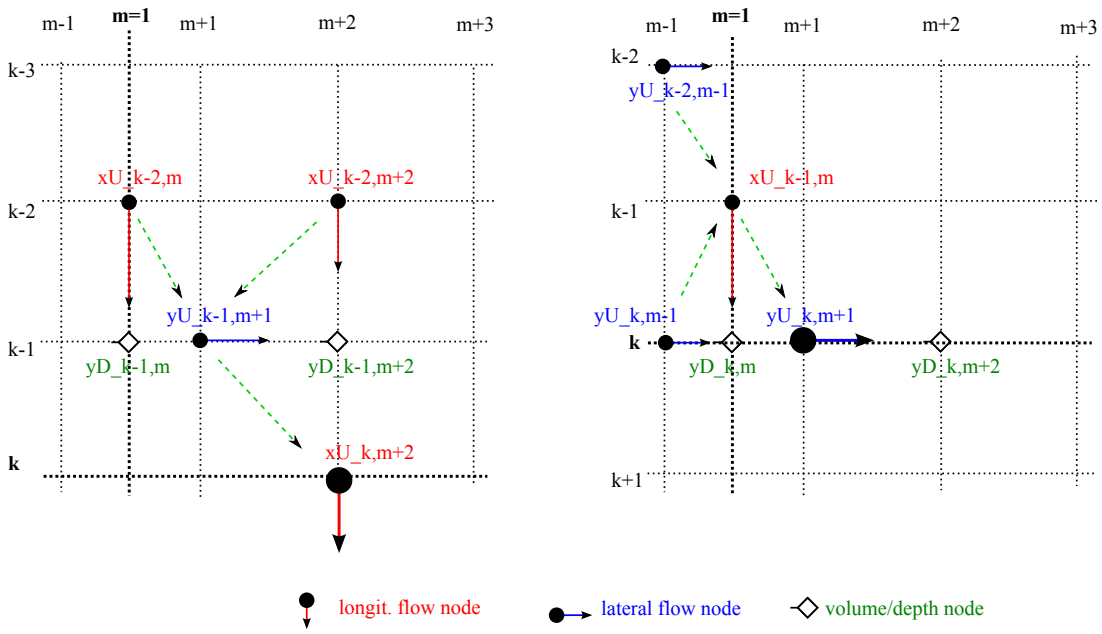


Figure 4.4: Discretization for the momentum cross over term
(left) from the y-axis to the x-axis and (right) from the x-axis to the y-axis
Note that flow directions are assumed positive on both $x-$ and $y-$ axes
.

Step 3

consists of replacing velocities xU, yU by volume flows $xQ = xU \cdot xB \cdot D, yQ = yU \cdot yB \cdot D$. This is done by multiplying both sides of the equation for the x-direction by $xL \cdot xB \cdot xD$ and in the y-direction by $yL \cdot yB \cdot yD$ [2].
In the following k is the longitudinal cell index for both the channel and the foreland, and $m = 1$ is the lateral index for the channel, combined with $n = 2, 4, ...$ creates in $m+n$ the index for subsequent lateral cells in the foreland.

[2] where $xD = xH - S$ and $yD = yH - yS$ to account for the higher bottom in the foreland

For the x-direction in the channel for $k = 2, 4, 6, ...$ and $m = 1$, where $D \rightarrow xD$:

$$\begin{aligned} & xL(k,m)\frac{d}{dt}\left(xB(k,m)\cdot xD(k,m)\right)\cdot xU(k,m) \\ = \; & +g\cdot\left(xB(k,m)\cdot xD(k,m)\right)\cdot\left[xD(k-1,m)-xD(k+1,m)\right] \\ & +\left(xB(k,m)\cdot xD(k,m)\right)\cdot xU(k,m)\cdot\left[xU(k-1,m)-xU(k+1,m)\right] \\ & +xc(k,m)\cdot yU(k-1,m-1)\cdot\left[xU(k-2,m-2)-xU(k-2,m)\right] \\ & \text{with} \\ & xc(k,m) \;=\; \frac{xB(k,m)\; xD(k,m)\; xL(k,m)}{yL(k-2,m-1)} \end{aligned}$$

and in the shallower foreland for $k = 2, 4, 6, ...$ and $m = 3$, where $D \rightarrow yD$:

$$\begin{aligned} & xL(k,m)\frac{d}{dt}\left(xB(k,m)\cdot yD(k,m)\right)\cdot xU(k,m) \\ = \; & +g\cdot\left(xB(k,m)\cdot yD(k,m)\right)\cdot\left[yD(k-1,m)-yD(k+1,m)\right] \\ & +\left(xB(k,m)\cdot yD(k,m)\right)\cdot xU(k,m)\cdot\left[xU(k-1,m)-xU(k+1,m)\right] \\ & +xc(k,m)\cdot yU(k-1,m-1)\cdot\left[xU(k-2,m-2)-xU(k-2,m)\right] \\ & \text{with} \\ & xc(k,m) \;=\; \frac{xB(k,m)\; yD(k,m)\; xL(k,m)}{yL(k-2,m-1)} \end{aligned}$$

or written in xQ in the channel for $k = 2, 4, 6, ...$ and $m = 1$, where $D \rightarrow xD$:

$$\begin{aligned} xL(k,m)\frac{d}{dt}\left(xQ(k,m)\right) \;=\; & +g\cdot\left(xB(k,m)\cdot yD(k,m)\right)\cdot\left[yD(k-1,m)-yD(k+1,m)\right] \\ & +xQ(k,m)\cdot\left[xU(k-1,m)-xU(k+1,m)\right] \\ & +xc(k,m)\cdot yU(k-1,m-1)\cdot\left[xU(k-2,m-2)-xU(k-2,m)\right] \\ & \text{with} \\ & xc(k,m) \;=\; \frac{xB(k,m)\; xD(k,m)\; xL(k,m)}{yL(k-2,m-1)} \end{aligned}$$

and in the shallower foreland/pool for $k = 2, 4, 6, ...$ and $m = 3$, where $D \rightarrow yD$:

$$\begin{aligned} xL(k,m)\frac{d}{dt}\left(xQ(k,m)\right) \;=\; & +g\cdot\left(xB(k,m)\cdot yD(k,m)\right)\cdot\left[yD(k-1,m)-yD(k+1,m)\right] \\ & +xQ(k,m)\cdot\left[xU(k-1,m)-xU(k+1,m)\right] \\ & +xc(m)\cdot yU(k-1,m+1)\cdot\left[xU(k-2,m-2)-xU(k-2,m)\right] \\ & \text{with} \\ & xc(k,m) \;=\; \frac{xB(k,m)\; yD(k,m)\; xL(k,m)}{yL(k-2,m-1)} \end{aligned}$$

The first term to the right hand side of the equation is the pressure force balance, and the second term is the balance of reaction forces due to the momentum in- and outflows. Both are familiar from chap.2 and 3. The third term is the balance of reaction forces due to momentum crossover in- and outflows.

And for the positive y-direction into the foreland cell for $k = 1, 3, 5, 7, ...$ and $m = 2$, where $D \rightarrow yD$:

$$
\begin{aligned}
& yL(k,m)\frac{d}{dt}\left(yB(k,m)\cdot yD(k,m)\right)\cdot yU(k,m) \\
= \quad & +g\cdot\left(yB(k,m)\cdot yD(k,m)\right)\cdot\left[yD(k,m-1)-yD(k,m+1)\right] \\
& +\left(yB(k,m)\cdot yD(k,m)\right)\cdot yU(k,m)\cdot\left[yU(k,m-1)-yU(k,m+1)\right] \\
& +yc(k,m)\cdot xU(k-1,m-1)\cdot\left[yU(k-2,m-2)-yU(k,m-2)\right] \\
& \text{with} \\
& yc(k,m) \;=\; \frac{yB(k,m)\;yD(k,m)\;yL(k,m)}{xL(k-1,m-2)}
\end{aligned}
$$

or written in yQ, again for $k = 1, 3, 5, 7, ...$ and $m = 2$, where $D \rightarrow yD$::

$$
\begin{aligned}
yL(k,m)\frac{d}{dt}\left(yQ(k,m)\right) \;=\; & +g\cdot\left(yB(k,m)\cdot yD(k,m)\right)\cdot\left[yD(k,m-1)-yD(k,m+1)\right] \\
& +yQ(k,m)\cdot\left[yU(k,m-1)-yU(k,m+1)\right] \\
& +yc(k,m)\cdot xU(k-1,m-1)\cdot\left[yU(k-2,m-2)-yU(k,m-2)\right] \\
& \text{with} \\
& yc(k,m) \;=\; \frac{yB(k,m)\;yD(k,m)\;yL(k,m)}{xL(k-1,m-2)}
\end{aligned}
$$

Again the first and second terms are straightforward transposes from the longitudinal terms, and the third term is the corresponding contribution from the momentum crossover flows.

Step 4

is the balance of volume flows for each cell in Fig.4.3. Its general form is, (see Fig.4.5):

$$\frac{d}{dt}V = L\cdot B\cdot\frac{d}{dt}D = \sum \text{inflows in } x\text{- and } y\text{-dir.} - \sum \text{outflows in } x\text{- and } y\text{-dir.}$$

that is for the channel cells ($k = 1, 3, 5, ...$ and $m = 1$)

$$
\begin{aligned}
\frac{d}{dt}V(k,m) \;&=\; xL(k,m)\cdot xB(k,m)\cdot\frac{d}{dt}xD(k,m) \\
&=\; \left[xQ(k-1,m)+yQ(k,m-1)\right]-\left[xQ(k+1,m)+yQ(k,m+1)\right]
\end{aligned}
$$

and for the foreland cells ($k = 1, 3, 5, ...$ and $m = 3$)

$$
\begin{aligned}
\frac{d}{dt}V(k,m) \;&=\; yL(k,m)\cdot yB(k,m)\cdot\frac{d}{dt}yD(k,m) \\
&=\; \left[xQ(k-1,m)+yQ(k,m-1)\right]-\left[xQ(k+1,m)+yQ(k,m+1)\right]
\end{aligned}
$$

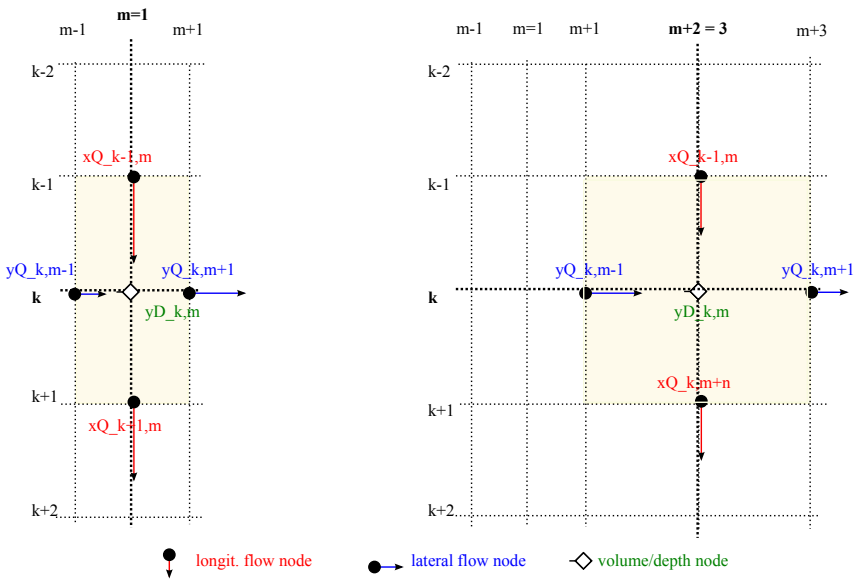


Figure 4.5: Volume balance cells with volume in- and outflows
(left) for the channel cell and
(right) for the foreland cell

Step 5

Bottom slopes:

- In the longitudinal direction of the channel, a fixed bottom slope at a typical value of $I := 1.0 \cdot 10^{-3}$ is assumed. It is applied to the foreland bottom as well, such that both run parallel downstream.
- A GMS friction coefficient with the typical value of $k_s := 31.6$ is applied for both longitudinal and lateral flow with individual parameters xk_s, yk_s. For convenience both are set to the same value in the simulations: $xk_s = yk_s = 31.6$.

- Note that on the foreland the flow direction yQ may reverse. This must be taken into account in the friction head loss and the momentum flow calculation, by replacing $yQ \cdot yU \rightarrow yQ \cdot abs(yU)$.
  Given the input value xQ and the fixed $xk_s \cdot I^{1/2} := 1.0$ establishes an equilibrium value for xD by an implicit nonlinear equation (to be computed iteratively with `iter4.sce`, see the next section 'implementation').
- The shoulder height from channel bottom to foreland bottom is computed as follows:
  For the given reference total flow value assume a given repartition of total flow between channel flow and foreland longitudinal flow, say 50 : 50. Then compute the steady state water depths for both channel and foreland areas at the inflow end, yielding xD_r, yD_r.
  The shoulder height $-\Delta S = S_r - yS_r$ is then $-\Delta S = xD_r - yD_r$. This shoulder height is of course kept constant for all other flow values.
- A laterally rising bottom slope I_{lat} is also introduced, in order to let the foreland dry out in a reasonable time interval after the total flow falls below the threshold value. Its value is set to $I_{lat} := 1.0 \cdot 10^{-3}$, again for convenience.

Step 6

Finally note that it saves significantly on model size, if the cell length yL in the foreland region is made significantly larger than the channel width xB.
In order to investigate this consider the basic model from chapter 2 applied to the lateral direction. Then the resonance period T_r of the basic oscillation mode is compared to the echo travelling time T_e of the Froude wave as a function of the multiple n of length yL_3 of the foreland cell to the channel cell $xB := yL\_1$.

The resonance period is computed first, by taking the results from the section 2.2.1 'Linearized state space form':

The time constants in this model are, with $n = 1, 2, 3, 4, 6, 8, 16, ...$

$$\begin{aligned} T_1 &= \frac{yL_1}{yU} \\ T_2 &= \frac{yL_2 \cdot yU}{g \cdot yD_2} = \frac{1+n}{2} \cdot \frac{yL_1 \cdot yU}{g \cdot yD_2}; \quad \text{where} \quad yL_2 := \frac{1+n}{2} \cdot yL_1 \\ T_3 &= \frac{yL_3}{yU} = \frac{n \cdot yL_1}{yU} \end{aligned}$$

Then the resonance frequency is

$$(\Omega_r)^2 = \frac{T_1 + T_3}{T_1 \cdot T_2 \cdot T_3} = \frac{2}{n} \cdot \left[\frac{yU_F}{yL_1}\right]^2$$

where the Froude velocity $yU_F := g \cdot yD_2$ has been inserted.

And the resonance period is

$$T_r = 2\pi \cdot \frac{1}{\Omega_r} = 2\pi \cdot \sqrt{\frac{n}{2}} \cdot \frac{yL_1}{yU_F}$$

The echo travelling time is

$$T_e = \frac{2 \cdot yL_{total}}{yU_F} = \frac{2 \cdot (1+n) \cdot yL_1}{yU_F}$$

Finally the ratio of resonance period to echo travelling time is

$$\frac{T_r}{T_e} = \frac{\pi}{\sqrt{2}} \cdot \frac{\sqrt{n}}{1+n}$$

The table collects the numerical results for $n = 1, 2, 3, 4, 6, 8, 16$:

1	2	3	4	6	8	16
1.1107	1.0472	0.9253	0.8885	0.7773	0.6981	0.5226

Note that for $n = 1, 2, 3, 4$, the fit is acceptable for practical purposes, as the error of the approximation is smaller than $\pm 12\%$. But $n > 4$ should be avoided.
Also note that in such a reduced order approximation, the actual shape of the lateral swell or sunk wave is lost. It will not be reproduced as well as with a higher order model on the lateral axis. But at least its base frequency is reproduced fairly well ...
Finally note that the formula is symmetric in $n \rightarrow 1/n$, which may be helpful in other cases.

4.3 Implementation

The implementation in `scilab/xcos` follows the same general pattern as in chapters 2 and 3, and thus needs no further explanation.
However some small changes had to be made, mainly for easier implementation.

- The basic superblock from chaper 2 and 3 implements two equations, one for the dynamic volume balance and one for the dynamic longitudinal momentum balance. This had to be split up into two basic superblocks, one for each dynamic balance plus a third superblock for the lateral dynamic momentum balance.

- The additional momentum crossflow implementations are inserted at the next level up, where all lateral blocks $m = 1, 2, 3$ at one pair of longitudinal locations (for $k-1$ and k) are assembled.

- The bottom shoulder from channel to foreland rises very steeply and not gradually as assumed in chapters 2 and 3. Therefore in the first lateral momentum balance across the shoulder ($m = 2$), the foreland bottom is virtually extended laterally at level yS_r up to the cell border at $m = 1$. The virtual depth there for the momentum balance is computed as follows:

$$\begin{aligned} xH(k,1) &= xD(k,1) + S(k,1) \\ yH(k,1) &= xH(k,1) \\ yD(k,1)_{virt} &= yH(k,1) - yS(k,2) \\ \text{and} \quad yD(k,1)_{virt} &\geq 0.0 \end{aligned}$$

- Parameter numerical values are now transmitted from the `_context.sce`-file by matrices $aP(k,m)$ instead of vectors $aV(k)$. The needed parameter values are picked up by the individual lowest level superblock by way of values for k and m inserted in its context.

- Finally note that for the lateral momentum balance $k = 1, 3, 5, ...$ at $m = 2$ the lateral velocities $yU(k, m = 0)$ and $yU(k, m = 4)$ are both zero due to the fixed ('hard') lateral cell borders.

All further details are to be found in the listings and diagrams.

Data

One motivation for this chapter has been a recent project)'Hänggelgiesse' on the 'Linth'-channel in eastern Switzerland, WGS 84 location: 47.17179, 0.01264). Therefore the main data have been taken from there, and are given in the table below.

Q_ref	360	m^3/s
$xL_{\ m=1}$	150	m
$xB_{\ m=1}$	36	m
I_r	$1.0 \cdot 10^{-3}$	-
GMS $xk_s = yk_s$	31.6	-

where the I_r-value is used both for the channel and the foreland area and both in the longitudinal and lateral directions.

4.4 Case A: The Channel and Floodplain case

4.4.1 Overview

The river geometry is illustrated in Fig.4.1.

Its cross section is asymmetric. It may be seen as one half of a symmetric one, which would be mirrored at the left hand longitudinal plane. In the real world, such symmetric (or nearly symmetric) geometries are much more frequent. However the motivation for this is a reduction of modelling effort and computation time. But an extension to a symmetric geometry is straightforward.

The floodplain is also known as foreland. It is to be 4 times larger than the channel width. This may be a bit small for real world situations, but from the previous section this is at the limit for an acceptable approximation by a one compartment model. Again this is motivated by a reduction of modelling effort and computation time.

Further the channel is to be deeper than the floodplain. For low flow ('normal') conditions the floodplain will 'fall dry' and all flow will be through the channel. For increasing flow, the water will be flooding the foreland again up to the longitudinal dam. – Thus 'falling dry' and 'flooding' will be interesting transient situations for the simulation.

As a consequence both the longitudinal flow velocity according the the GMS-law and Froude's velocity will be greater in the channel than on the floodplain. Thus an increase of Q_total will propagate faster downstream. This will create a level difference from channel to floodplain. And this drives a lateral flow from channel to floodplain. It will decay to zero when the new flow steady state regime is attained. This effect will be larger for small water depth on the floodplain, that is just after flooding. – This transient will be another interesting situation for the simulation.

The investigation will proceed as follows: The operating points and the associated boundary conditions shall be given and discussed first. Then the model properties are discussed for the inflow area, the main river part and the outflow area. Next the implementation in `Scilab/Xcos` is given, the simulation results are presented, and discussed.

4.4.2 Operating Points

Case Specific Data

$Q\_ref := Q\_r$	360	m^3/s
Q_lowlow	40	m^3/s
Q_low	120	m^3/s
$Q\_highhigh$	640	m^3/s
$yL_{\ m=2}$	90	m
$xB_{\ m=3}$	144	m
longit. inclination	-0.150	m per $xL = 150$ m
floodplain lateral inclin.	+0.072	m per $xB_3/2 = 72$m
Shoulder height ΔS_r	+1.5602	m
channel bottom S_r	-1.6322	m
foreland bottom yS_r	-0.072	m
$Q\_flood/dry\ value$	71.48	m^3/s
channel depth D_r	2.7832	m at channel flow $180m^3/s$
foreland depth yD_r	1.2230	m at foreland flow $180m^3/s$

Comments

Note that this covers two typical cases of river flows. – The first one is a river used for barge navigation with a dredged channel for seasonal low flow and a less deep area on both sides, which may be navigable by smaller boats at medium to high flow. This is suggested by the $Q\_r = 360\ m^3/s$-selection. – The second one would be a river with 'normal' flows at around 40 m^3/s, subject to frequent 'flash floods' from torrential rain, thundershowers, snow melt, etc., but with no upstream lake to buffer such large inflow rises. Then the $Q\_r = 360\ m^3/s$-selection may indicate typical cases occurring several times a year, and the $Q\_r = 640\ m^3/s$ could be the maximum for dike dimensioning...

Here for the reference flow Q_r the channel and floodplain depths are dimensioned such that the total flow splits into 180 m^3/s each. The calculation is performed with `s_c4_casA_iter.sce`:

```
// s_c4_caseA_iter.sce
// iteration for D_I
// Glf 15.12.2014

k_s = 31.6;  I_r = 0.0010;

Q_I = 180.;
// Q_I = 71.48;

  B_I = 36;
 // B_I = 144;

// initialzing 'iteration loop'
D_I = 3.0;
Q = k_s*(B_I)*(D_I)*(D_I)^(2/3)*(I_r)^(1/2);
n = 1;
sw = %T;
// error feedback gain
gainQ = 0.01*(D_I/Q_I);

// iteration
while sw  then
    R_I = (B_I*D_I)/(B_I + 2*D_I);
    Q   = (k_s*(I_r)^(1/2)*B_I)*D_I*(R_I)^(2/3);
    eQ  = -Q_I + Q;
    D_I = D_I - gainQ*eQ;
    n = n + 1;
    if eQ<-0.01 then sw =%T, elseif eQ>0.01 ...
                then sw=%T, else sw =%F, end;
end
```

It can be used to determine the channel flow values at flooding and drying as well.

The Q-values in the table have been selected as follows:
$Q = 360\ m^3/s$ as the initial value, then down to
$Q = 40\ m^3/s$ to be at one ninth of the reference flow and well below the flooding/dryfall value,
$Q = 120\ m^3/s$ to be above this value and at one third of Q_r,
$Q = 360\ m^3/s$ back to the reference value,
$Q = 640\ m^3/s$ and finally to very high flow at 16 times the lowest value (compare also to sect. 3.5).

The flow changes are performed as a sequence of ramps with stops at the given values to wait for steady state.
The 'up'-ramp is set to $+0.5\ m^3/s$ in 6 s, that is $+50\ m^3/s$ in 10 minutes. In reality this would correspond to a veritable 'flash flood' for the small changes, but may be much slower for the large changes.
And the 'down ramp' is set to $-1\ m^3/s$ in 6 s, that is $-\ 50\ m^3/s$ in 5 minutes. Note that this is about a factor of 50...100 faster than in reality for the large down-ramps, but is realistic for small changes for instance due to lock cycling.

4.4.3 Extensions to the Model

Inflow Area

Longitudinal flow change on both the channel and the floodplain must be simultaneous, in order to have a clear inflow situation for studying the flow propagation downstream both longitudinally and laterally. This can be obtained by inserting a hydropower station with weirs, turbines and possibly locks, and a tail pool side by side to the upstream end of the main area model. The outflow of this tail pool is then split to the channel and the floodplain by applying the GMS-law (see sect.2.6 and 3.1), with the common pool level as input. And both longitudinal velocities need be evaluated as well for the momentum inflow to the main area model. Finally the inflow to the tail pool is taken as *Q_total* shaped by the ramp sequence from above. Concerning the geometry of the tail pool, the overall width is set to $xB\_1 + xB\_3 = 180m$, its length to $xL = 150\ m$, and its bottom at $S = -(1.6322 + 5.0)\ m$. No bottom inclination and no friction slope are introduced. Also no longitudinal or lateral momentum balances are included. These assumptions are motivated by the larger depth of the pool.

Main Area

In the previous chapter the total length of the river flow was separated into twenty segments, each having one integrator for the volume/depth variable and one integrator for the momentum/flow variable, plus one outflow depth integrator, that is in all 41 integrators. Here a similar number of integrators is envisaged: For each segment in length there are two integrators for level and three integrators for momentum/flow (two longitudinal and one lateral) that is five integrators per segment. For eight segments, this would be 40 integrators.

Finally note that the lateral flow direction may reverse. Thus the expression for the friction head loss and the momentum flow in the momentum/flow balance must be replaced by $yQ \cdot (yU) \rightarrow (yQ) \cdot abs(yU)$ to produce the correct sign for both flow directions.

Outflow Area

In the outflow segment the two integrators for the longitudinal momentum/flow balances are replaced by two algebraic blocks implementing the GMS-law for the outflows. However the two integrators for volume/depth and one integrator for the lateral momentum/flow remain. The overall system length *L_tot* is $10 \cdot 150\ m = 1'500\ m$

With the 40 integrators for the main area plus one for the inflow pool level yields a grand total of 44 integrators.

4.4.4 Implementation

.zcos-Diagrams and -Superblocks

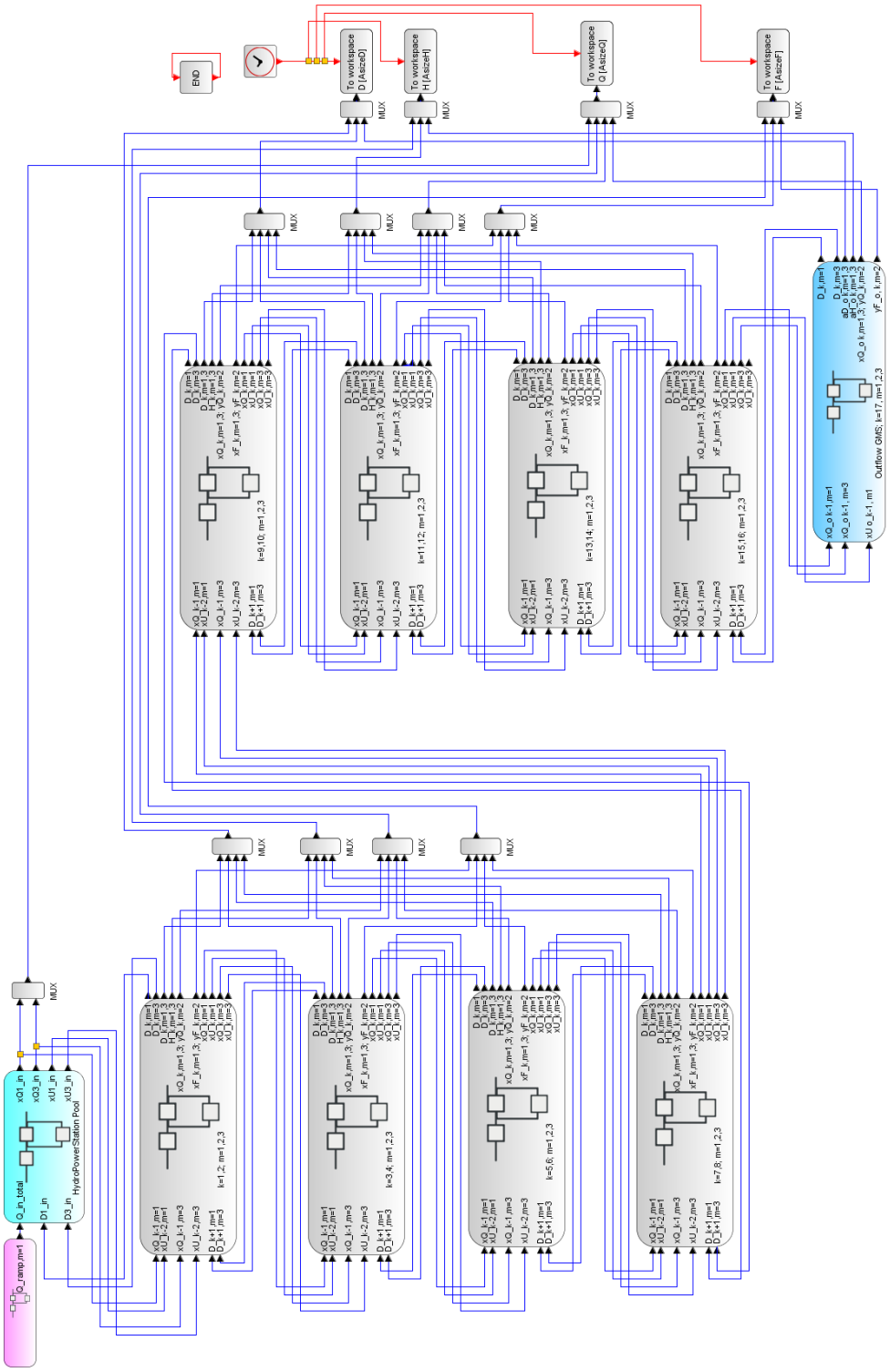


Figure 4.6: *s_c4_caseA.zcos*. zcos-Diagram, top level,
grey: 8 main area segments, dark blue; outflow segment
light blue: power station tail pool, pink: total flow generator

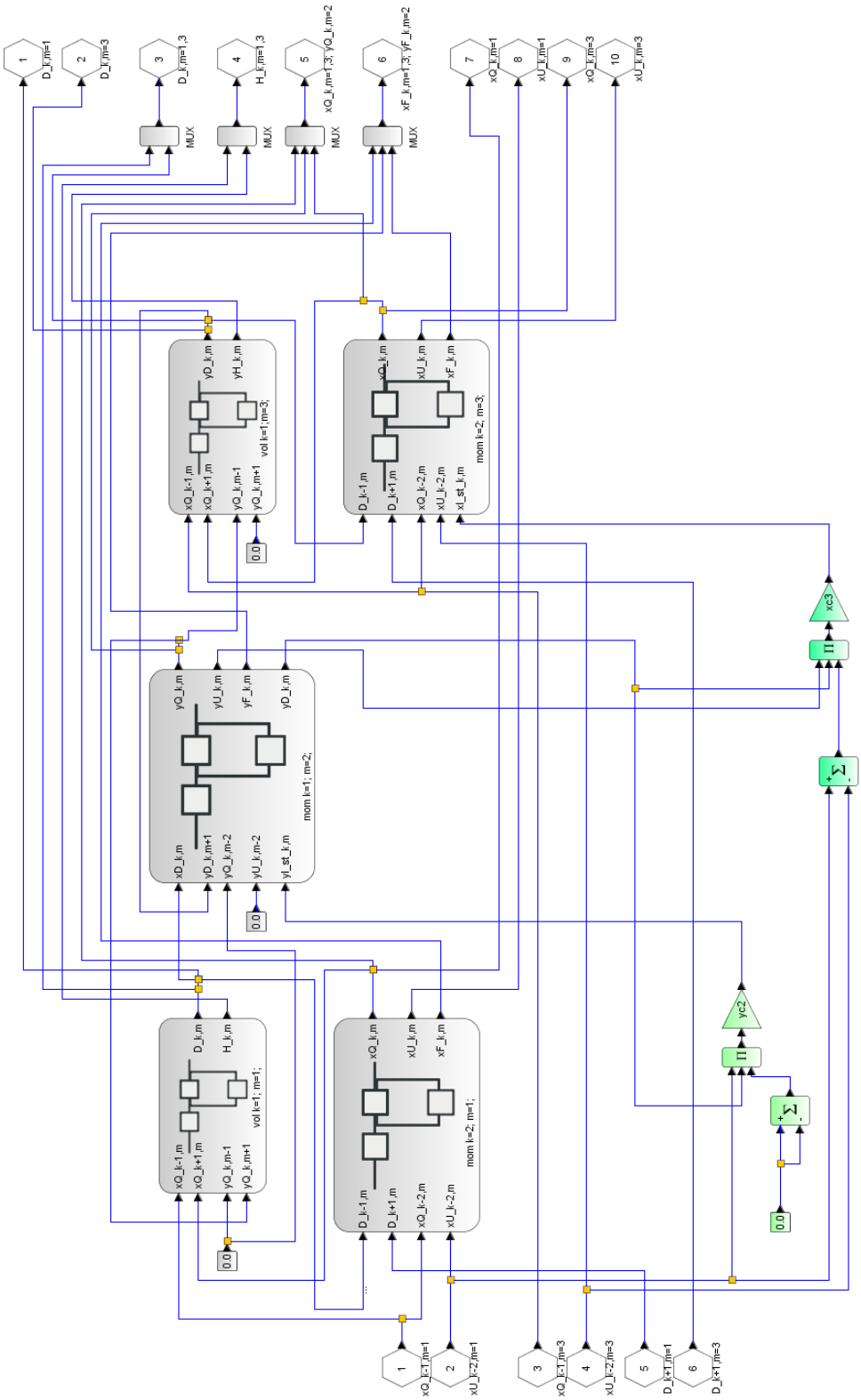


Figure 4.7: *s_c4_caseA.zcos*. first level down superblock
zcos-Diagram for one segment of main area
grey: superblocks for volume and flow balances, k, m-values are assigned in their contexts
green: momentum crossover longit. to lateral, greenblue: momentum crossover lateral to longit.

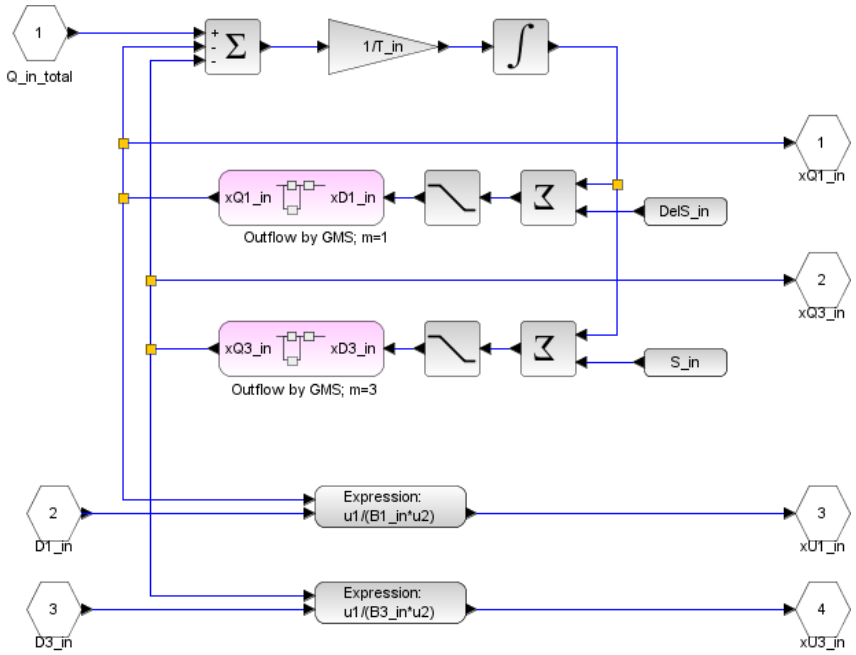


Figure 4.8: *s_c4_caseA.zcos*. first level down superblock
zcos-Diagram for the power station tail pool,
pink: superblocks for the GMS-law

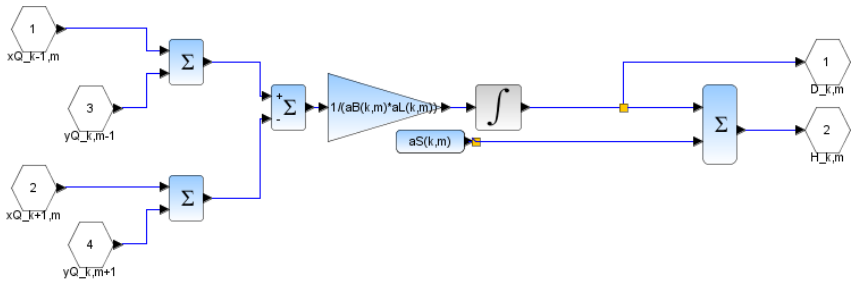


Figure 4.9: *s_c4_caseA.zcos*. second level down superblockzcos-Diagram for volume/depth balance

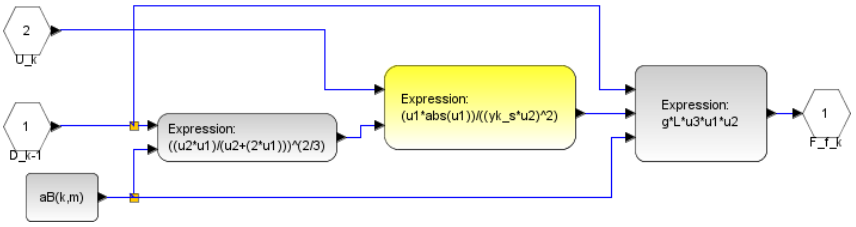


Figure 4.10: *s_c4_caseA.zcos*. third level down superblock
xcos-Diagram for calculating friction head loss in Fig.4.11
yellow: expression block for reverse flow $yU^2 \rightarrow yU \cdot abs(yU)$

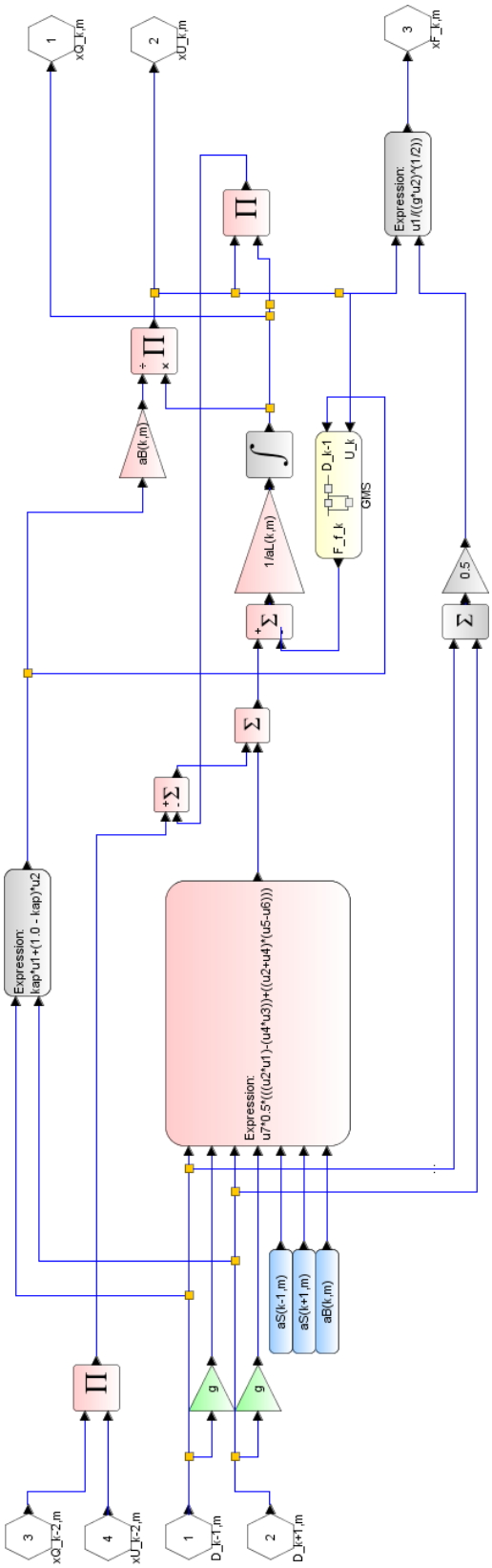


Figure 4.11: *s_c4_caseA.zcos*. second level down superblock zcos-Diagram for **longitudinal** $(x;k)$ momentum/flow balance

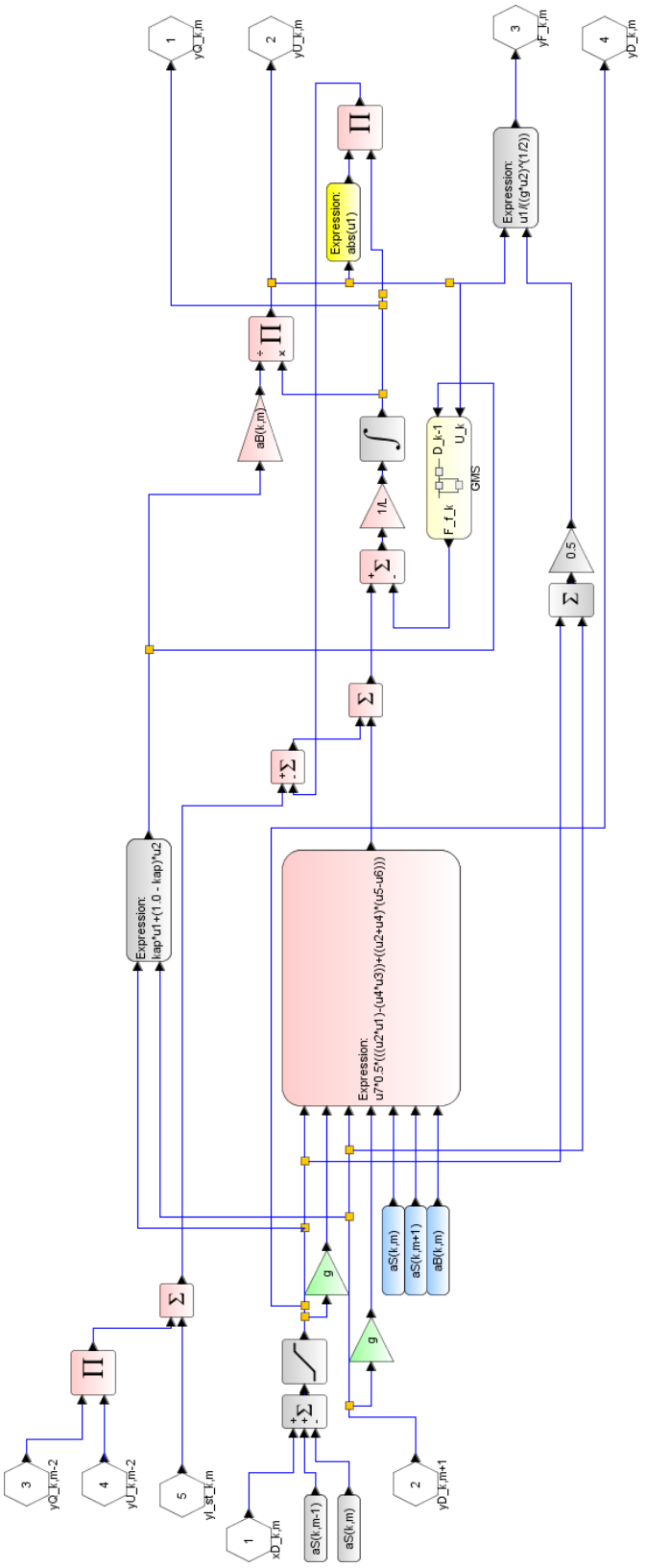


Figure 4.12: *s_c4_caseA.zcos*. second level down superblock zcos-Diagram for **lateral** $(y; m)$ momentum/flow balance

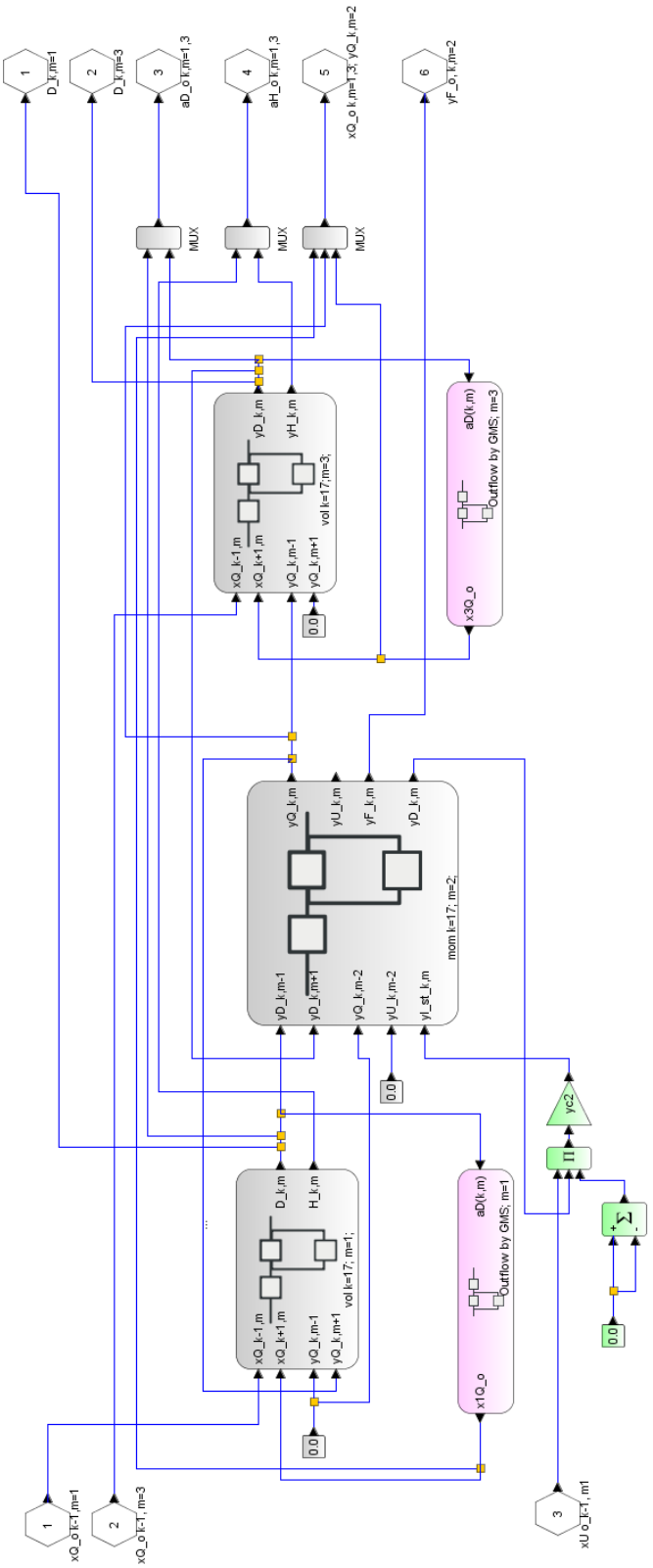


Figure 4.13: *s_c4_caseA.zcos*. first level down superblock
zcos-Diagram for **outflow segment**
pink: superblock for the GMS-law; green: momentum crossover longit. to lateral

.sce-Listings

```
// s_c4_caseA_context
// Glf   2014.12.09
// no vertical dynamics
// wide channel; lateral dynamics
// Pool at inflow with GMS

g = 10.; L = 150.; kap = 0.5; // centered
xN = 8;  // number of longit. segments
yN = 1; //  lateral extension


// GMS-friction coefficient
   xk_s = 31.6; // longit.
   yk_s = 1.0*31.6; // lateral

   xc3 = ((4*36.0*150.0)/(144.0));
   yc2 = 0.*(150.0*144.0)/150.0;

// reference bottom slope
  Q_r = 360.0; B_r = 36.0;
  D_r1 =  2.7832; D_r3 =1.2230;
  S_r = 1.6322*(-1.0); yS_r = 0.072*(-1.0);

    I_r = 0.001;

 D_min = +0.0000001;  D_max = 40*D_r1;
 Q_min = +0.0000001;  Q_max = 40*Q_r;


// channel geometry
//*****************
// Basic layout longit.: constant width
vx = 1.0*ones(1,(2*xN+1));  vxB = (B_r*vx)';
vy = 1.0*ones(1,(2*yN+1));  vyB = (L*vx)';
aB = [vxB,vyB,4.0*vxB];
vxL=(L*vx)'; vyL1=(2.5*B_r*vx)'; vyL2=(4.0*B_r*vx)';
aL = [vxL, vyL1,vxL];

//  basic layout longit.: bottom 'horizontal'
vxS0 = (S_r)*vx';
vyS1 = (yS_r+0.00)*vx'; vyS2 = (yS_r+0.072)*vx';
aS0 = [vxS0, vyS1, vyS2];

vx1D0 = (D_r1*vx)';
vy2D0=((D_r1+S_r)*vx)'-vyS1; vx3D0=((D_r1+S_r)*vx)'-vyS2;
aD0 = [vx1D0, vy2D0, vx3D0];

vx1Q0=(0.5*Q_r*vx)'; vyQ0=(Q_min*vx)'; vx3Q0=(0.5*Q_r*vx)';
aQ0 = [vx1Q0, vyQ0, vx3Q0];

// Inflow data
    B1_in = aB(1,1); B3_in = aB(1,3);
    S1_in = aS0(1,1);S3_in = aS0(1,3);
    D1_in = aD0(1,1);D3_in = aD0(1,3);
    Q1_in = aQ0(1,1);Q3_in = aQ0(1,3);

// friction slope of the bottom vS
vS0    = S_r*ones(1,(2*xN+1));
vi0    = ones(1,(2*xN+1));
vdelSf = zeros(1,(2*xN+1));
vSf    = +0.0*ones(1,(2*xN+1));
vS     = ones(1,(2*xN+1));

for kk=2:2:(2*xN),
  vI0(kk)    = I_r*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4 = 1:1:(2*xN+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end

aSf = [vSf', vSf', vSf'];
aS  = aS0 + aSf;

//  outflow data
    x1B_o = aB(2*xN+1,1);  x3B_o = aB(2*xN+1,2*yN+1);
    x1S_o = aS(2*xN+1,1);  x3S_o = aS(2*xN+1,2*yN+1);
    x1D_o = aD0(2*xN+1,1); x3D_o = aD0(2*xN+1,2*yN+1);
    x1Q_o = aQ0(2*xN+1,1); x3Q_o = aQ0(2*xN+1,2*yN+1);

// inflow pool
T_in = (B1_in+B3_in)*aL(1,1);
DelS_in = -5.0;
S_in = S_r + DelS_in; D0_in = D1_in -  DelS_in;

xg_in = xk_s*((I_r)^(1/2));


//Q down to 40 m^3/s, stepwise up
   dQ = 1.0;
t_st_1 = 1000.0;
  r_1_0 = Q_r; r_1_1=(1.0-0.8888*dQ)*Q_r;
t_st_2 = 14000.0;
  r_2_0 = 0.;  r_2_1 = +0.2222*dQ*Q_r;
t_st_3 = 20000.0;
  r_3_0 = 0.;  r_3_1 = +0.6667*dQ*Q_r;
t_st_4 = 26000.0;
  r_4_0 = 0.;  r_4_1 = +0.7778*Q_r;
t_st_5 = 35000.0;
  r_5_0 = 0.;  r_5_1 = +0.0*Q_r;
t_st_6 = 35000.0;
  r_6_0 = 0.;  r_6_1 = +0.0*Q_r;
T_fin  = 45000.0;

  // inflow slew rate
g_st = 10.0; u_up_st = 0.250; u_dn_st = -0.5;
tau_st = 3.0; Q_st_0 = Q_r;

// channel outflow by GMS
xg_o = xk_s*((I_r)^(1/2));
yg_o = yk_s*((I_r)^(1/2));

// Data transfer to Plots
CN = 4500;      // no of clockticks up to Tfin
delT = T_fin/CN;  // intervall for clock ticks
AsizeD = 1.02*(19)*CN; // size of  Data arrays
AsizeH = 1.02*(19)*CN;
AsizeQ = 1.02*(30)*CN;
AsizeF = 1.02*(26)*CN;
```

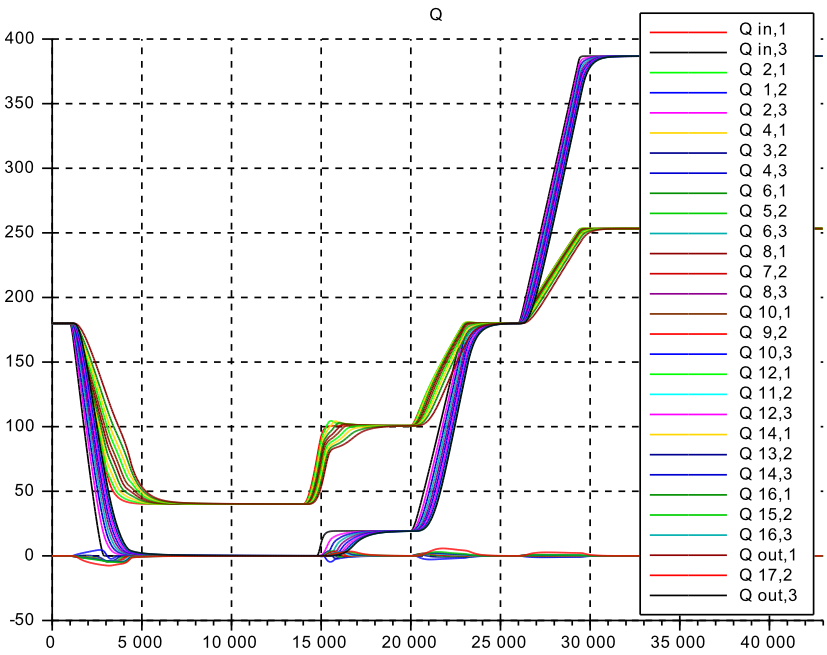
```
// s_c4_caseA_crunplot
//  Glf  2014.12.09
//  no vertical dynamics
// wide channel;  lateral dynamics
// with pool at inflow

stacksize('max');exec('s_c4_caseA_context.sce', -1);
importXcosDiagram('s_c4_caseA.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

  for kfig = 1:1:8,  clf(kfig); end

vcolor18= [ 5, 1, 3, 2, 6,32, 9,11,13,15,...
           17,19,21,22,25,27, 5, 1];
vcolor25= [ 5, 1, 3, 2, 6,32, 9,11,13,15,...
           17,19,21,22,25, 5, 2, 3, 4, 6,...
           32,19,21, 5, 1];
vcolor29= [ 5, 1, 3, 2, 6,32, 9,11,13,15,...
           17,19,21,22,25, 5, 2, 3, 4, 6,...
           32, 9,11,13,15,17,19, 5, 1];

// full view
   T0 = 0.0;   T4 = T_fin;

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor29,rect=[T0,-50.0,T4,400.]);
legend("Q in,1","Q in,3",...
 "Q  2,1","Q  1,2","Q  2,3","Q  4,1","Q  3,2","Q  4,3",...
 "Q  6,1","Q  5,2","Q  6,3","Q  8,1","Q  7,2","Q  8,3",...
 "Q 10,1","Q  9,2","Q 10,3","Q 12,1","Q 11,2","Q 12,3",...
 "Q 14,1","Q 13,2","Q 14,3","Q 16,1","Q 15,2","Q 16,3",...
 "Q out,1","Q 17,2","Q out,3",4);
xtitle("Q"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor18,rect=[T0,-0.5,T4,3.5]);
legend(...
"D  1,1","D  1,3","D  3,1","D  3,3","D  5,1","D  5,3",...
"D  7,1","D  7,3","D  9,1","D  9,3","D 11,1","D 11,3",...
"D 13,1","D 13,3","D 15,1","D 15,3","D 17,1","D 17,3",4);
xtitle("D"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor18,rect=[T0,-2.0,T4,2.0]);
legend(...
"H  1,1","H  1,3","H  3,1","H  3,3","H  5,1","H  5,3",...
"H  7,1","H  7,3","H  9,1","H  9,3","H 11,1","H 11,3",...
"H 13,1","H 13,3","H 15,1","H 15,3","H 17,1","H 17,3",4);
xtitle("H"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor25,rect=[T0,-0.30,T4,0.7]);
legend(...
"F  2,1","F  1,2","F  2,3","F  4,1","F  3,2","F  4,3",...
"F  6,1","F  5,2","F  6,3","F  8,1","F  7,2","F  8,3",...
"F 10,1","F  9,2","F 10,3","F 12,1","F 11,2","F 12,3",...
"F 14,1","F 13,2","F 14,3","F 16,1","F 15,2","F 16,3",...
"F 17,2", 4);
xtitle("F"); xgrid(1);

// Zoom-in on flooding of floodplain
  T5 = t_st_2;  T6 = t_st_3;

f5 = scf(5);
plot2d(Q.time,Q.values,vcolor29,rect=[T5,-5.0,T6,+5.0]);
legend("Q in,1","Q in,3",...
 "Q  2,1","Q  1,2","Q  2,3","Q  4,1","Q  3,2","Q  4,3",...
 "Q  6,1","Q  5,2","Q  6,3","Q  8,1","Q  7,2","Q  8,3",...
 "Q 10,1","Q  9,2","Q 10,3","Q 12,1","Q 11,2","Q 12,3",...
 "Q 14,1","Q 13,2","Q 14,3","Q 16,1","Q 15,2","Q 16,3",...
 "Q out,1","Q 17,2","Q out,3",4);
xtitle("Q  zoom-in"); xgrid(1);
//
f6 = scf(6);
plot2d(D.time,D.values,vcolor18,rect=[T5,-0.05,T6,0.35]);
legend(...
"D  1,1","D  1,3","D  3,1","D  3,3","D  5,1","D  5,3",...
"D  7,1","D  7,3","D  9,1","D  9,3","D 11,1","D 11,3",...
"D 13,1","D 13,3","D 15,1","D 15,3","D 17,1","D 17,3",4);
xtitle("D  zoom-in"); xgrid(1);
//
f7 = scf(7);
plot2d(H.time,H.values,vcolor18,rect=[T5,-1.8,T6,0.40]);
legend(...
"H  1,1","H  1,3","H  3,1","H  3,3","H  5,1","H  5,3",...
"H  7,1","H  7,3","H  9,1","H  9,3","H 11,1","H 11,3",...
"H 13,1","H 13,3","H 15,1","H 15,3","H 17,1","H 17,3",4);
xtitle("H  zoom-in"); xgrid(1);
//
f8 = scf(8);
plot2d(F.time,F.values,vcolor25,rect=[T5,-0.15,T6,0.55]);
legend(...
"F  2,1","F  1,2","F  2,3","F  4,1","F  3,2","F  4,3",...
"F  6,1","F  5,2","F  6,3","F  8,1","F  7,2","F  8,3",...
"F 10,1","F  9,2","F 10,3","F 12,1","F 11,2","F 12,3",...
"F 14,1","F 13,2","F 14,3","F 16,1","F 15,2","F 16,3",...
"F 17,2", 4);
xtitle("F  zoom-in"); xgrid(1);

// longitudinal Q-profiles at steady state
vQ=Q.values(1395,:);
for k0 = 1:1:9,
    k6 = 3*k0; k7 = k6+1; k8=k6+2;
    vQ1(k0) = vQ(k6); vQ2(k0) = vQ(k7); vQ3(k0) = vQ(k8);
end
aQlp_040 = [vQ1,vQ2,vQ3];

vQ=Q.values(2600,:);
for k0 = 1:1:9,
    k6 = 3*k0; k7 = k6+1; k8=k6+2;
    vQ1(k0) = vQ(k6); vQ2(k0) = vQ(k7); vQ3(k0) = vQ(k8);
end
aQlp_360 = [vQ1,vQ2,vQ3];

vQ=Q.values(3500,:);
for k0 = 1:1:9,
    k6 = 3*k0; k7 = k6+1; k8=k6+2;
    vQ1(k0) = vQ(k6); vQ2(k0) = vQ(k7); vQ3(k0) = vQ(k8);
end
aQlp_640 = [vQ1,vQ2,vQ3];
```

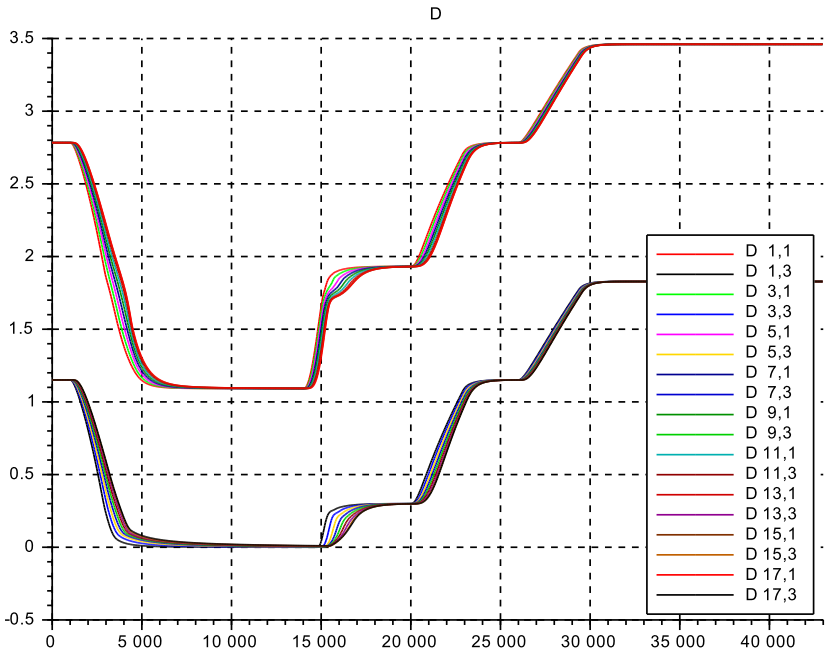
3

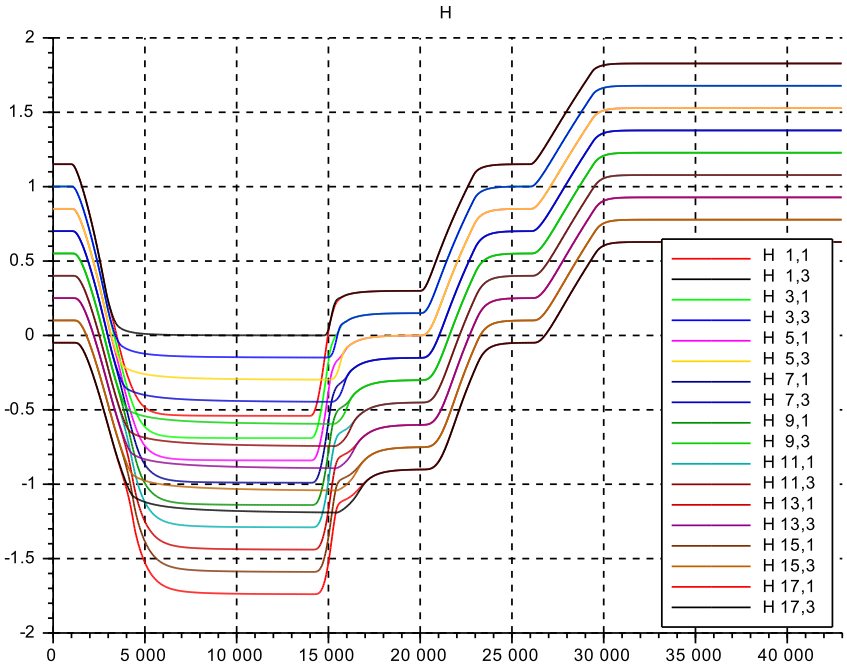
[3]Here the simulation runtime is significantly longer than for the previous cases ...

4.4.5 Transients

Figure 4.14: *s_c4_caseA*: flow variations






H
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
5 000
10 000
15 000
20 000
25 000
30 000
35 000
40 000
H 1,1
H 1,3
H 3,1
H 3,3
H 5,1
H 5,3
H 7,1
H 7,3
H 9,1
H 9,3
H 11,1
H 11,3
H 13,1
H 13,3
H 15,1
H 15,3
H 17,1
H 17,3

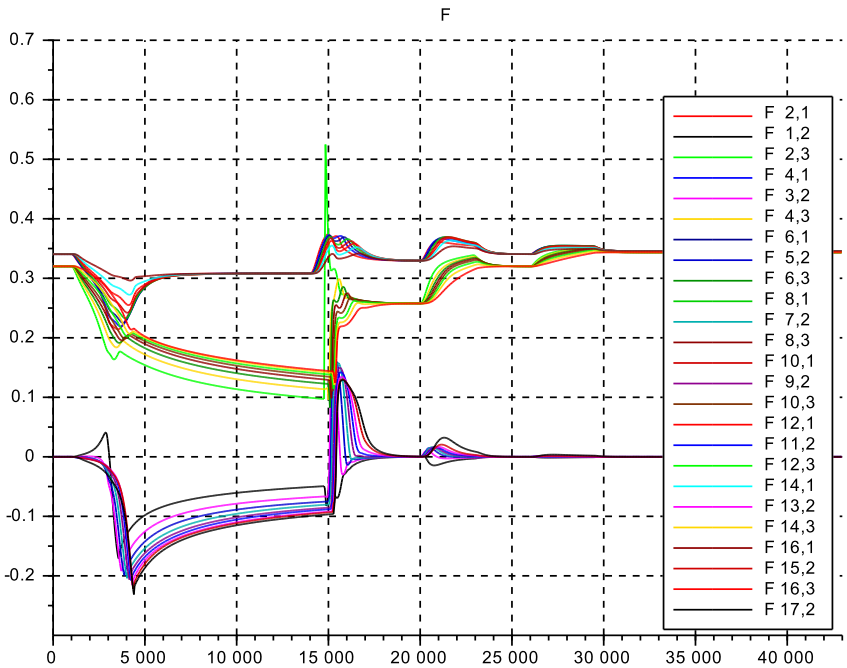
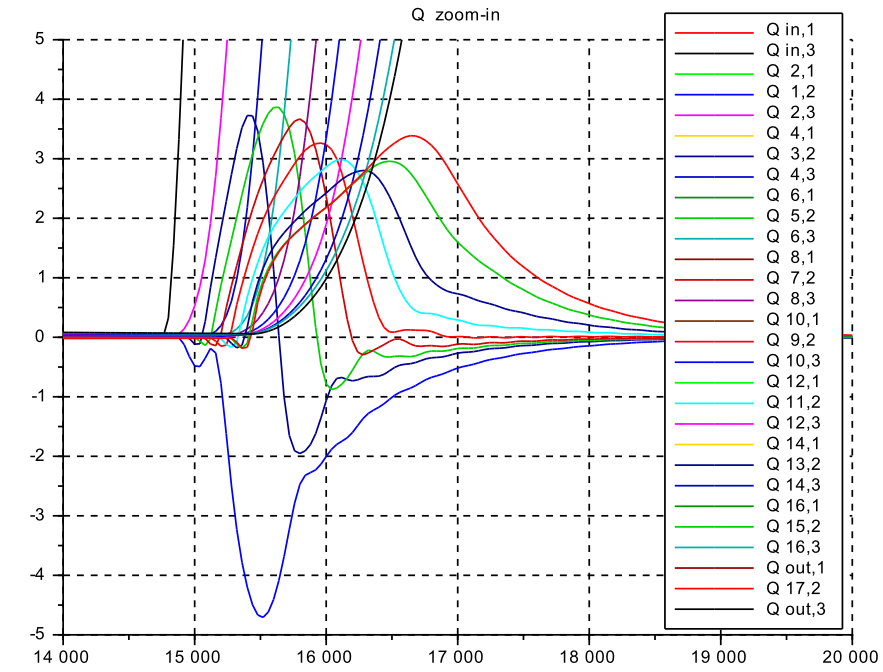
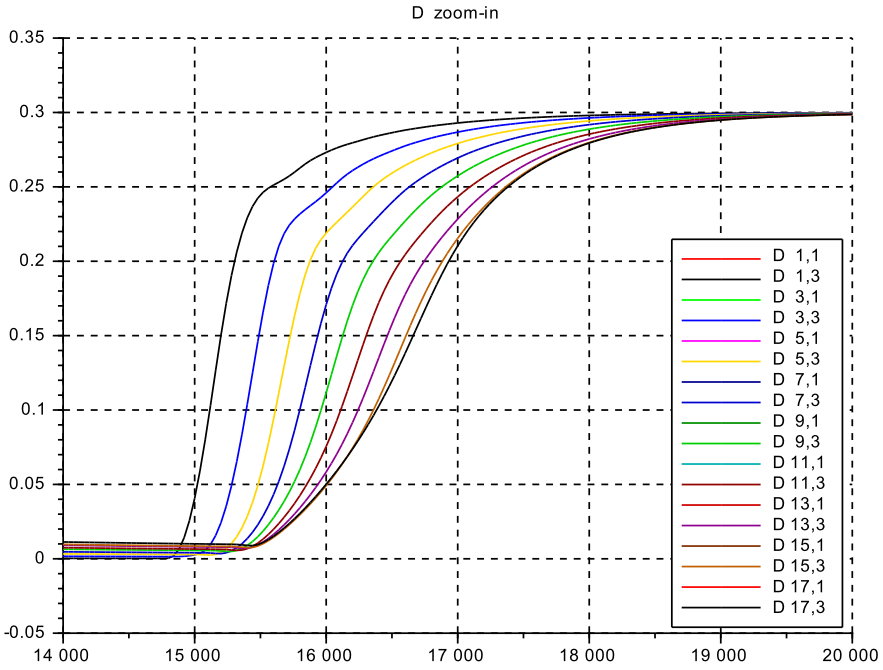
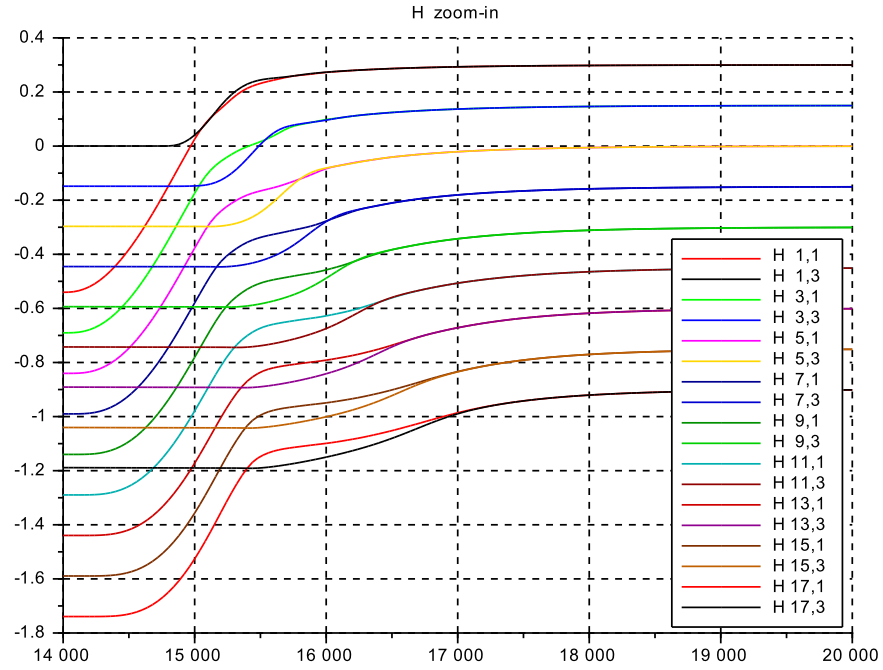
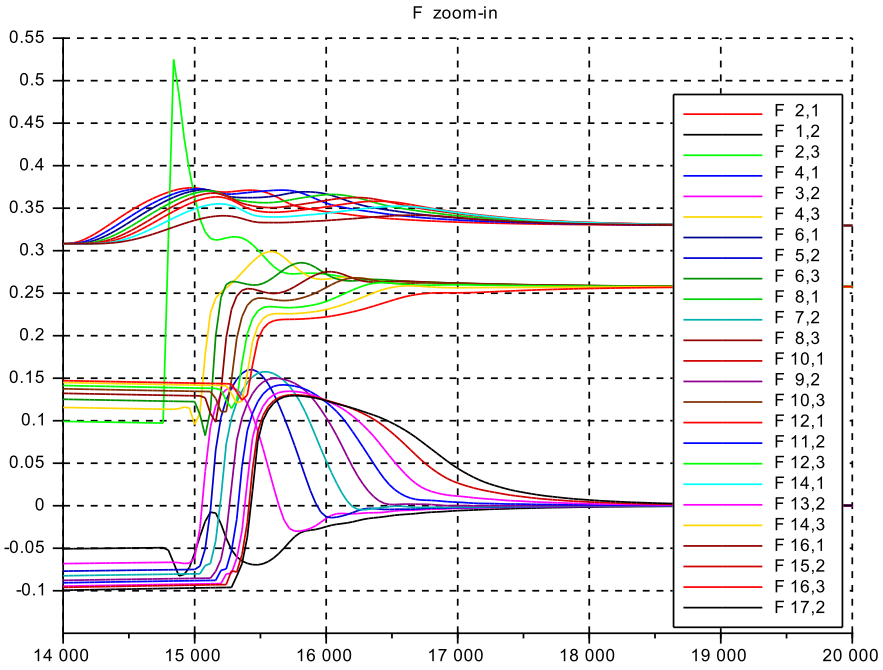

F
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
0
5 000
10 000
15 000
20 000
25 000
30 000
35 000
40 000
F 2,1
F 1,2
F 2,3
F 4,1
F 3,2
F 4,3
F 6,1
F 5,2
F 6,3
F 8,1
F 7,2
F 8,3
F 10,1
F 9,2
F 10,3
F 12,1
F 11,2
F 12,3
F 14,1
F 13,2
F 14,3
F 16,1
F 15,2
F 16,3
F 17,2

Figure 4.15: s_c4_caseA, zoom-in on flooding of floodplain






H zoom-in
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
-1.4
-1.6
-1.8
14 000
15 000
16 000
17 000
18 000
19 000
20 000
H 1,1
H 1,3
H 3,1
H 3,3
H 5,1
H 5,3
H 7,1
H 7,3
H 9,1
H 9,3
H 11,1
H 11,3
H 13,1
H 13,3
H 15,1
H 15,3
H 17,1
H 17,3


F zoom-in
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
14 000
15 000
16 000
17 000
18 000
19 000
20 000
F 2,1
F 1,2
F 2,3
F 4,1
F 3,2
F 4,3
F 6,1
F 5,2
F 6,3
F 8,1
F 7,2
F 8,3
F 10,1
F 9,2
F 10,3
F 12,1
F 11,2
F 12,3
F 14,1
F 13,2
F 14,3
F 16,1
F 15,2
F 16,3
F 17,2

Flow pattern at $Q\_tot =$ 40, 360, $640 m^3/s$

see in `s_c4_caseA_crunplot.sce` for `aQlp_xx`

```
// s_c4_caseA_FlowPattern
// 2017_06_12  Glf
```

```
 aQlp_40  =
k/m    1           2           3
 2    40.03225   - 0.0000439    0.0014255
 4    40.033372  - 0.0007975    0.0066071
 6    40.036331  - 0.0024628    0.0145703
 8    40.041551  - 0.0045047    0.0252585
10    40.05019   - 0.0076639    0.0367465
12    40.061405  - 0.0099636    0.0492359
14    40.076887  - 0.0139587    0.0610140
16    40.094247  - 0.0156300    0.0731332
18    40.116366  - 0.0203015    0.0829704
```

```
aQlp_360  =
k/m    1           2             3
 2   180.01416  - 0.0044737    180.01613
 4   180.01472  - 0.0035564    180.0082
 6   180.01407  - 0.0028524    179.99952
 8   180.01228  - 0.0021496    179.98995
10   180.00935  - 0.0013628    179.97956
12   180.00525  - 0.0004331    179.96852
14   179.99993    0.0006783    179.95717
16   179.99349    0.0018563    179.94591
18   179.97951    0.0096679    179.94187
```

```
 aQlp_640  =
k/m    1           2             3
 2   253.27887  - 0.0001957    386.76511
 4   253.27887  - 0.0002096    386.76511
 6   253.27887  - 0.0002164    386.7651
 8   253.27886  - 0.0002157    386.7651
10   253.27886  - 0.0002085    386.76509
12   253.27886  - 0.0001951    386.76509
14   253.27886  - 0.0001755    386.76508
16   253.27885  - 0.0001504    386.76508
18   253.27885  - 0.0001162    386.76509
```

4.4.6 Discussion

The flow patterns for steady state at 40, 360, 640 m^3/s show that the longitudinal flows are stable and stay very close to the inflow values. And the lateral flows are very close to zero ($|yQ\_k, 2| \leq 0.01\ m^3/s$).

During the 'flooding' transient from 40 $m^3/s \rightarrow$ 120 m^3/s, the lateral flow is $|yQ\_k, 2|_{max} \sim 4\ m^3/s$. This is about 20 % of the floodplain longitudinal steady state flow ($\approx$ 20 m^3/s).
And the transient level differences driving these lateral flows are $|\Delta H\_k, 2| \leq 0.10\ m$.

The time to peak flow is $\approx 900\ s$ and the settling time to steady state $\approx 3'600s$.

The lateral flow distribution along the channel length in the lower and outflow areas is positive (from the channel to the floodplain), as expected from the discussion above.
However in the inflow area in the first segment the flow is negative. This is due to the fact that here the delay between channel and floodplain longitudinal flows is still small. Further the level in the channel is still far below the flooding threshold and has to filled up first. This is clearly visible in the zoomed-in transient of H_k, m. – Thus both the level difference driving the lateral flow and the lateral flow itself are negative.

The high flow zoom-in transients from 120 $m^3/s \rightarrow$ 360 m^3/s and from 360 $m^3/s \rightarrow$ 640 m^3/s are not shown here for brevity. But the results shall be summarized briefly: The signs of the lateral flows are the same as above. The lateral flows magnitude are $|yQ\_k, 2|_{max} \sim 1\ m^3/s$, and the driving level differences are $|\Delta H\_k, m| < 0.01\ m$. The settling time to steady state is $\approx 1'000s$. – Thus the transients are smaller and faster, due to the higher flow levels.

To summarize, the simulation model responds as expected.

Chapter 5

The Short Channel Model

5.1 Introduction

The main feature of such short and open-ended channels is that the water surface is no longer level but is significantly inclined along the longitudinal axis x. And the inclination changes along x, that is the level surface has a significant curvature. In other words the mass along the vertical axis z must be accelerated to follow this curvature, and this needs changing the pressure $p(x)$ at its bottom from the purely hydrostatic one $p(x) = \rho \cdot g \cdot D(x)$ used up to now. Therefore this chapter will deal with added-on 'vertical dynamics'.

This shall be done by investigating several specific situations which often arise in Hydro Plants.
But first the modelling and implementation of the basic "compartments element" shall be explained in more detail in the next section. The main subject will be the spacial discretisation along the $z-$ and the $x-$axis. – The modelling of the boundary conditions both from the channel geometry and from the inflow and the outflow will be done individually for each specific case.

The first situation is a **"spillway"**. It is typically used in Hydro plants to discharge large inflows to the basin which exceed the flow through the turbines, in order to keep the basin level within bounds. It is normally located at the dam crown and may be always open or be shut by a flap, which is lowered when needed. Its outflow beyond the flap may proceed into a long channel or turn into a 'free jet' after the channel bottom ends.
Within the context here it is a 'short open ended channel'. Its geometry is an inlet confusor followed by a constant width part with an essentially horizontal bottom. Flow velocity starts at a very low value at the confusor inlet and accelerates up to near the Froude velocity U_F for the end depth D.
Two different operational modes shall be investigated. The first one is with the exit flap. Initially it is raised to a point where a small flow through the channel is maintained. The initial surface level is then approx. equal to the basin level. Then the flap is slowly[1] lowered to its fully open position which is lying flat on the channel bottom. Normally the intake level is constant irrespective of the spillway flow, due to the very large surface area of the forebay. Here this is modelled by inserting a PI-level-controller. Its setpoint is fixed at such a value that the nominal discharge (125 m^3/s, see further below) through the spillway is reached. The results are the quasi-stationary length profiles of the main variables Q, D, H, F, vertical velocity W, and bottom pressure P.
Instead of $P = \rho \cdot P^*$ the variable $\Delta P^* = P^* - g \cdot D$ will be plotted both over time t and length x. This variable shows the deviation of pressure from the hydrostatic one, and thus the additional effect of the 'vertical dynamics'.
The second operational mode is modelled by keeping the flap fully open at all times and by letting the forebay water level slowly rise at the confusor intake. This is done by continuously adjusting the inflow to the confusor by applying again the addtitional level controller of PI-type. Its setpoint is ramped up slowly from initially close above the flat part of the channel bottom to the same basin level as above.
Both operational modes are of equal importance for practical design. Therefore they are attributed a

[1] 'slowly' as compared to the filling of the channel at full flow

separate case:

(1) The first case is to be the spillway with the slowly lowered outflow flap

(2) and the second case is the spillway with fully open flap and slowly rising forebay level.

(3) The third case is a so-called **'dam break'**. This is a (hopefully never to happen) situation often investigated at the design stage for storage hydro plants. It is to estimate water levels and flows (both peak values and time evolutions) at the critical location closely downstream of the dam.
This will be modelled by using the same model as above for the spillway. However the overall basin length is increased to obtain a more realistic emptying time. Then the inflow is set to a very small fraction of the design flow ($Q_r := 50\ m^3/s$) and kept constant throughout the transient. Finally the transient will be generated by lowering the 'flap' from above to fully open, but now 'very fast' in order to catch the maximum flow values.

(4) The fourth case is a so-called **'surge wave'**. This is a very dangerous situation in river beds used as a bypass to a hydro power station. There the flow is very low in normal operation. Then the spillways may be opened raidly due to a turbine trip. This will generate a surge wave in the river bed. Also surge waves may be produced by sudden inflow increase due to torrential rain, or by emptying of a temporary lake behind an ice barrier which suddenly collapses.
The similarity to the 'dam break' case is evident. However the focus is now shifted from the upstream basin to the downstream channel. – The model for the first case is modified as follows: The length of the individual compartments is kept short to get a good view on the surge wave front. Also the outflow modelling is changed from the flap to a continuing open channel with GMS-friction. Again the initial flow states are very small. Then the inflow is stepped up 'very fast' to the nominal channel flow value ($Q_r = 50\ m^3/s$).

In all four cases so far the inflow to the model is initially very low and thus also the initial Froude numbers F at this location, $0 < F \ll 1.0$. During the transients the peak Froude numbers at the channel end will rise up to around 1.0. Thus the flow characteristics will change from *low subcritical* to *around critical*.
In contrast the next two cases will start initially from a *supercritical* flow $F_{init} \gg 1.0$. The flow is disturbed in such a way that a 'hydraulic jump' appears, where the supercritical flow transits to a subcritical one ($F_{end} \ll 1.0$). From observations on real hydro plants this transition zone is *not* very short (such as in sonic shock fronts from supersonic aircraft) but may extend over a longitudinal distance of several times the water depth D_{end}, and will show large and strong turbulence ('rollers').

(5) The fifth case is a **'hydraulic jump at a weir'**. This situation considers the flow under a weir at low partial opening and a consecutive channel with essentially horizontal bottom.
From the high upstream water level this generates a relatively thin supercritical jet down near the channel bottom where the weir opening is located.
Then the weir shall be moved up a small amount which results in a stepwise increase of jet flow. And the outflow of the channel shall be kept constant at the initial inflow value (as a simple model for e.g. turbine flow). This disturbance will result in a hydraulic jump which moves upstream with constant but low speed. – Before the jump front reaches the weir position however, the weir shall be moved down again to restore the initial inflow value. The hydraulic jump then stops moving and will remain stationary at this longitudinal position.
Thus the hydraulic jump is generated by a flow disturbance and persists after the flow disturbance is no longer there. In other words the supercritical flow regime is at an unstable equilibrium, while the supercritical-hydraulic jump-subcritical regime is a stable equilibrium.

(6) And the sixth case is a **'hydraulic jump at a low dam'**. This situation considers a dam with overfall and a slope on the downstream side (see also subsect.3.7.3), such that the overfall flow will cling to the sloped backside surface and accelerate to a supercritical velocity. The flow depth shall be such that no travelling waves are observed [2].
About halfway along the total channel length the bottom shall flatten out into being essentially horizontal. This geometric disturbance will generate a stationary hydraulic jump, but only for

[2] which are visible at very low flow, that is in a thin water layer

intermediate flow conditions. At low flows the jump will be very small and barely noticeable. And for very high flow the dam will be 'flooded' or 'submerged' and the typical hydraulic jump is no longer visible.

Again the simulation results on the dynamic model are expected to reproduce these general observations on real hydro installations. Also a reasonable approximation for the overall length of the transition zone would be nice to have. More cannot be expected, as the large scale turbulence in this zone would clearly need much more than one single compartment in the vertical direction (see next section).

5.2 Modelling

The focus is on the two compartment case, which is the building block for higher order models required for longer channels.

The starting point is (as in Chap.2 and 4) the 'Reynolds Averaged Navier Stokes Equations', with the independent variables t and x (longitudinal) and z (vertical), while y (lateral) is omitted. The dependent variables are u (longit. velocity), w (vertical velocity) and p is the local pressure, fluid density is ρ and $\ldots$ indicate the eddy viscosity terms.

momentum balances

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (uw)}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + ... \\ \frac{\partial w}{\partial t} + \frac{\partial (uw)}{\partial x} + \frac{\partial (ww)}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial z} - g + ... \end{aligned}$$

volume balance

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0$$

for $w = w^s$ along the surface contour $h(t,x)$

$$w^s = \frac{\partial h}{\partial t} + u^s \cdot \frac{\partial h}{\partial x}$$

and for $w = w^b$ along the bottom contour $s(t,x)$

$$w^b = \frac{\partial s}{\partial t} + u^b \cdot \frac{\partial s}{\partial x}$$

and as the bottom is fixed in time $\frac{\partial s}{\partial t} = 0 \quad \rightarrow \quad w^b = u \cdot \frac{\partial s}{\partial x}$

Step 1

As in chap.4 the product rule is applied to the spacial derivatives in the momentum equations. Then the volume equation is used in the same way as in chap.4, which finally leads to the following momentum equations

$$\begin{aligned} \frac{\partial u}{\partial t} + u \cdot \frac{\partial u}{\partial x} + w \cdot \frac{\partial u}{\partial z} &= -\frac{1}{\rho}\frac{\partial p}{\partial x} + ... \\ \frac{\partial w}{\partial t} + w \cdot \frac{\partial w}{\partial z} + u \cdot \frac{\partial w}{\partial x} &= -\frac{1}{\rho}\frac{\partial p}{\partial z} - g + ... \end{aligned}$$

Step 2

The next step is to move to spacial discretisation. Along the x-axis the index is again k, and i in the z-direction.
So far the momentum balance compartment in the x-direction has not been separated vertically into more than one layer. Being consistent to the models from chap.2 would mean that this assumption of one layer should be carried over to here. As a direct consequence, the momentum compartment in the z-direction should also not be divided into more than one consecutive compartments, that is it consist of just one balance for $i := 1$.
This also means that the representative longitudinal velocity U (representative values are denoted by capital letters in the following) is no longer a function of z, but constant on z:

$$\frac{\partial U}{\partial z} := 0$$

And this implies that vertical cross-sections stay vertical along x.
Further the vertical velocity function $w(z)$ is replaced by one representative vertical velocity W.

The representative values of the variables are chosen by the "finite volume" method ([3]) rather than the "finite element" method. This means that values at the interior of the element, at or very near its center of gravity, are used rather than the values on the surfaces of the element.
Specifically for the vertical velocity there are two boundary values, w^s at the surface and w^b at the bottom, which are *not* the same in the general case[3]. The value of $w(z)$ is assumed to be linear between w^b and w^s, that is

$$w(z) = w^b + \frac{w^s - w^b}{D} \cdot z \quad = \quad w^b + \zeta \cdot z \quad \text{for} \quad z = 0...D$$

which defines ζ.
The representative value for the **vertical velocity** W at location x is calculated using the mean value of the vertical momentum content as follows

$$\begin{aligned}
I(x) &= \int_0^D w(z)dm = \rho LB \int_0^D w(z)dz \\
&= \rho LB \left[w^b \int_0^D dz \;+\; \zeta \int_0^D z dz \right] \\
&= \rho LB \left[w^b \cdot D \;+\; \zeta \cdot \frac{1}{2}\, D^2 \right] \\
&= \rho LBD \left[w^b + \frac{1}{2}\left(w^s - w^b\right) \right] \\
&= \rho LBD \left[\frac{1}{2}\left(w^s + w^b\right) \right] \\
&= \rho LBD \cdot W(x) \quad \text{with} \quad W(x) := \frac{1}{2}\left(w^s + w^b\right)
\end{aligned}$$

where the representative value $W(x)$ is the arithmetic mean of $w^s(x)$ and $w^b(x)$.

And by discretising at x-location k and using the main variable D instead of H:

$$\begin{aligned}
W_k &= \frac{1}{2}\left[W_k^s + W_k^b\right] \\
&= \frac{1}{2}\left[\frac{dH_k}{dt} + U_{k-1}\left(\frac{H_k - H_{k-2}}{L_k}\right) + U_{k-1}\left(\frac{S_k - S_{k-2}}{L_k}\right)\right] \\
&= \frac{1}{2}\left[\frac{dD_k}{dt} + \frac{U_{k-1}}{L_k}\left[(D_k + 2S_k) - (D_{k-2} + 2S_{k-2})\right]\right]
\end{aligned}$$

where $dS_k/dt := 0$ is assumed.

[3]They are only equal for steady state flow in a constant cross section channel with constant bottom inclination just compensating the friction slope.

Using the finite volume [3] approach, the representative value of the **vertical momentum flow** is

$$W_k \cdot |W_k|$$

This is to be the outflow term in the vertical momentum flow balance. And the inflow term there is set to

$$W_k^b \cdot |W_k^b| \quad \text{with} \quad W_k^b = \frac{1}{2} U_{k-1} \frac{s_k - s_{k-2}}{L_k}$$

For the **local pressure** p a slightly different approach is used. The modelling in chap.2 has implicitly used the hydrostatic pressure at the bottom to calculate the longitudinal pressure force. To be consistent with this, the bottom pressure P is used here as well for the representative value. This is also the pressure which keeps the water column above it in place or lets it move up or down along the streamline trajectory. And it is further assumed that the pressure decreases linearly along z to zero at the surface.
Using

$$p^* := \frac{1}{\rho} \cdot p$$

this means

$$\left.\frac{\partial p^*}{\partial z}\right|_k \quad \rightarrow \quad -\frac{P_k^*}{D_k}$$

Step 3

Starting with the **volume balance** centered around the x-axis location at k with

$$\left.\frac{\partial u}{\partial x}\right|_k \rightarrow \frac{1}{L_k} \cdot (U_{k+1} - U_{k-1}) \text{ and } \quad \left.\frac{\partial w}{\partial z}\right|_k \rightarrow \frac{1}{D_k} \cdot \left(W^s - W^b\right) \Bigg|_k$$

Volume balance

$$\begin{aligned} 0 &= \frac{1}{L_k}\left(U_{k+1} - U_{k-1}\right) + \frac{1}{D_k}\left(W_k^s - W_k^b\right) \\ &= D_k\left(U_{k+1} - U_{k-1}\right) + L_k\left(W_k^s - W_k^b\right) \\ &= D_k\left(U_{k+1} - U_{k-1}\right) + L_k\left[\frac{d}{dt}H_k + U_k \cdot \frac{1}{L_k}\left((H_{k+1} - H_{k-1}) - (S_{k+1} - S_{k-1})\right)\right] \end{aligned}$$

that is

$$L_k \cdot \frac{d}{dt}H_k = -D_k\left(U_{k+1} - U_{k-1}\right) - U_k \cdot \left(D_{k+1} - D_{k-1}\right)$$

Using linear interpolation

$$D_k = \frac{1}{2}\left[D_{k+1} + D_{k-1}\right]; \qquad U_k = \frac{1}{2}\left[U_{k+1} + U_{k-1}\right]$$

and inserting yields after a short calculation

$$L_k \cdot \frac{d}{dt}H_k = + D_{k-1} \cdot U_{k-1} - D_{k+1} \cdot U_{k+1}$$

Introducing a constant channel width $B_{k-1} = B_k = B_{k+1}$ gives

$$B_k L_k \frac{d}{dt}H_k = +B_{k-1}D_{k-1} \cdot U_{k-1} - B_{k+1}D_{k+1} \cdot U_{k+1}$$

or finally in the form used in the previous chapters

$$\frac{d}{dt}V_k = Q_{k-1} - Q_{k+1}$$

where $d/dt(S) := 0$ is used and D, H and both Q's are directly such representative values taken at the center of the corresponding balance compartment (namely k for volume and $k-1, k+1$ for flows).

Step 4

Then for the **vertical momentum balance**

$$\frac{d}{dt}W_k + \frac{W_k \cdot |W_k| - W_k^b \cdot |W_k^b|}{D_k} + U_{k-1} \cdot \frac{W_k - W_{k-2}}{L_k} = \frac{P_k^*}{D_k} - g$$

$$\text{or} \qquad D_k \frac{d}{dt}W_k + \left[W_k \cdot |W_k| - W_k^b \cdot |W_k^b|\right] + U_{k-1}\frac{D_k}{L_k}\left[W_k - W_{k-2}\right] = P_k^* - gD_k$$

or by inserting B_k then

$$B_k D_k L_k \frac{d}{dt}W_k + B_k L_k \left[W_k \cdot |W_k| - W_k^b \cdot |W_k^b|\right] + B_k D_k U_{k-1} \cdot (W_k - W_{k-2}) := B_k L_k \left[P_k^* - g \cdot D_k\right]$$

where

$$\begin{aligned} B_k D_k L_k &= V_k && \text{is the volumetric ('mass') content} \\ B_k L_k W_k &= zQ_{k-1} && \text{is the volumetric ('mass') vertical flow} \\ B_k D_k U_{k-1} &\approx xQ_{k-1} && \text{is the volumetric ('mass') longitudinal flow, and} \\ B_k L_k \left[P_k^* - g \cdot D_k\right] & && \text{is the resulting vertical 'pressure' force} \end{aligned}$$

That is

$$V_k \frac{d}{dt}W_k := B_k L_k \left[P_k^* - gD_k\right] + xQ_{k-1}\left[W_{k-2} - W_k\right] + zQ_k\left[W_k^s - W_k^b\right]$$

Step 5

And for the **longitudinal momentum balance** starting from

$$\frac{\partial u}{\partial t} + u \cdot \frac{\partial (u)}{\partial x} = -\frac{1}{\rho}\frac{\partial p}{\partial x}$$

discretising at location $k+1$ on the x-axis

$$\frac{d}{dt}U_{k+1} = -\frac{1}{2}\frac{P_{k+2}^* - P_k^*}{L_{k+1}} - U_{k+1} \cdot \frac{(U_{k-1} - U_{k+1})}{L_{k+1}}$$

where the pressure derivative is taken 'centered' and the velocity derivative taken 'upstream', such that both directly use the representative variables.
Going to pressure forces and momentum flows on the mass element

$$\begin{aligned} V_{k+1}\frac{d}{dt}U_{k+1} &= B_{k+1}\left[\frac{1}{2}\left(P_k^* D_k - P_{k+2}^* D_{k+2}\right) + \frac{1}{2}\left(P_k^* + P_{k+2}^*\right)\left(S_k - S_{k+2}\right)\right] ... \\ &\quad + B_{k+1} D_{k+1} U_{k+1}\left[U_{k-1} - U_{k+1}\right] \\ \text{or} \qquad L_{k+1}\frac{d}{dt}Q_{k+1} &= B_{k+1}\left[\frac{1}{2}\left(P_k^* D_k - P_{k+2}^* D_{k+2}\right) + \frac{1}{2}\left(P_k^* + P_{k+2}^*\right)\left(S_k - S_{k+2}\right)\right] ... \\ &\quad + Q_{k+1}\left[U_{k-1} - U_{k+1}\right] \end{aligned}$$

where for $j = 1, 3$

$$U_{k-j} = \frac{Q_{k-j}}{B_{k-j} \cdot D_{k-j}} \approx \frac{Q_{k-j}}{B_{k-j} \cdot D_{k-j-1}}$$

again by using the upstream representative variable D_{k-j-1} instead of D_{k-j}.

To simplify matters further, the product $Q_{k+1}U_{k-1}$ is replaced by $Q_{k-1}U_{k-1}$. This introduces a (small) error in the transient but not in steady state, where Q is constant along the channel.
Then finally

$$\begin{aligned} L_{k+1}\frac{d}{dt}Q_{k+1} &= B_{k+1}\left[\frac{1}{2}\left(P_k^* D_{k+1} - P_{k+2}^* D_{k+1}\right)\right] + \\ &\quad + B_{k+1}\left[\frac{1}{2}\left(P_k^* + P_{k+2}^*\right)\left(S_k - S_{k+2}\right)\right] + \\ &\quad + \left[\frac{Q_{k-1}^2}{B_{k-1}D_{k-2}} - \frac{Q_{k+1}^2}{B_{k+1}D_k}\right] \end{aligned}$$

Now if the 'no curvature' case (see below) is considered, that is $P_j^* := gD_j$ for $j = k, k+2$ then

$$\begin{aligned} L_{k+1}\frac{d}{dt}Q_{k+1} &= gB_{k+1}D_{k+1}\left[(D_k - D_{k+2}) + (S_k - S_{k+2})\right] + \left[\frac{Q_{k-1}^2}{B_{k-1}D_{k-2}} - \frac{Q_{k+1}^2}{B_{k+1}D_k}\right] \\ &= gB_{k+1}D_{k+1}\left[(H_k - H_{k+2})\right] + \left[\frac{Q_{k-1}^2}{B_{k-1}D_{k-2}} - \frac{Q_{k+1}^2}{B_{k+1}D_k}\right] \end{aligned}$$

as in the previous chapters.

Step 6 Discussion:

The influence of the flow line 'curvature' (its spacial second derivative) on the pressure P^* will be investigated.
Steady state flow is assumed in the vertical momentum balance, with $dH_k/dt = 0$. Linking the vertical velocity to surface and bottom geometry

$$\begin{aligned} W_k &= \frac{1}{2}\left[U_{k-1}\frac{H_k - H_{k-2}}{L_k} - U_{k-1}\frac{S_k - S_{k-2}}{L_k}\right] \\ \text{and} \quad W_{k-2} &= \frac{1}{2}\left[U_{k-3}\frac{H_{k-2} - H_{k-4}}{L_{k-2}} - U_{k-3}\frac{S_{k-2} - S_{k-4}}{L_{k-2}}\right] \\ \text{Setting} \qquad & L_k = L_{k-2} = L \end{aligned}$$

$$\begin{aligned} \frac{2L}{U_{k-1}}(W_k - W_{k-2}) &= (H_k - Hk - 2) - \frac{U_{k-3}}{U_{k-1}}(H_{k-2} - H_{k-4})\cdots \\ &\quad - (S_k - S_{k-2}) + \frac{U_{k-3}}{U_{k-1}}(S_{k-2} - S_{k-4}) \end{aligned}$$

For small longitudinal velocity variations $U_{k-3} := U_{k-1} = U$

$$\begin{aligned} \frac{1}{L}(W_k - W_{k-2}) &\rightarrow \frac{1}{2}U\left[\frac{H_k - 2H_{k-2} + H_{k-4}}{L^2} - \frac{S_k - 2S_{k-2} + S_{k-4}}{L^2}\right] \\ &= \frac{1}{2}U\left[\frac{D_k - 2D_{k-2} + D_{k-4}}{L^2}\right] \end{aligned}$$

where in the first equation the first term is the 'curvature' (the second spacial derivative) of the surface contour H_{k-j} and the second is the curvature of the bottom contour. In the second equation this is collected into the curvature of depth D_{k-j}.
If the depth is constant, then $U_{k-1} = U_{k-3}$, the second derivative is zero and the pressure tends to be equal to the hydrostatic one.

A more general result for the influence of $W_k - W_{k-2}$ of P_k^* can be obtained by starting from

$$\frac{1}{L}(W_k - W_{k-2}) = \frac{1}{2}\frac{1}{L^2}\left[U_{k-1}(D_k - D_{k-2}) - U_{k-3}(D_{k-2} - D_{k-4})\right]$$

after some algebra this leads to

$$\frac{1}{L}(W_k - W_{k-2}) = \frac{1}{2}\frac{Q}{BL^2}\left[\frac{D_k D_{k-4} - D_{k-2}^2}{D_k D_{k-2}}\right]$$

Setting now D_{k-2} to the arithmetic mean of D_k, D_{k-4}, that is straight but convergent flow lines, then

$$\frac{1}{L}(W_k - W_{k-2}) = \frac{1}{2}\frac{Q}{BL^2}\cdot\frac{1}{4}\left[D_{k-4} - D_k\right]^2 \neq 0$$

Thus for convergent flow lines, $D_{k-4} > D_k$, the pressure P_k^* gets augmented to accelerate the flow.

Step 7 Preparing the Implementation

The starting point is the set of three differential equations, the first one for volume (that is state variable D) at compartment positions $k, k-2, ...$, the second for vertical momentum (for W) at the same positions, and the third for longitudinal momentum (that is Q, U) at positions $k-1, k+1, ...$

$$\begin{aligned}
L_k B_k \frac{d}{dt} D_k &= Q_{k-1} - Q_{k+1} \\
L_k B_k D_k \frac{d}{dt} W_k &= B_k L_k \left[P_k^* - g D_k\right] + Q_{k-1}\left[W_{k-2} - W_k\right] + B_k L_k \left[W_k^b \cdot |W_k^b| - W_k \cdot |W_k|\right] \\
L_{k+1} \frac{d}{dt} Q_{k+1} &= B_{k+1} \left[\frac{1}{2}\left(P_k^* D_{k+1} - P_{k+2}^* D_{k+1}\right) + \frac{1}{2}\left(P_k^* + P_{k+2}^*\right)\left(S_k - S_{k+2}\right)\right] ... \\
&\quad + \left[\frac{Q_{k-1}^2}{B_{k-1} D_{k-2}} - \frac{Q_{k+1}^2}{B_{k+1} D_k}\right]
\end{aligned}$$

There are four unknown variables, D, W, Q, P^*. Thus one further equation is needed. This is the algebraic equation which links W to the bottom contour $S(x)$ and the surface/level contour $H(x)$, see also [4].

$$\begin{aligned}
W_k &:= \frac{1}{2}\left[\frac{d}{dt} H_k + U_{k-1} \frac{H_k - H_{k-2}}{L_k} + U_{k-1} \frac{S_k - S_{k-2}}{L_k}\right] \quad \text{with} \quad H_k := D_k + S_k \\
\text{finally} \quad W_k &:= \frac{1}{2}\left[\frac{d}{dt} D_k + U_{k-1} \frac{(D_k + 2S_k) - (D_{k-2} + 2S_{k-2})}{L_k}\right]
\end{aligned}$$

And H is produced by the volume balance equation for D and the geometric relation $H_k := D_k + S_k$. Also note that the time derivative of H need not be obtained by differentiation, but can be picked up at the input to the integrator for D (as the bottom contour is not time-varying).
In other words W is not really a free variable, but is 'forced' by these geometric boundary conditions.

As W_k is thus given, then P_k^* can be obtained from the vertical momentum equation:

$$P_k^* = D_k \cdot \left[\frac{d}{dt} W_k + \frac{1}{D_k}\left[W_k \cdot |W_k| - W_k^b \cdot |W_k^b|\right] + U_{k-1} \frac{W_k - W_{k-2}}{L_k} + g\right]$$

Note that the time derivative of W must be obtained by differentiation. This shall be implemented by a high pass filter of first order $G_F(s)$. It is created by a feedback loop with forward gain ω and integral unity feedback $1/s$:

$$G_F(s) = \frac{\omega}{1 + \omega \frac{1}{s}} = s \cdot \frac{1}{1 + \frac{s}{\omega}}$$

with filter break frequency ω to be selected sufficiently above the highest harmonic of the longitudinal dynamics (see further below).

- Then P_k^* ist introduced into the longitudinal momentum equation, producing Q_{k+1}.
- This Q_{k+1} is then entered into the volume balance equation, to produce D_k and further also U_{k+1}
- and next into the geometric condition for W_k
- and last into the vertical momentum balance equation for P_k^*.
- Putting this again into the longitudinal volume balance produces a correction on dQ_{k+1}/dt

and then on Q_{k+1}, etc.

This *iteration loop* is handed over to the integration method/subroutines of the simulation package.

Step 8 Implementation of the Basic Element for $k, k+1$

The diagram Fig.5.1 shows how the 'Basic Element' is implemented in `scilab/xcos`. This will serve as the building block to assemble the channel model of 41th order, and thus produce the main superblock. The inflow and outflow generating blocks for the different case studies given in the introduction will be specified and added to this main superblock in the corresponding sections in the `*.zcos`-diagrams.
There all parameter values will be given as well in the associated `*_*.sce`-file pairs.

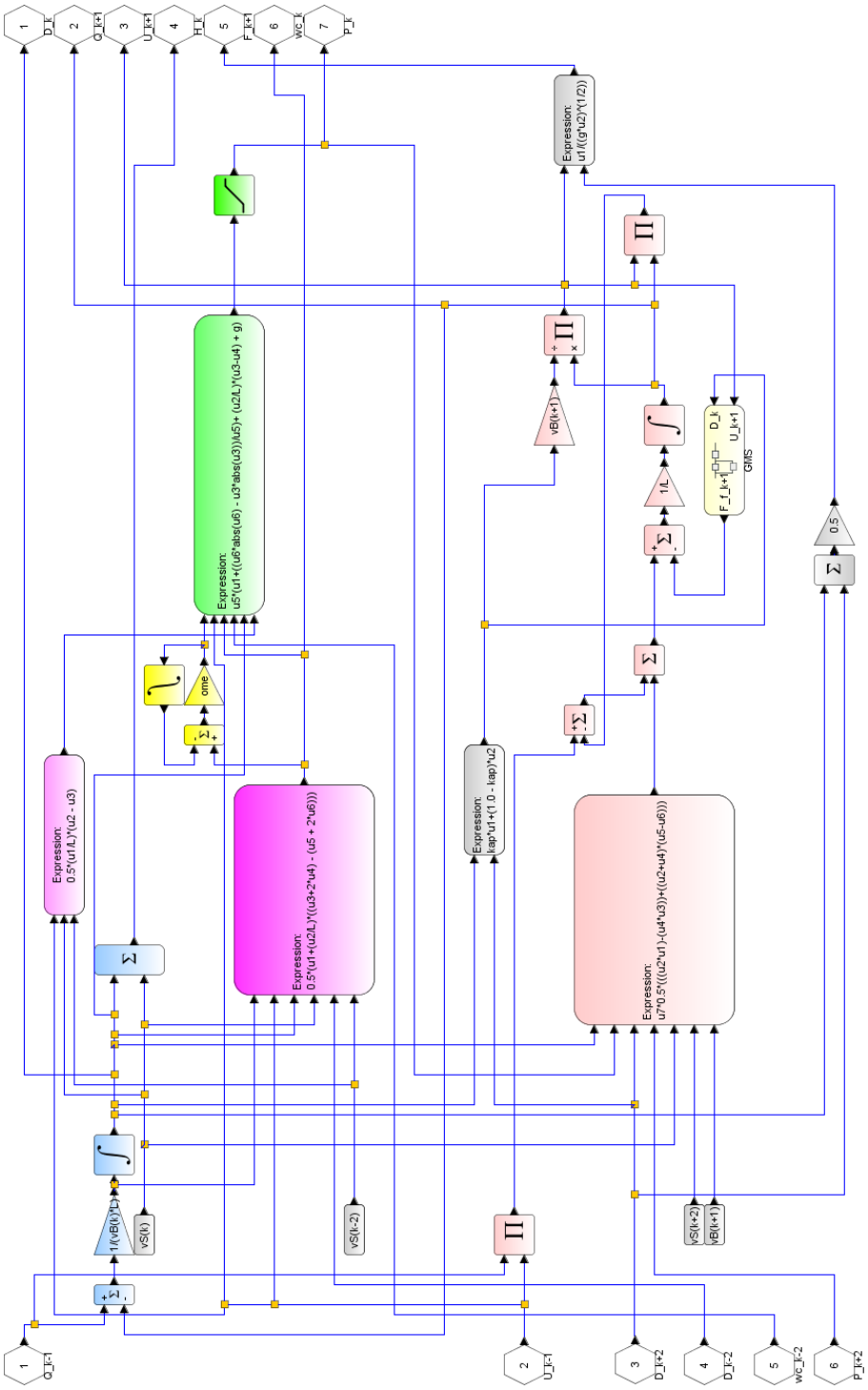


Figure 5.1: For *s_c5_0*_*.zcos* 'Basic Element', xcos-Diagram, bottom level,
red: longitudinal momentum balance for Q, light blue: volume balance for D, H
violet: W- and W_b -generators, yellow: time derivative of W green: P^*-generator

5.3 spillway cases: common material

First a 'basic result' shall be presented for steady state flow conditions and at the outflow with fully opened flap.
Then the common data set for both cases shall be given.

5.3.1 The 'basic result'

The spillway of Fig.5.2 consists of two elements, the confusor part between locations _0 to _1 and the horizontal part between locations _1 to _2 with constant cross section. The channel width in the confusor part is strongly decreasing $B\_0 \gg B\_1 = B\_2$ similar to the bottom contour $|S\_0| \gg |S\_1|$.
The aim is to find H_2 such that the outflow Q_2 is at its maximum.

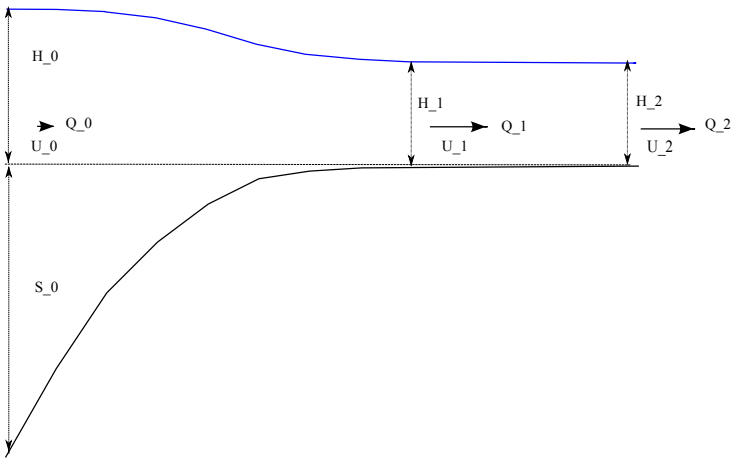


Figure 5.2: lateral view of the spillway

Then applying Bernoulli's law to the confusor part, and neglecting friction losses:

$$\frac{1}{2}U_0^2 + gH_0 = \frac{1}{2}U_1^2 + gH_1 \quad \rightarrow \quad U_1 = \sqrt{2g\,(H_0 - H_1)}$$

as $U_0 \rightarrow 0$ due to the large inflow cross section $B_0(H_0 - S_0)$

and to the second part with constant cross section, no friction and horizontal bottom

$$\frac{1}{2}U_0^1 + gH_1 = \frac{1}{2}U_2^2 + gH_2 \quad \rightarrow \quad U_1 = U_2; \;\; H_1 = H_2$$

Now the outflow is

$$\begin{aligned} Q_2 = Q_1 &= B_1 D_1 U_1 = B_1 H_1 U_1 = B_1 H_1 \sqrt{2g\,(H_0 - H_1)} \\ &= B_1 H_0 \sqrt{2gH_0} \cdot \left(\frac{H_1}{H_0}\right) \sqrt{1 - \left(\frac{H_1}{H_0}\right)} = c \cdot x \cdot \sqrt{1-x} \end{aligned}$$

$$\text{or} \quad (Q_1)^2 = c^2 \cdot x^2\,(1-x) = c^2 \cdot \left(x^2 - x^3\right)$$

For extremal $Q_1(x)$ set:

$$\begin{aligned} \frac{d}{dx}(Q_1)^2 &:= 0 \quad = c^2\left(2x - 3x^2\right) \\ \text{as} \quad c^2 &\neq 0 \\ 0 &= x \cdot (2 - 3x) \quad \text{for} \quad x = 0 \quad \rightarrow \quad \text{minimum} \quad Q \\ &\qquad\qquad \text{and for} \quad x = \frac{2}{3} \quad \rightarrow \quad \text{maximum} \quad Q \end{aligned}$$

Then

$$\begin{aligned} H_1 &= \frac{2}{3}H_0 \quad \text{that is} \quad H_0 - H_1 = \frac{1}{3}H_0 \\ \text{or} \quad H_0 &= \frac{3}{2}H_1 \quad \text{that is} \quad H_0 - H_1 = \frac{1}{2}H_1 \end{aligned}$$

That is

$$U_{1_{max}} = \sqrt{2g(H_0 - H_1)} \quad = \quad \sqrt{2g\frac{1}{2}H_1} \quad = \quad \sqrt{gH_1} \quad \stackrel{!}{=} \quad U_{1_{\text{Froude}}}$$

Note that this provides no information about the length of the acceleration zone from location _0 to location _1. From observation however this distance is about 4 to 5 times the final depth D_2 (which is here equal to H_1)

5.3.2 Common Data Sets

Basic geometry and flows

$$\begin{aligned} Q_r &= 50\ m^3/s \\ B_r &= 10\ m \\ H_r &= +2.5\ m \\ S_r &= -0.0\ m \\ U_r &= 2.0\ m/s \\ \rightarrow\ D_r &= 2.5\ m \\ \rightarrow\ U_F &= 5.0\ m/s \quad \text{and} \quad Q_F = 125\ m^3/s \\ \rightarrow\ H_0 &= 1.50 * 2.5\ m \quad = \quad 3.75\ m \end{aligned}$$

GMS-friction coefficient $k_s = 100.$

Compartment length L

Inserting the total length for the acceleration phase at $4 \cdot D_r :\approx 10m$ and letting this to be half of the total length for the 40 interleaved compartments yields $L_{tot} = 20\ m$ and thus

$$L_r := 1.0\ m$$

Thus for the acceleration phase there are 10 interpolation points for H or depth D and 10 such points for flow Q and Froude number F.

To investigate the influence of the L_r-value selection on the longitudinal profiles of D, H and Q, F, two adjacent values for L are prepared at 0.80 $L_r = 0.8\ m$ and at 1.25 $L_r = 1.25\ m$.

confusor geometry contour

The procedure for generating the contour is based on the investigation in chapter 2 where the main parameters were $\Delta B/B = const$ and $\Delta S/S = const$. This produces exponential contours. Further from chapter 2 the parameter values should not be larger than ≈ 0.20.
Simulations have shown however that this simple procedure needs to be refined: There is a non-smooth transition of the longitudinal derivative of F from the constant cross section part to such a confusor part. This is improved by letting the increments $\Delta B/B, \Delta S/S = const$ increase linearly over a given longitudinal distance from zero to its final constant value.

Further simulations have shown that this discontinuity can be reduced to a small enough value by trimming the parameters as follows:

- For the reference case $L = L_r = 1.0\ m$ the constant increments $\Delta B_k/B_k, \Delta S_k/S_k$ are set to 0.122 for the first 2.0 m of length, and then linearly decreased to zero for the next 4.0 m, that is for an overall length of 6 m or 30% of the total length of 20 m from above.
  Thus the index of arrays $vB(k), vS(k)$ for the constant increment part runs from 1 to 4. and for the linearly decreasing increment part runs from 5 to 12.

This results at the inflow location a width $B\_in \approx 24\ m$ and bottom position $S\_in = -1.4\ m$. Adding the level at the inflow location $\approx 1.5 * 2.5 = 3.75\ m$ yields an inflow depth $D\_in \approx 5.15\ m$ and thus an inflow cross section area $A\_in \approx 123.6\ m^2$. And this leads to an inflow velocity at full flow $Q = 125\ m^3/s$ of $U\_in \approx 1.0\ m/s$ or 20% of outflow velocity U_F.
Further this leads to an additional level decrease $\Delta H\_[in]$ from $U\_in = 0$ (an infinitely large upstream reservoir as assumed for the basic Euler result from above)

$$\Delta H_{in} = U_{in}^2/(2 \cdot g) \approx 0.050\ m \quad \text{or} \quad \approx 4\ \% \quad \text{of the total level difference from inflow to outflow}$$

Thus the deviation from the basic Euler case is small enough to be neglected. Therefore no wider inflow width B_in and inflow bottom position S_in are considered.

Finally the simulations have shown that the contour thus generated should be independent of the current compartment length L. This has been achieved here by trimming the increment value and the lengths for constant and linearly decreasing increments as follows:

- for 0.80 L_r the constant increment is set to := 0.100 over length index 1 to 6 and linearly decreasing over length index 7 to 16, that is an overall length $16 * 0.40\ m = 6.4\ m$.
- and for 1.25 L_r the constant increment is set to := 0.144 over length index 1 to 3 and linearly decreasing over length index 4 to 9, that is an overall length of 5.625 m.

Note that the integer nature of the index does not let to achieve exactly the same contour for all three cases within the given length of 6 m. This would require a confusor length of 10 m. However the simulations again have shown that this is too long compared to the finite overall model length, especially so for short values of $L_r \leq 0.80\ m$ where the overall model length is 16 m ...

This procedure is implemented in the `_contour.sce`-file, and the resulting contours are documented in Fig.5.3.

Filtered derivative, *ome*

The reference for selecting the bandwidth *ome* for the filter is the typical frequency of the highest resonance mode ω:

$$\omega = \sqrt{\frac{2gD_{typ}}{L^2}}$$

Let $D_{typ} := 3.20\ m$ and $L := 0.80\ m$, then $\omega = 10\ [rad/s]$ and thus

$$ome := 10 \cdot \omega = 100\ [rad/s]$$

Designing the Level Controller gains gQ_i and T_Q_i

The inflow Q_in is rising with approx 1.0 m^3/s from near zero to full flow $Q = 125\ m^3/s$, that is in about 2 minutes.
The design proceeds in three steps. First the integral action gain is designed to provide a small enough steady state error along the flow ramp up. Then the proportional gain is determined to give reasonable closed loop dynamics. Finally the result is checked for the actual plant dynamics.

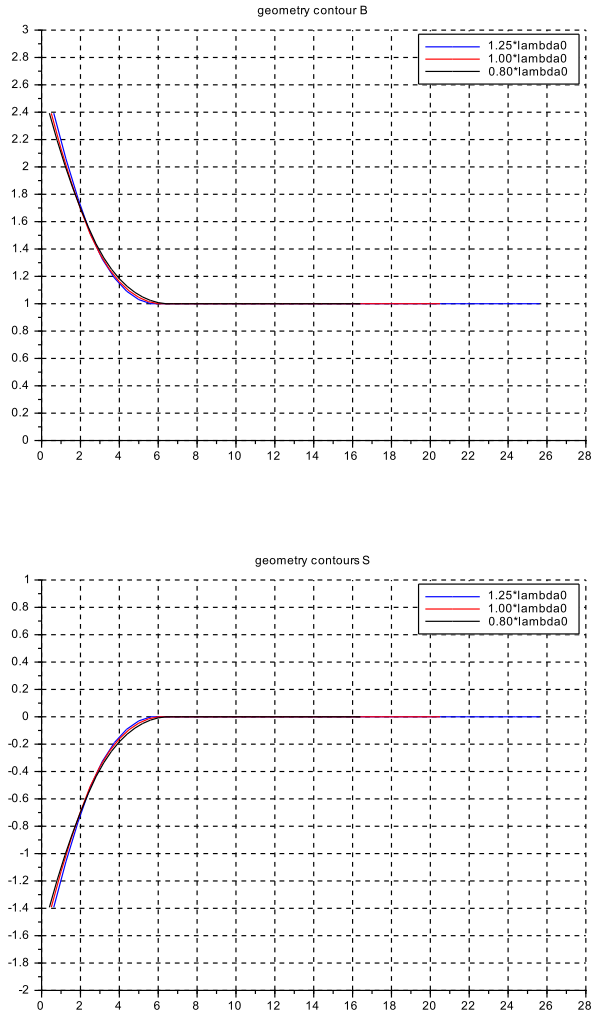


Figure 5.3: Width contour B and bottom contour S for the three cases of L_r

1. With a steady state control error $\overline{e}$ at the input to the integral action the output rising given is attained by

$$\frac{\overline{dQ\_in}}{dt} := 1.0 = \overline{e} \cdot \frac{1}{T\_Q\_i}$$

   Thus setting $T\_Q\_i := 0.010$ yields $\overline{e} := 0.01\ [m] = 1\ [cm]$

2. The 'plant' dynamics is assumed to be of first order open integrator type with time constant T_in (this is checked in the third step below), where

$$T\_in = \frac{L_r}{U\_in} := 1.0\ [s]$$

   using $L_r = 1.0\ m$ and $U\_in = 1.0\ [m/s]$ from the inlet confusor geometry design above. then from the standard closed loop pole placement procedure at Ω and 2 ζ

$$\begin{aligned} \frac{1}{T\_Q\_i} &= \Omega^2 \cdot T\_in \quad \rightarrow \quad \Omega = 10 \\ gQ\_i &= 2 \cdot \zeta \cdot \Omega \cdot T\_in \quad \rightarrow \quad gQ\_i \approx 20 \end{aligned}$$

3. In fact the 'plant' is of third order (see chapter 2).
   However the gain gQ_i is high, which reduces the time constant of the first volume integrator from $T\_1 = 1.0\ s$ to $T\_1 \quad \rightarrow \quad 1/20\ s$.
   Also the resonance frequency (see chapter 2) increases to

$$\omega = \sqrt{\frac{T_1 + T_3}{T_1 T_2 T_3}} \quad \rightarrow \quad \omega' \approx 4.5 \cdot \omega = 36\ [rad/s]$$

   Thus both the first volume dynamics (T_1) and the momentum dynamics (ω') can be considered as 'may be neglected' and the second volume dynamic is dominant. In other words the 'plant' can be approximated as one open integrator (this is valid for low enough frequencies only).

Note that this is a very approximative design, as the $\Omega = 10\ [rad/s]$-value from step 2 is not that far from the 'neglected' dynamics at 20 $[rad/s]$ and 36 $[rad/s]$.

5.4 Case 1: Spillway with slowly lowered outflow flap

5.4.1 Modeling and Data set

Inflow

As mentioned above the inflow level is set to $H\_in = 1.50 \cdot 2.50\ [m] = 3.75\ [m]$

Outflow

The outflow flap width covers the full channel width, $B\_K := B\_o$.

The overfall function Q_o (ΔdH_o) with $dH\_o = H\_o - Y\_K$, where Y_K is the position of the tip of the flap, is

$$\begin{aligned} \text{if} \quad dH\_o &\leq 0\,, \quad \text{then} \quad Q\_o = 0.0 \\ \text{if} \quad dH\_o &> 0\,, \quad \text{then} \quad Q\_o = B\_o \cdot dH\_o \cdot \sqrt{g \cdot dH\_o} \;=\; gQ\_K \cdot dH\_o^{1.5} \\ \text{with} \quad gQ\_K &:= B\_o \cdot g^{0.5} \end{aligned}$$

and where finally H_o will be replaced by D_o, and Y_o is measured from the bottom position S_o up.

The initial tip flap position will be chosen for the simulations such that the initial outflow is about $Q\_o\_0 = 5\ m^3/s$.

$$dH\_o = \left[\frac{Q\_o\_0}{\mu\ B\_o \cdot \sqrt{2\ g}}\right]^{2/3} \quad \text{with} \quad \mu = (2/3)$$

The interest here is on the quasi stationary response from the 'Basic Result' above. Therefore the flap shall move slowly and with no discontinuities on its slew rate.

This is implemented as in Fig.5.4 by a servo loop with a proportional controller. (Note that ON-OFF-valves are used in most applications).
To avoid sloshing transients the flap position setpoint is ramped down at a slew saturation of 30 s per 1 m that is 120 s for 4 m from fully up/closed to fully down/open. This is slow enough compared to the overall filling time $T\_f = L\_tot\ /\ U_r = 10s$. And the position control loop inserts an additional first order filter with time constant 10 s.

5.4.2 Implementation of Case 1

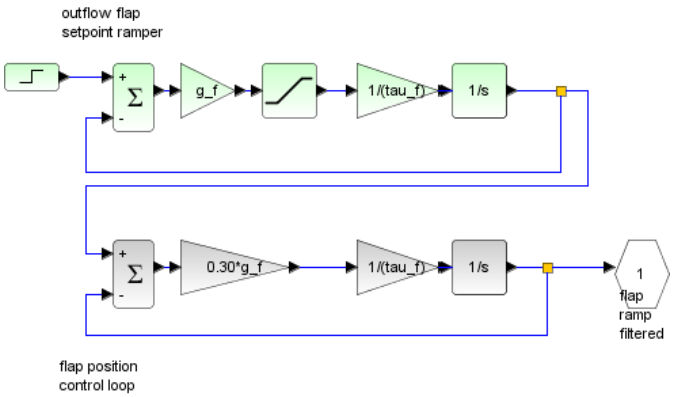


Figure 5.4: Outflow flap position control loop

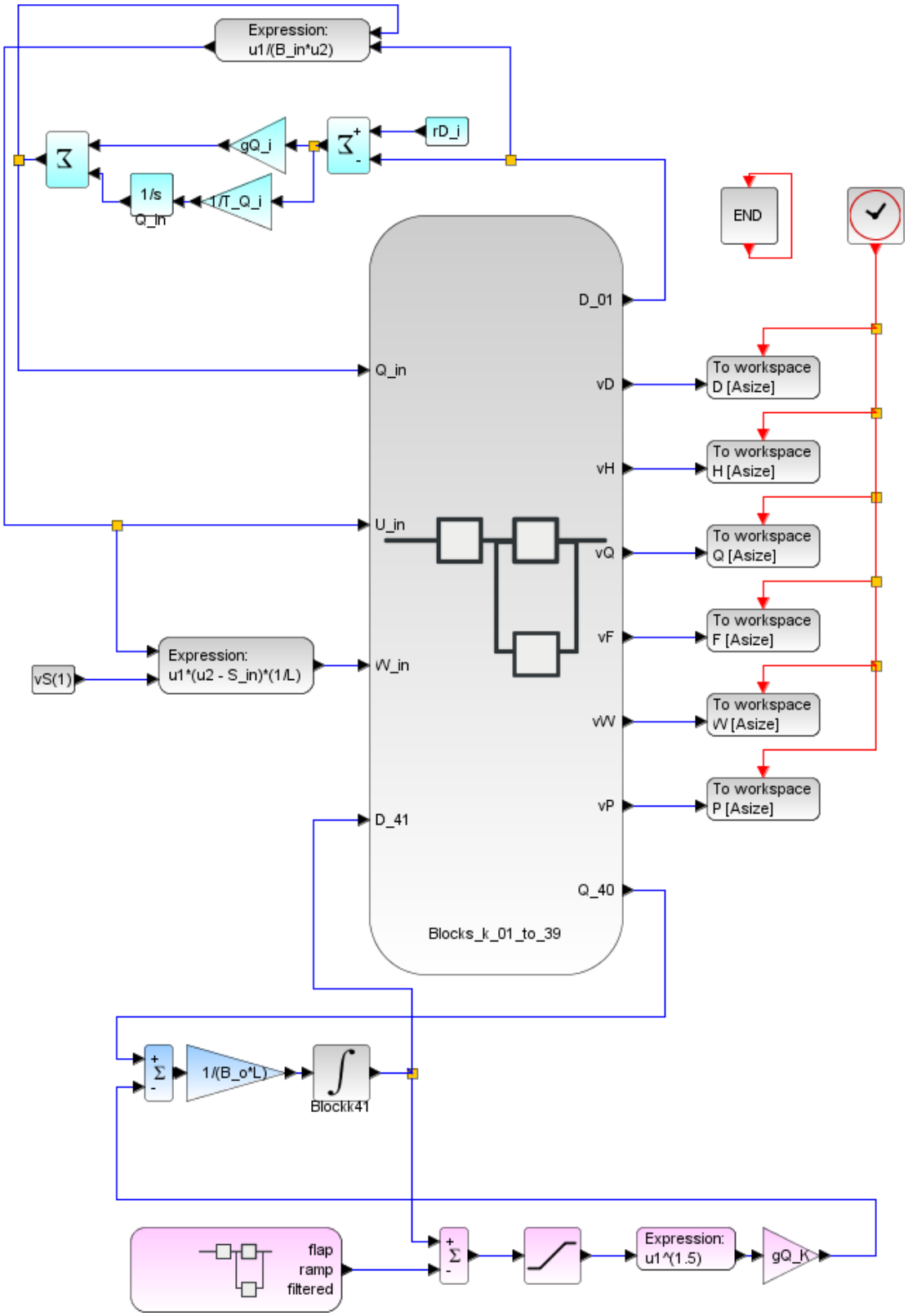


Figure 5.5: Main diagram for Case 1: Spillway with Inflow level control and outflow flap position control

```
// s_c5_01_b_context.sce
// Glf  2015_08_11
// outflow flap lowering
// Inflow with PI-LevelContr

g = 10.; Q_0 = 50.0; H_0 = 2.5; kap = 1.0;

lambda0 = 0.40;
//******************
// lambda = (1.25)*lambda0;
   lambda = (1.00)*lambda0;
// lambda = (0.80)*lambda0;
// lambda = (0.64)*lambda0;
//******************
L = lambda*H_0;  S_0 = 0.0*(-1.0);
D_0 = H_0 - S_0; B_0 = 10.; U_0 = 2.0;
U_F = sqrt(g*D_0);

N= 20;      // no of Vol+Momentum-Segments

// channel inlet geometry
vb_1 = ones(1,((2*N)+1)); vb_2 = vb_1;
vs_1 = vb_1; vs_2 = vs_1;

if lambda == (1.25)*lambda0 then
ik_E = 3; ik_R = 6;  delta_b = 0.144;
elseif lambda == 1.0*lambda0 then
ik_E = 4; ik_R = 8;  delta_b = 0.122;
elseif lambda == 0.80*lambda0 then
ik_E = 6; ik_R = 10; delta_b = 0.100;
else
ik_E = 8; ik_R = 12; delta_b = 0.089;
end
//**************************************

ik_T = ik_E + ik_R;
vib = ones(1,ik_T);
vd_b = ik_R*ones(1,(ik_T));
for ii = 1:ik_R
 vd_b(ii) =  ii;
end
vd_b = delta_b*vd_b/ik_R;
vincr_b  = vib + vd_b;

for ij = 1:(ik_T)
    vib(ij+1) = vib(ij)*vincr_b(ij);
end

vb_0 = vib(ik_T);
vb = vb_0*ones(1,ik_T);

for ijk = 1:ik_T
    nijk = ik_T+1 - ijk;
    nvib(nijk) = vib(ijk);
end

for k8 = 1:ik_T
    vb_2(k8) = nvib(k8);
    vs_2(k8) = nvib(k8);
end

vb_2 = vb_2 - vb_1;
vB = B_0*(lambda*H_0*vb_2 + vb_1);

vs_2 = vs_2 - vs_1;
vS = -(lambda*H_0*vs_2);

// setting initial conditions
//**************************************
 D_min = +0.001; D_max = 40*D_0;
 Q_min = +0.001; Q_max = 40*Q_0;

h0 = 1.52;  vh0 = ones(1,(2*N+1));
vH0 = h0*H_0*vh0; vD0 = vH0 - vS;

q0  = 0.060; vq0 = ones(1,(2*N+1));
vQ0 = q0*Q_0*vq0;

vDdot0 = 0*vh0; ome = 100;
P_max = 100.0; P_min = 0.0;

// Inflow-(_i) boundary cond.
    B_in = vB(1) + (vB(1) - vB(2))*(1+delta_b);
    S_in = vS(1) + (vS(1) - vS(2))*(1+delta_b);
    D_in = vH0(1) - S_in;    Q_in = vQ0(1);

// GMS-coefficient
  k_s = 100.;

// Friction-inclination of bottom
vD_E = H_0*vh0; vQ_E = U_F*B_0*H_0*vq0;

vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vRtilda(kk) = ((vB(kk)*vD_E(kk))/ ...
                 (vB(kk)+2*vD_E(kk)))^(2/3);
  vI_E(kk) = (vQ_E(kk)/(vB(kk)* ...
                vD_E(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI_E(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS(k4)  + vSf(k4);
    vH0(k4) = vH0(k4) + vSf(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow-(_o) boundary cond
    B_o = vB(kk+1);
    S_o = vS(kk+1);
    D_o = vD0(kk+1); H_o = vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow PI - level controller
  gQ_i = 20.; T_Q_i = 0.01;  rD_i = vD0(1) ;

// outflow by flap
  dH_o = (Q_o/(((2*g)^0.5)*(2/3)*B_o))^(2/3);
  t_f_1 = 75.;  f_1_0 = D_o - dH_o; f_1_1 = S_o;
  g_f = 10.0; u_up_f = +1.0; u_dn_f = -1.0;
  tau_f = 30.0; K_f_0 = f_1_0;
  gQ_K = B_o*((g)^(0.5));
  delQ_o = ((dH_o^(1.5))*gQ_K); // for checking

T_fin = 300.;

// Datatransfer to Plots
CC = 21;           // no of channels 20+1 for time
CN = 3000;          // no of clockticks to Tfin
delT = T_fin/CN;   // readout-interval clock ticks
Asize = 1.01*CC*CN; // size of data arrays
```

```
// s_c5_01_b_crunplot
//  Glf  2015_04_27

stacksize('max'); exec('s_c5_01_b_context.sce',-1);
importXcosDiagram('s_c5_01_b.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:11,  clf(kfig);  end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,...
       rect=[0.,0.0,300,150.]);
xtitle("Q_2 to Q_40");
xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,...
       rect=[0.,2.0,300,5.5]);
xtitle("D_1 to D_39");
xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,...
       rect=[0.,2.0,300,4.0]);
xtitle("H_1 to H_39");
xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,...
       rect=[0.,0.0,300,1.2]);
xtitle("F_2 to F_40");
xgrid(1);

f5 = scf(5);
plot2d(W.time,W.values,vcolor,...
       rect=[0.,-1.0,300,+1.0]);
xtitle("W_1 to W_39");
xgrid(1);

f6 = scf(6);
plot2d(P.time,(P.values-g*D.values),vcolor,...
       rect=[0.,-5.0,300,5.0]);
xtitle("deltaP*_1 to  deltaP*_39");
xgrid(1);

//*****************************************

eX = H_0*lambda*(1:1:N); eXend = 22;
for k6= 1:1:N,
  vSr(k6) = vS(2*k6-1);
end
```

```
vcolor=[2,5];

f7 = scf(7);
clf();
yD =D.values;
iD = yD(700,:); oD = yD(2900,:);
iH = iD + vSr'; oH = oD + vSr';
plot2d(eX',[iH',oH'],vcolor,rect=[0.,2.0,eXend,+4.0]);
xtitle("Longit.Profile H at 70 s (bl) and 290 s (rd)");
xgrid(1);
//
f8 = scf(8);
clf();
yF =F.values;
iF = yF(700,:); oF = yF(2900,:);
plot2d(eX',[iF',oF'],vcolor,rect=[0.,0.0,eXend,+1.2]);
xtitle("Longit.Profile F at 70 s (bl) and 290 s (rd)");
xgrid(1);
//
f9 = scf(9);
clf();
yW =W.values;
iW = yW(700,:); oW = yW(2900,:);
plot2d(eX',[iW',oW'],vcolor,rect=[0.,-1.0,eXend,+1.0]);
xtitle("Longit.Profile  W at 70 s (bl) and 290 s (rd)");
xgrid(1);
//
f10 = scf(10);
clf();
yP =P.values;
iP = yP(700,:)-g*yD(700,:); oP = yP(2900,:)-g*yD(2900,:);
plot2d(eX',[iP',oP'],vcolor,rect=[0.,-5.0,eXend,+5.0]);
xtitle("Longit.Profile (P*-gD) at 70s (bl) and 290s (rd)");
xgrid(1);
//
f11 = scf(11);
clf();
eY = 0.5*H_0*lambda*(1:1:(2*N)+1);
plot2d(eY',[0.1*vB',vS'],vcolor,rect=[0.,-3.0,eXend,+3.0]);
xtitle("Longit.Channel Geometry,0.1*B(x) (bl), S(x) (rd)");
xgrid(1);

// for assembling logitudinal contours
if lambda == (1.25)*lambda0 then
   ac_125 = [eY',[0.1*vB',vS']];
   aH_125 = [eX',oH']; aF_125 = [eX',oF'];
elseif lambda == 1.0*lambda0 then
   ac_100 = [eY',[0.1*vB',vS']];
   aH_100 = [eX',oH']; aF_100 = [eX',oF'];
elseif  lambda == 0.80*lambda0 then
   ac_080 = [eY',[0.1*vB',vS']];
   aH_080 = [eX',oH']; aF_080 = [eX',oF'];
else
   ac_064 = [eY',[0.1*vB',vS']];
   aH_064 = [eX',oH']; aF_064 = [eX',oF'];
end
```

```
// s_c5_01_b_contassemb
// Glf 2015_08_11

  for kfig = 12:1:16,  clf(kfig); end
 eXend = 30;

vcolor3 = [ 2, 5, 1, 13];
vcolor4 = [ 2, 5, 1, 13, 2, 5, 1, 13];

f12 = scf(12);
plot2d([ac_125(:,1),ac_100(:,1),ac_080(:,1),ac_064(:,1),...
 ac_125(:,1),ac_100(:,1),ac_080(:,1),ac_064(:,1)],...
[ac_125(:,2),ac_100(:,2),ac_080(:,2),ac_064(:,2) ...
 ac_125(:,3),ac_100(:,3),ac_080(:,3),ac_064(:,3)],...
vcolor4,rect=[0.,-2.0,eXend,+3.0]);
xtitle("geometry contour 0.1*B and S");
legend("L = 1.25 m","1.00 m","0.80 m", "0.64 m",...
       "L = 1.25 m","1.00 m","0.80 m", "0.64 m")
```

```
xgrid(1);

f13 = scf(13);
plot2d([aH_125(:,1),aH_100(:,1),aH_080(:,1),aH_064(:,1)],...
[aH_125(:,2),aH_100(:,2),aH_080(:,2),aH_064(:,2)],...
vcolor3,rect=[0.,2.0,eXend,+4.0]);
xtitle(" contours H");
legend("L = 1.25 m","1.00 m","0.80 m", "0.64 m")
xgrid(1);

f14 = scf(14);
plot2d([aF_125(:,1),aF_100(:,1),aF_080(:,1),aF_064(:,1)],...
[aF_125(:,2),aF_100(:,2),aF_080(:,2),aF_064(:,2)],...
vcolor3,rect=[0.,0.0,eXend,+1.2]);
xtitle(" contours F");
legend("L = 1.25 m","1.00 m","0.80 m", "0.64 m")
xgrid(1);
```

5.4.3 Simulation overview for Case 1

Three different situations will be investigated by applying their respective parameter sets.

1. The first situation is generated by the *nominal* parameter set: $\_E$ for end / outflow values

$$\begin{aligned} H\_E &= H\_0 = 2.50\ m \\ S\_E &= 0.0\ m \\ D\_E &= 2.50\ m \ \rightarrow \ U\_F = 5.0\ m/s \\ \lambda_0 &= 0.40 \quad \text{and} \quad \lambda = (1.00) \cdot \lambda_0 \\ \rightarrow L &= 1.0\ m \quad \text{given by} \quad L = \lambda \cdot H_0 \\ \rightarrow Q\_E &= 125\ m^3/s \end{aligned}$$

   Transients are given for $vQ(t),\ vD(t),\ vH(t),\ F(t),\ W(t),\ \Delta P^*(t)$. in the time window $t = 0...300\ s$ and longitudinal profiles over x for $vH(x),\ vF(x),\ vW(x),\ \Delta P^*(x)$ for two time instants, after stabilizing at low initial flow $t = 70\ s$, and after stabilizing at full flow $t = 290\ s$.

2. The second situation is for testing different compartment lengths L, to check whether the nominal value of L is a 'good enough' choice.
   All other variables are carried over from situation 1. Four values for L are entered:

$$L = 1.25,\ 1.00,\ 0.80,\ 0.64\ m$$

   This covers a spread of factor ≈ 2 and includes the nominal value. The effect of the value of L is shown by assembling the longitudinal profiles of $vH(x)$ and $vF(x)$ in one plot each.

3. The third situation (with the triplet `s_c5_01_c`) is a variation of the nominal depth D_E at the outflow end

$$D\_E = H\_0 = 3.60,\ 2.50,\ 1.60\ m$$

   again covering a spread of ≈ 2 and including the nominal value.
   This generates

$$\begin{aligned} U\_F\_E &= 6.0, \quad 5.0, \quad 4.0\ m/s \\ Q\_E &= 216, \quad 125, \quad 64\ m^3/s \end{aligned}$$

   The compartment length is set to the same relation $L = 1.00 \cdot \lambda_0 \cdot H\_0$ as in the nominal case. In other words L varies proportionally to H_0. This keeps the geometry of the compartments affine in the lateral plane.
   Also the inclination of the bottom is adjusted to the value pairs of $U\_F\_E,\ D\_E$ with constant GMS coefficient $k\_s = 100$.
   The longitudinal profiles of $vH(x)$ and $vF(x)$ are plotted both in absolute values and on scaled values:

$$\begin{aligned} \text{on the x-axis} \quad x_{scaled} &:= \frac{U\_F_{nom}}{U\_F_{actual}} \cdot x_{actual} \\ &= \sqrt{\frac{D\_E_{nom}}{D\_E_{actual}}} \cdot x_{actual};\ \text{with factors } \left[\sqrt{\frac{2.50}{3.60}},\ \sqrt{\frac{2.50}{2.50}},\ \sqrt{\frac{2.50}{1.60}}\right] \\ \text{on the y-axis for } H \quad H_{scaled} &:= \frac{D\_E_{nom}}{D\_E_{actual}} \cdot H_{actual};\ \text{with factors } \left[\frac{2.50}{3.60},\ \frac{2.50}{2.50},\ \frac{2.50}{1.60}\right] \\ \text{on the y-axis for } F \quad F_{scaled} &:= F_{actual};\ \text{no scaling required} \end{aligned}$$

5.4.4 Case 1: Simulation results for the reference parameter set

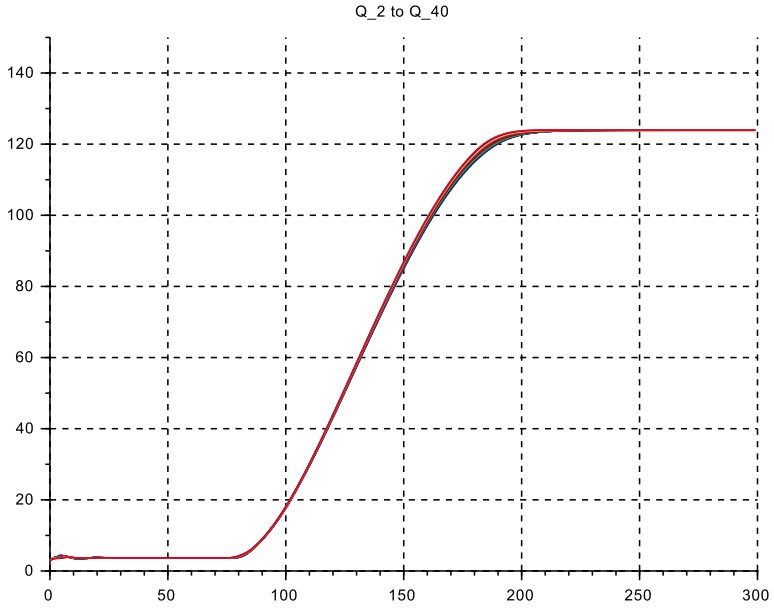


Figure 5.6: C 1: Transient Q for $L = 1$ $[m]$

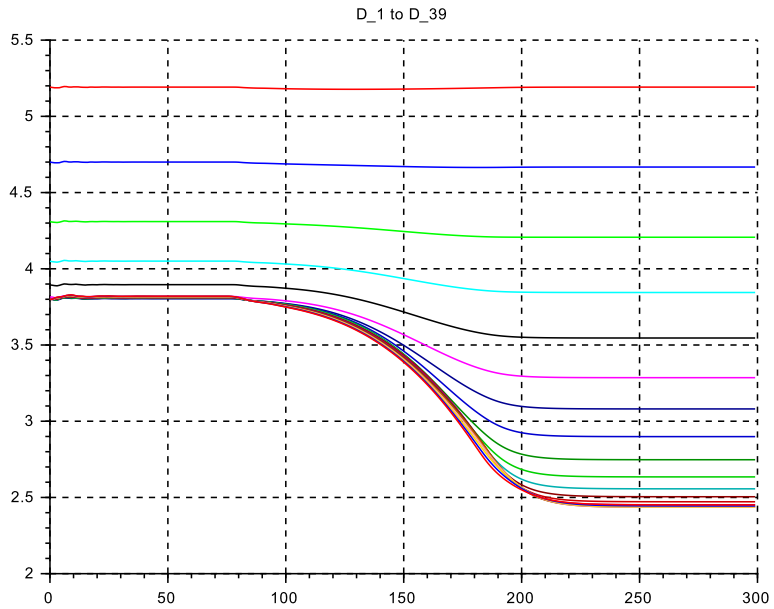


Figure 5.7: C 1: Transient D for $L = 1$ $[m]$

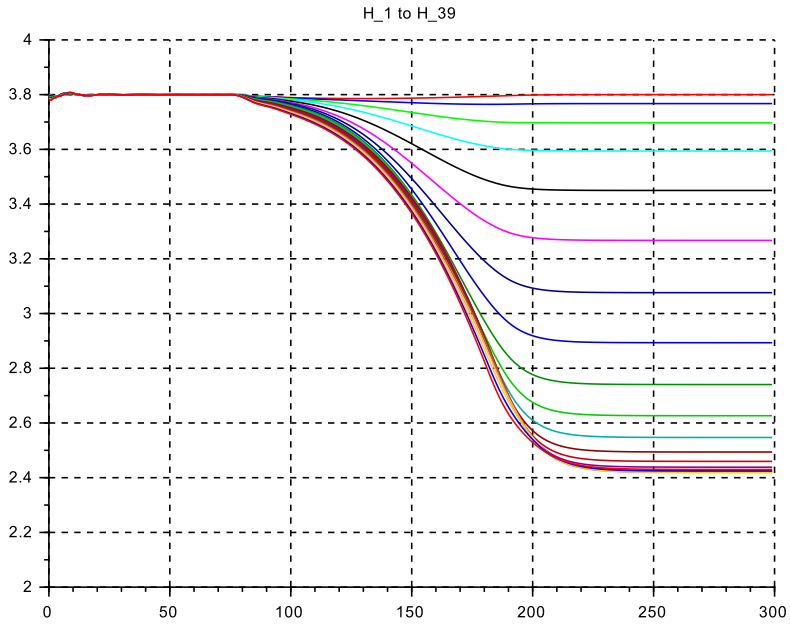


Figure 5.8: C 1: Transient H for $L = 1$ $[m]$

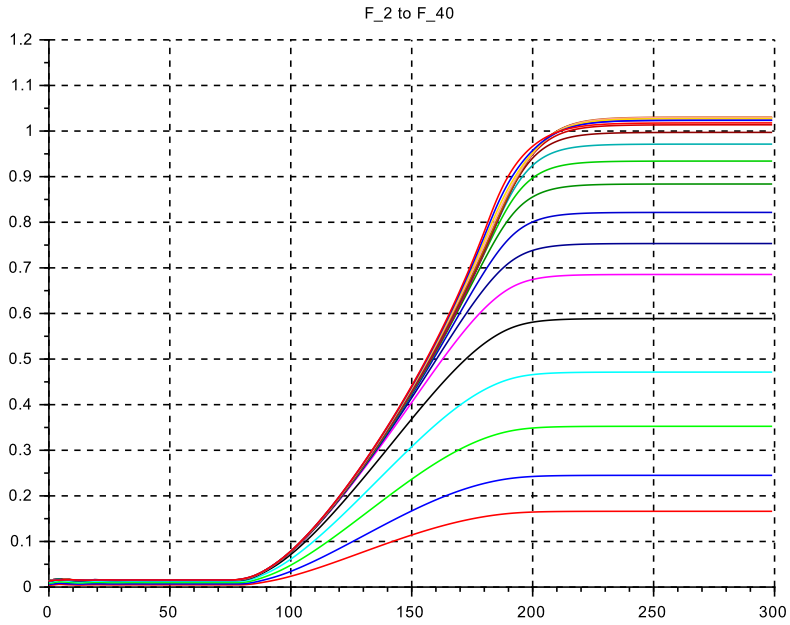


Figure 5.9: C 1: Transient F for $L = 1$ $[m]$

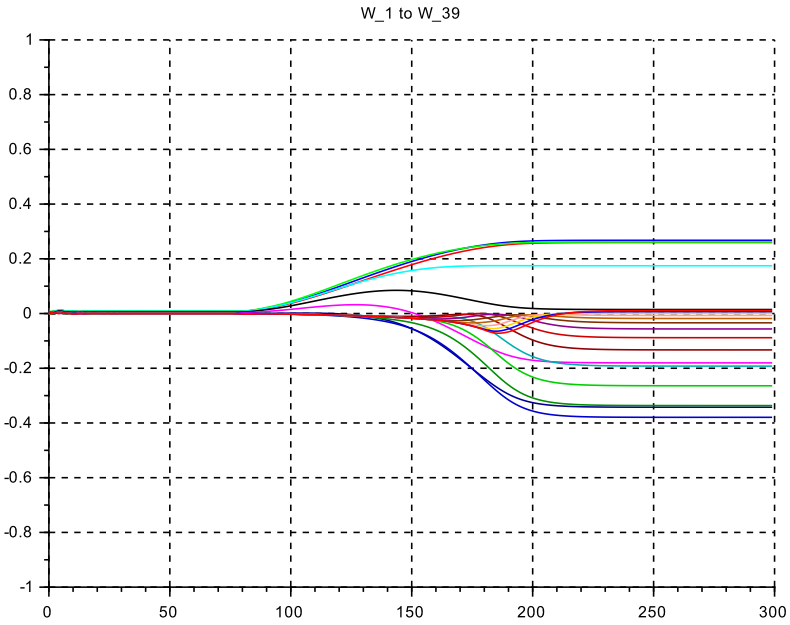


Figure 5.10: C 1: Transient W for $L = 1$ $[m]$

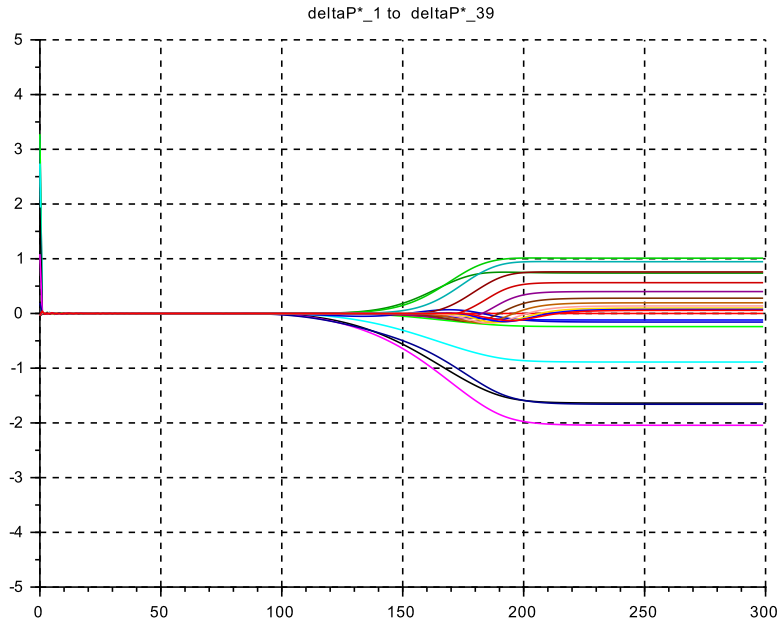


Figure 5.11: C 1: Transient $P^* - gD$ for $L = 1$ $[m]$

Longitudinal profiles at time 290 s

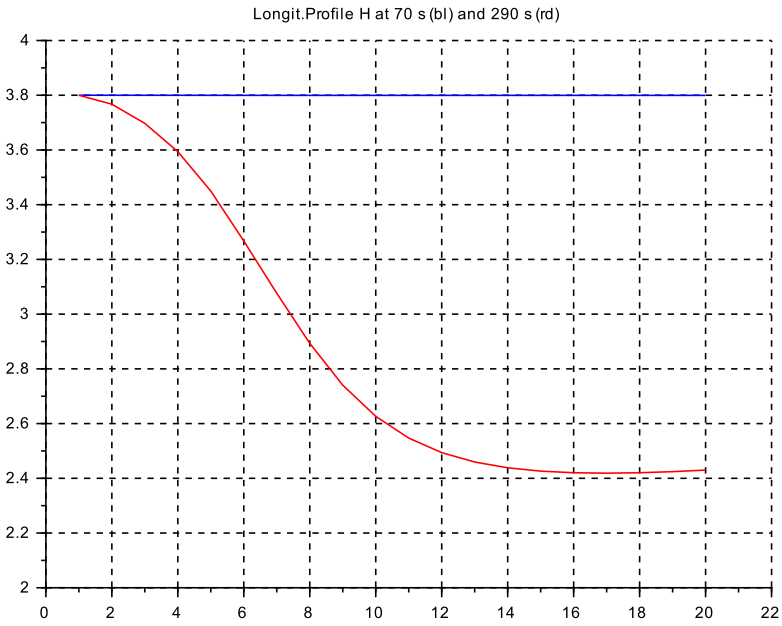


Figure 5.12: C 1: Profile for H for $L = 1$ $[m]$

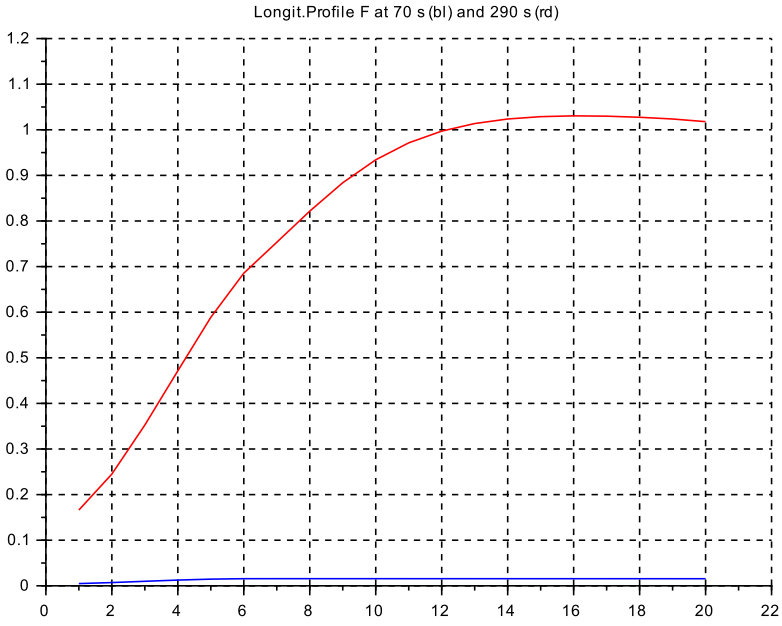


Figure 5.13: C 1: Profile for F for $L = 1$ $[m]$

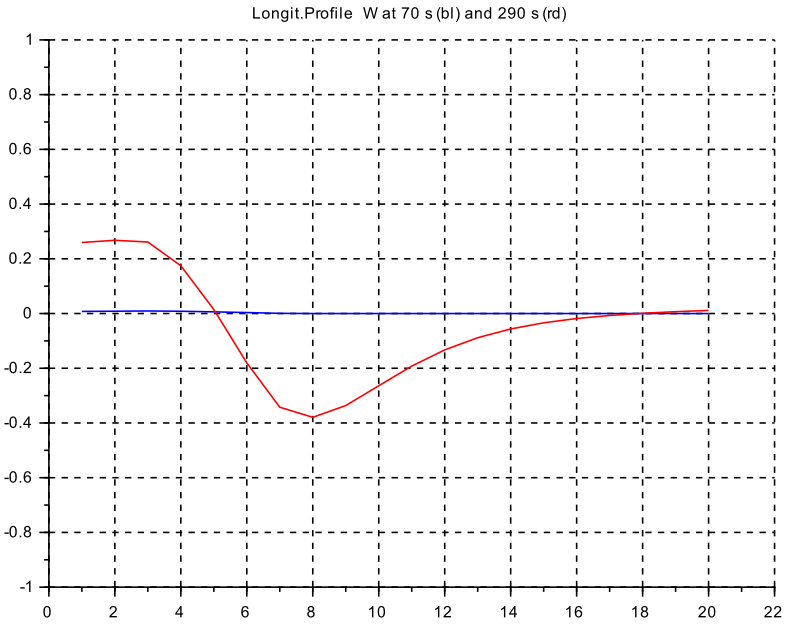


Figure 5.14: C 1: Profile for W for $L = 1$ $[m]$

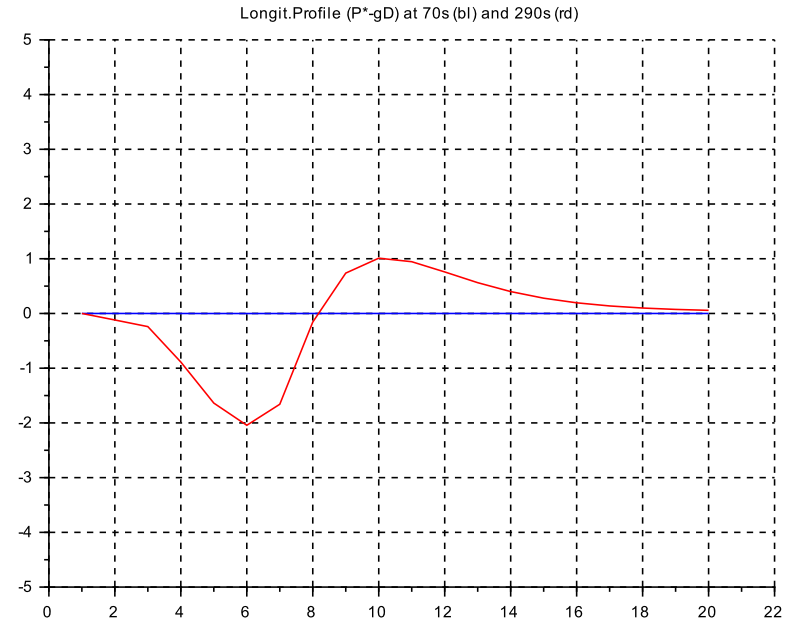


Figure 5.15: C 1: Profile for $P^* - gD$ for $L = 1$ $[m]$

5.4.5 Testing for different compartment lengths $L = 1.25,\ 1.00,\ 0.80,\ 0.64\ m$: Assembly of longitudinal profiles of H and F at time $290\ s$

```
// s_c5_01_b_contassemb
// Glf 2015_08_11

  for kfig = 12:1:16,  clf(kfig); end
 eXend = 30;

vcolor3 = [ 2, 5, 1, 13];
vcolor4 = [ 2, 5, 1, 13, 2, 5, 1, 13];

f12 = scf(12);
plot2d([ac_125(:,1),ac_100(:,1),ac_080(:,1),ac_064(:,1),...
 ac_125(:,1),ac_100(:,1),ac_080(:,1),ac_064(:,1)],...
[ac_125(:,2),ac_100(:,2),ac_080(:,2),ac_064(:,2) ...
 ac_125(:,3),ac_100(:,3),ac_080(:,3),ac_064(:,3)],...
vcolor4,rect=[0.,-2.0,eXend,+3.0]);
xtitle("geometry contour 0.1*B and S");
legend("L = 1.25 m","1.00 m","0.80 m", "0.64 m",...
       "L = 1.25 m","1.00 m","0.80 m", "0.64 m")
xgrid(1);

f13 = scf(13);
plot2d([aH_125(:,1),aH_100(:,1),aH_080(:,1),aH_064(:,1)],...
[aH_125(:,2),aH_100(:,2),aH_080(:,2),aH_064(:,2)],...
vcolor3,rect=[0.,2.0,eXend,+4.0]);
xtitle(" contours H");
legend("L = 1.25 m","1.00 m","0.80 m", "0.64 m")
xgrid(1);

f14 = scf(14);
plot2d([aF_125(:,1),aF_100(:,1),aF_080(:,1),aF_064(:,1)],...
[aF_125(:,2),aF_100(:,2),aF_080(:,2),aF_064(:,2)],...
vcolor3,rect=[0.,0.0,eXend,+1.2]);
xtitle(" contours F");
legend("L = 1.25 m","1.00 m","0.80 m", "0.64 m")
xgrid(1);
```

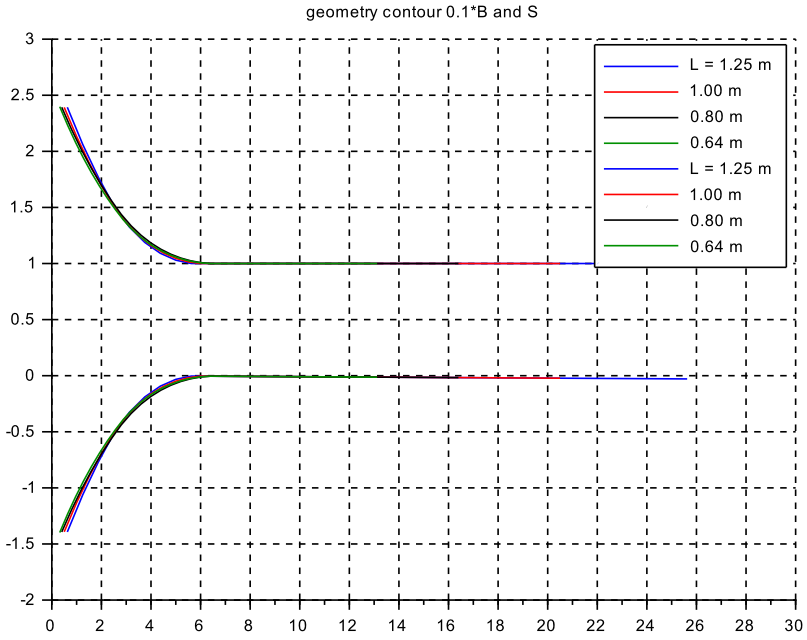


Figure 5.16: C 1: Assembly of Geometry Profiles for $0.1 \cdot B$ and S

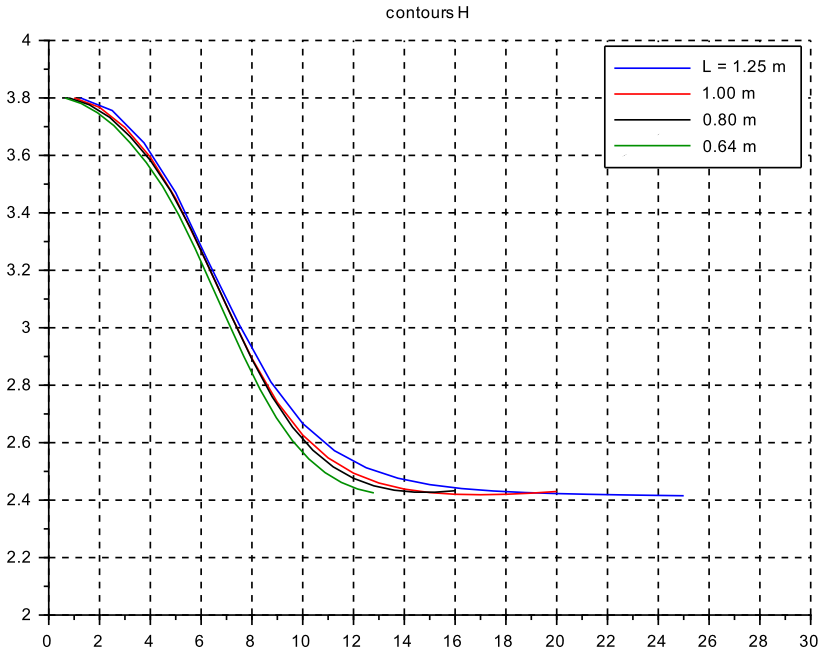


Figure 5.17: C 1: Assembly of Profiles for H

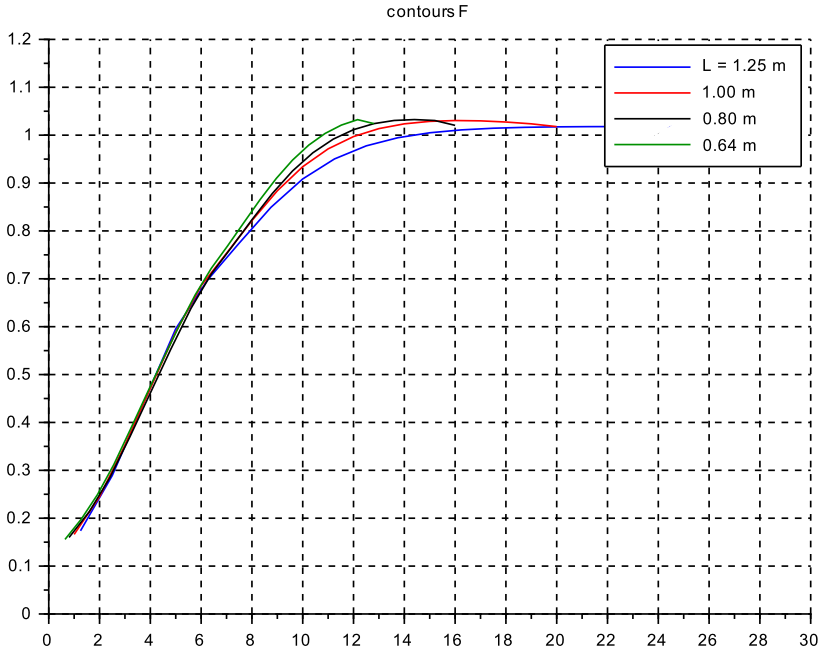


Figure 5.18: C 1: Assembly of Profiles for F

5.4.6 Variation of outflow level $H_E = 3.60,\ 2.50,\ 1.60\ m$ and corresponding flow $Q = 216,\ 125,\ 64\ m^3/s$

The main diagram `s_c5_01_c.zcos` is renamed from `s_c5_01_b.zcos` without changes.

```
// s_c5_01_c_context.sce
// Glf  2015_08_11
// outflow flap lowering
// Inflow with PI-LevelContr

g = 10.; Q_0 = 50.0;

//H_0 = 3.6;
//H_0 = 2.5;
  H_0 = 1.6;
//******************

lambda0 = 0.40; kap = 1.0;
//******************
//lambda = (1.25)*lambda0;
  lambda = (1.00)*lambda0;
//lambda = (0.80)*lambda0;
//lambda = (0.64)*lambda0;
//******************
L = lambda*H_0;
S_0 = 0.0*(-1.0);
D_0 = H_0 - S_0;
B_0 = 10.;
U_0 = 2.0;
U_F = sqrt(g*H_0);

N= 20;      // no of Vol+Momentum-Segments


// channel inlet geometry
//*********************************************

vb_1 = ones(1,((2*N)+1)); vb_2 = vb_1;
vs_1 = vb_1; vs_2 = vs_1;

if H_0 == 3.60 then
ik_E = 3; ik_R = 6;  delta_b = 0.144;
elseif H_0 == 2.50 then
ik_E = 4; ik_R = 8;  delta_b = 0.122;
else
ik_E = 6; ik_R = 10; delta_b = 0.100;
end
//*************************************

ik_T = ik_E + ik_R;
vib = ones(1,ik_T);
vd_b = ik_R*ones(1,(ik_T));
for ii = 1:ik_R,
 vd_b(ii) =  ii ;
end
vd_b = delta_b*vd_b/ik_R;
vincr_b  = vib + vd_b;

for ij = 1:(ik_T),
    vib(ij+1) = vib(ij)*vincr_b(ij);
end

vb_0 = vib(ik_T);
vb = vb_0*ones(1,ik_T);

for ijk = 1:ik_T,
    nijk = ik_T+1 - ijk;
    nvib(nijk) = vib(ijk);
end

for k8 = 1:ik_T,
    vb_2(k8) = nvib(k8);
    vs_2(k8) = nvib(k8);
end

vb_2 = vb_2 - vb_1;
vB = B_0*(lambda*H_0*vb_2 + vb_1);

vs_2 = vs_2 - vs_1;
vS = -(lambda*H_0*vs_2);

// setting initial conditions
//*****************************************
 D_min = +0.001; D_max = 40*D_0;
 Q_min = +0.001; Q_max = 40*Q_0;

h0 = 1.52;
vh0 = ones(1,(2*N+1));
vH0 = h0*H_0*vh0;
vD0 = vH0 - vS;

q0  = 0.060;
vq0 = ones(1,(2*N+1));
vQ0 = q0*Q_0*vq0;

vDdot0 = 0*vh0;  ome = 100;
P_max = 100.0; P_min = 0.0;

// Inflow-(_i) boundary cond.
    B_in = vB(1) + (vB(1) - vB(2))*(1+delta_b);
    S_in = vS(1) + (vS(1) - vS(2))*(1+delta_b);
    D_in = vH0(1) - S_in;
    Q_in = vQ0(1);

// GMS-coefficient
  k_s = 100.;

// Friction-inclination of bottom
vD_E = H_0*vh0;  vQ_E = U_F*B_0*H_0*vq0;
vdelSf = zeros(1,(2*N+1)); vSf = zeros(1,(2*N+1));
vRtilda = vdelSf; vI_E = vSf;
for kk=2:2:(2*N),
  vRtilda(kk) = ((vB(kk)*vD_E(kk))/ ...
                 (vB(kk)+2*vD_E(kk)))^(2/3);
  vI_E(kk) = (vQ_E(kk)/(vB(kk)* ...
                vD_E(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI_E(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS(k4)  + vSf(k4);
    vH0(k4) = vH0(k4) + vSf(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow-(_o) boundary cond
    B_o = vB(kk+1);  S_o = vS(kk+1);
    D_o = vD0(kk+1); H_o = vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow PI - level controller
  gQ_i = 30.; T_Q_i = 0.01;  rD_i = vD0(1);

// outflow by flap
  dH_o = (Q_o/(((2*g)^0.5)*(2/3)*B_o))^(2/3);
  t_f_1 = 45.;  f_1_0 = D_o - dH_o; f_1_1 = S_o;
  g_f = 10.0; u_up_f = +1.0; u_dn_f = -1.0;
  tau_f = 30.0; K_f_0 = f_1_0;
  gQ_K = B_o*((g)^(0.5));
  delQ_o = ((dH_o^(1.5))*gQ_K); // for checking

T_fin = 300.;

// Datatransfer to Plots
CC = 21;           // no of channels 20+1 for time
CN = 300;          // no of clockticks to Tfin
delT = T_fin/CN;   // readout-interval clock ticks
Asize = 1.01*CC*CN; // size of data arrays
```

```
// s_c5_01_c_contassemb
// Glf 2015_08_11

  for kfig = 12:1:14,  clf(kfig); end
 eXend = 30;

vcolor3 = [ 2, 5, 1];
vcolor4 = [ 2, 5, 1, 2, 5, 1];

f12 = scf(12);
plot2d([ac_360(:,1),ac_250(:,1),ac_160(:,1), ...
        ac_360(:,1),ac_250(:,1),ac_160(:,1)],...
       [ac_360(:,2),ac_250(:,2),ac_160(:,2), ...
        ac_360(:,3),ac_250(:,3),ac_160(:,3)],...
        vcolor4,rect=[0.,-2.0,eXend,+3.0]);
xtitle("geometry contour 0.1*B and S");
legend("H_0 = 3.60 m","2.50 m","1.60 m", ...
       "H_0 = 3.60 m","2.50 m","1.60 m")
xgrid(1);
```

```
f13 = scf(13);
plot2d([aH_360(:,1),aH_250(:,1),aH_160(:,1)],...
       [aH_360(:,2),aH_250(:,2),aH_160(:,2)],...
        vcolor3,rect=[0.,1.0,eXend,+6.0]);
xtitle(" contours H");
legend("H_0 = 3.60 m","2.50 m","1.60 m")
xgrid(1);
legend("H_0 = 3.60 m","2.50 m","1.60 m")
xgrid(1);

f16 = scf(16);
plot2d([aF_360(:,1),aF_250(:,1),aF_160(:,1)],...
       [aF_360(:,2),aF_250(:,2),aF_160(:,2)],...
        vcolor3,rect=[0.,0.0,eXend,+1.2]);
xtitle(" contours F");
legend("H_0 = 3.60 m","2.50 m","1.60 m")
xgrid(1);
```

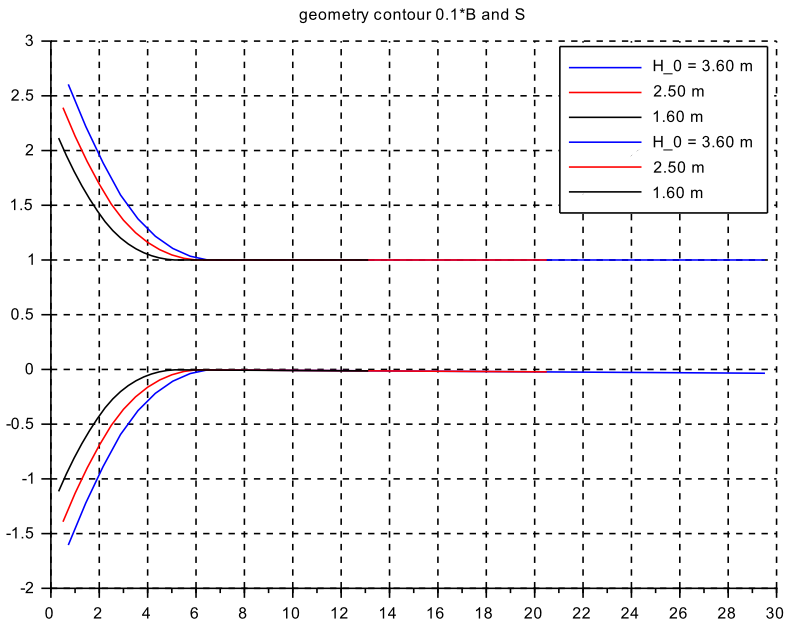


Figure 5.19: C 1: Assembly of Geometry Profiles for $0.1 \cdot B$ and S

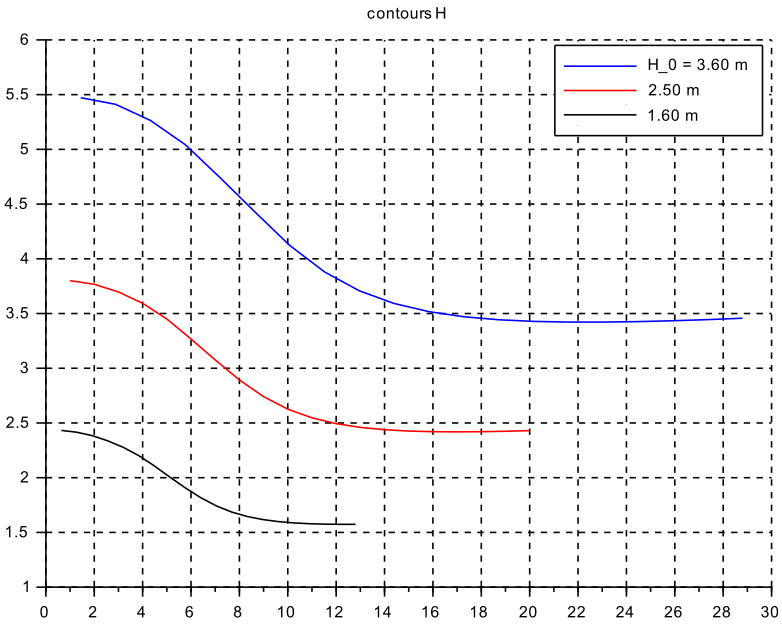


Figure 5.20: C 1: Assembly of Profiles for H

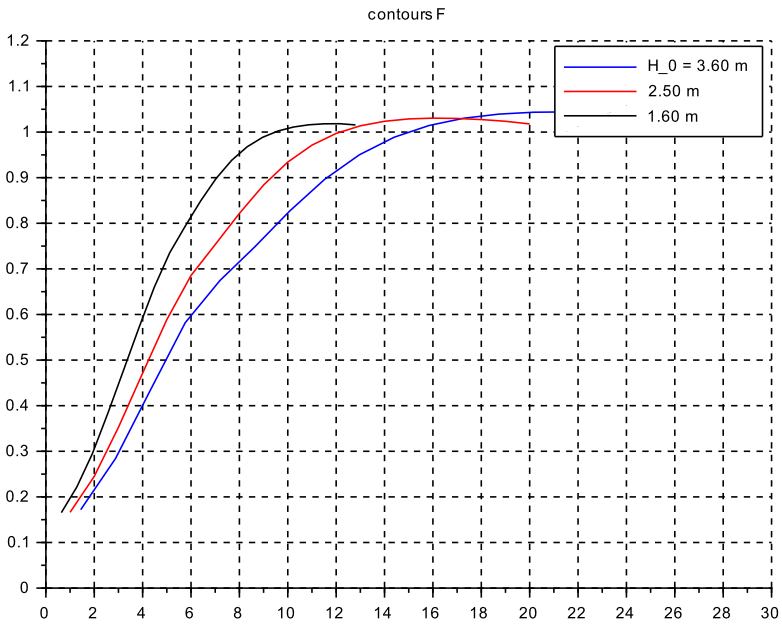


Figure 5.21: C 1: Assembly of Profiles for F

Assemblies with scaling factors applied

```
// s_c5_01_c_contassemb_scaled
// Glf 2015_08_12

  for kfig = 15:1:16,  clf(kfig); end
 eXend = 30;

vcolor3 = [ 2, 5, 1];

f15 = scf(15);
plot2d([aH_360(:,1)*sqrt(2.5/3.6),aH_250(:,1), ...
        aH_160(:,1)*sqrt(2.5/1.6)],...
       [aH_360(:,2)*(2.5/3.6),aH_250(:,2), ...
        aH_160(:,2)*(2.5/1.6)],...
        vcolor3,rect=[0.,2.0,eXend,+4.0]);
xtitle(" contours H, scaled");
legend("H_0 = 3.60 m","2.50 m","1.60 m")
xgrid(1);


f16 = scf(16);
plot2d([aF_360(:,1)*sqrt(2.5/3.6),aF_250(:,1), ...
        aF_160(:,1)*sqrt(2.5/1.6)],...
       [aF_360(:,2),aF_250(:,2),aF_160(:,2)],...
        vcolor3,rect=[0.,0.0,eXend,+1.2]);
xtitle(" contours F, scaled");
legend("H_0 = 3.60 m","2.50 m","1.60 m")
xgrid(1);
```

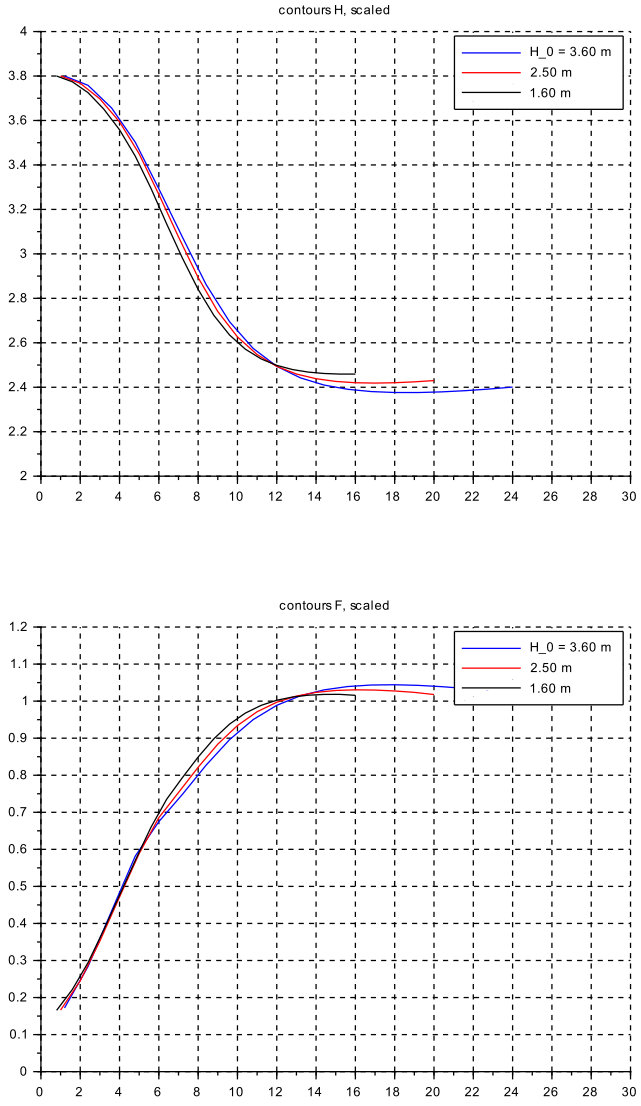


Figure 5.22: C 1: Assembly of scaled Profiles for H (top) and F (bottom)

5.4.7 Discussion for Case 1

Transients for the nominal parameter set

B, S The channel geometry is extended backwards to $B\_in = 26.84\ m, S\_in = -1.684\ m$, using the generating rule (see `_context.sce`), but not plotted. Thus the inflow cross section is $B\_inD\_in \approx 148\ m^2$, that is ≈ 6 times the outflow cross section, and thus $U\_in \approx 0.85\ m/s$ and further $\Delta H\_in = H_{\text{inf}} - H\_in \approx 0.036\ m$.

Q The final flow turns out to be $\approx 125\ m^3/s$, as expected.
The initial flow is set to $\approx 5\ m^3/s$. However the bottom slope is set to full flow at the outflow location. This causes a small settling transient.

D, H The inflow level H_in is set to $1.52 \cdot H\_0 = 3.80m$ that is $+0.05\ m$. The bottom level at the outflow location is at $S\_E = -0.022\ m$. At full flow both H_o and D_o are $\approx 0.05\ m$ below $2.50\ m$.

F Initially F starts at ≈ 0.01 and rises to slightly above 1.0 at full flow.

W and

ΔP^* Both are initially very small and decay back to near zero for steady state full flow.
Values in between are difficult to distinguish. - This has been the main motivation to implement the longitudinal plots.

Note that the movement of the outflow flap is slow enough to avoid sloshing.

Longitudinal profiles for the nominal parameter set

H There is a smooth descent along the x-axis from $H\_in = 3.8\ m$ to H_end,

F and a smooth rise of F from the inflow value ≈ 0.17 to its outflow value ≈ 1.017. Note the small overshoot at downstream position 16 m up to 1.025 The small bump at 6 m is from the end of the confusor part.

W At the inflow W is positive ($\approx +0.25\ m/s$) due to the rising bottom profile S. W starts to descend at 3 m, and crosses zero at $\approx 5\ m$. This is due to the superposition of vertical velocities at the bottom and the surface. At position 8 m W attains its minimum ($\approx -0.40\ m/s$) and then decays slowly to zero at the outflow location.

ΔP^* starts from near zero at the inflow location, drops to -2.0 units at location 6 m (the end of the confusor zone), crosses zero at 8 m, rises to $+1.0$ units at location 10 m, and decays to zero at the outflow. - Note that the corresponding hydrostatic pressure would be around 30 units, that is the deviations of P^* are $-6\%, +3\%$. - Both W and its spacial derivative ΔP^* are 'smooth enough' for practical purposes. And this confirms that the model length $40 + 1$ compartments is adequate for this type of flow regimes in 'short channels'.

Note how informative the longitudinal profiles are. For this reason they will be shown and discussed exclusively for the next situations.

Longitudinal profiles for different compartment lengths L at nominal conditions

B, S This is shown to illustrate that the channel contours are the same for all four cases of L. The end of the confusor zone is at position 6 m. The plot starts at $vB(1),\ vS(1)$, not $B\_in,\ S\_in$.

H The traces for all four cases of L are close together from the inflow position up to location 8 m at $H \approx 2.9\ m$. Then they start to spread out on an interval of $\approx \pm 0.05\ m$, but end at the same level $H\_E \approx 2.43\ m$. The slowest decay is for $L = 1.25\ m$, and the fastest for $L = 0.64\ m$. For the shortest L the final state on H is clearly not yet attained: the model length of $40 + 1$ compartments would have to be increased ...

F mirrors the same form. There is an overshoot of $\approx +0.025$ in all cases of L. The end point overshoot F_E is nearly the same for all L, (at $\approx +0.017$).

To summarize $L = 1.0\ m = 0.40 \cdot H\_0$ seems to be a 'good enough' compromise between model length / complexity / computation runtime and results being close to experimental findings.

Longitudinal profiles for different levels $D\_E = H\_0$, absolute axes

For page layout reasons the $\cdots$`.sce` files are placed just after the three plots. The full outflow values are not shown in plots, but are given here: for $D\_E = [3.60, 1.60\ m]$ then $Q\_E \approx \left[220, 66\ m^3/s\right]$.

B, S Fig.5.19 shows the channel contours for the three subcases of D_E's and associated L's, again starting from $vB(1),\ vS(1)$. No effort has been made to produce the same contours.

H The profiles are as expected. For the subcase $D\_E = 3.60\ m$ the final $H\_E \approx 3.50\ m$. Part of the deviation is due to the higher friction slope of the bottom.

F Again the profiles are as expected. The maximum overshoot on F is now $\approx +0.035$ decaying to $\approx +0.020$ at the outflow location.

Longitudinal profiles for different levels $D\_E = H\_0$, scaled axes

The idea here is to check if there are possibly some similarity laws (?)
As mentioned above, tentatively, the x-axis is scaled by the basic Froude velocity U_F_E , and the z-axis for H is scaled by the outflow depth D_E. On the z-axis for F obviously no scaling needs to be applied.

H Up to position 10 the traces are bundled within $\approx \pm 0.05\ m$. Again there is a divergence when approaching the outflow position. There is a distinct undershoot down to -0.10 units, below the target at 2.5 units. This is more pronounced for the larger full flow for $D\_E = 3.60\ m$.

F This is mirrored for F. Traces stay close together up to position 6 m, then diverge to a band of ± 0.05. Then there is an overshoot of F to a maximum of $+0.035... + 0.016$ for the largest/smallest D_E subcases, and then decaying to $+0.020... + 0.015$.

So one may tentatively deduce from the experimental results that this scaling is indeed valid, and points to similarity properties. Of course this is a conjecture and no mathematical proof.

5.5 Case 2: Spillway with slowly rising forebay level $H\_in(t)$

5.5.1 Modeling and Data

The focus here shall be on the elements which are really new. Thus data shall be carried over from case 1 as far as feasible. This concerns the reference data for flow and depth, the channel geometry, the friction and the resulting bottom slope.

inflow boundary

The inflow level is not constant as in case 1, but moves up from a low initial value up to the value linked to full flow. The initial level is selected at $H = 0.80\ m$ such that a steady state flow $\overline{Q} \approx 12\ m^3/s$ results that is approx 10 % of full flow. The final value is set as in case 1 at $H = 3.80\ m$ which has lead to $\overline{Q} = 125\ m^3/s$ in case 1.
In order to conserve the same flow pattern over the run-up experiment, no lower initial flow level is attempted here, as this would lead to drying out of the channel and subsequent flooding of the channel which much higher Froude numbers.
The run-up is planned to take approx 100 s, which is much faster than in most practical applications.
The setpoint ramp is filtered by a first order lag to suppress any sloshing.
Finally the level controller from case 1 with its parameter settings is carried over for tracking the rising forebay level setpoint by providing the necessary inflow to it.

outflow boundary

The outflow boundary condition is generated here by the GMS law, using the bottom slope I from case 1. It has been determined for $D\_E = 2.5\ m$ and $Q\_E = 125\ m^3/s$, which implies the Froude condition at this operating point.
Next the case of much lower flow is investigated. As a first approximation, assume that the Froude condition is still valid. Then for $Q = 12\ m^3/s$ and $B = 10\ m$

$$\begin{aligned} Q &= B\sqrt{g}\cdot D^{3/2} \quad \rightarrow \quad D = \left[\frac{Q}{B\sqrt{g}}\right]^{2/3} \\ \text{and numerically} \quad D &= D\_o = 0.524\ m; \quad \text{and} \quad H\_in = 1.5 * D\_o = 0.790\ m \end{aligned}$$

Applying the GMS-rule

$$\begin{aligned} U &= k\_s R^{2/3} I^{1/2} \quad \text{with} \quad R = \frac{BD}{B+2D} = D\frac{1}{1+(2D/B)} \\ \text{with} \quad F &= \frac{U}{\sqrt{gD}} \end{aligned}$$

both to the full flow case $\_f$ and the partial flow case $\_p$ using $I\_p := I\_f$ yields

$$\begin{aligned} \frac{F\_p}{F\_f} &= \left(\frac{R\_p}{R\_f}\right)^{2/3}\cdot\left(\frac{D\_f}{D\_p}\right)^{1/2} = \left(\frac{D\_p}{D\_f}\right)^{1/6}\cdot\left(\frac{1+(2D\_f/B)}{1+(2D\_p/B)}\right)^{2/3} \\ &= \left(\frac{0.524}{2.50}\right)^{1/6}\cdot\left(\frac{(1+0.50)}{1+1.105}\right)^{2/3} = 0.945 \end{aligned}$$

which is approx. 5 % below the Froude condition $F = 1$. This is to be checked with the simulation results.

5.5.2 Implementation of Case 2

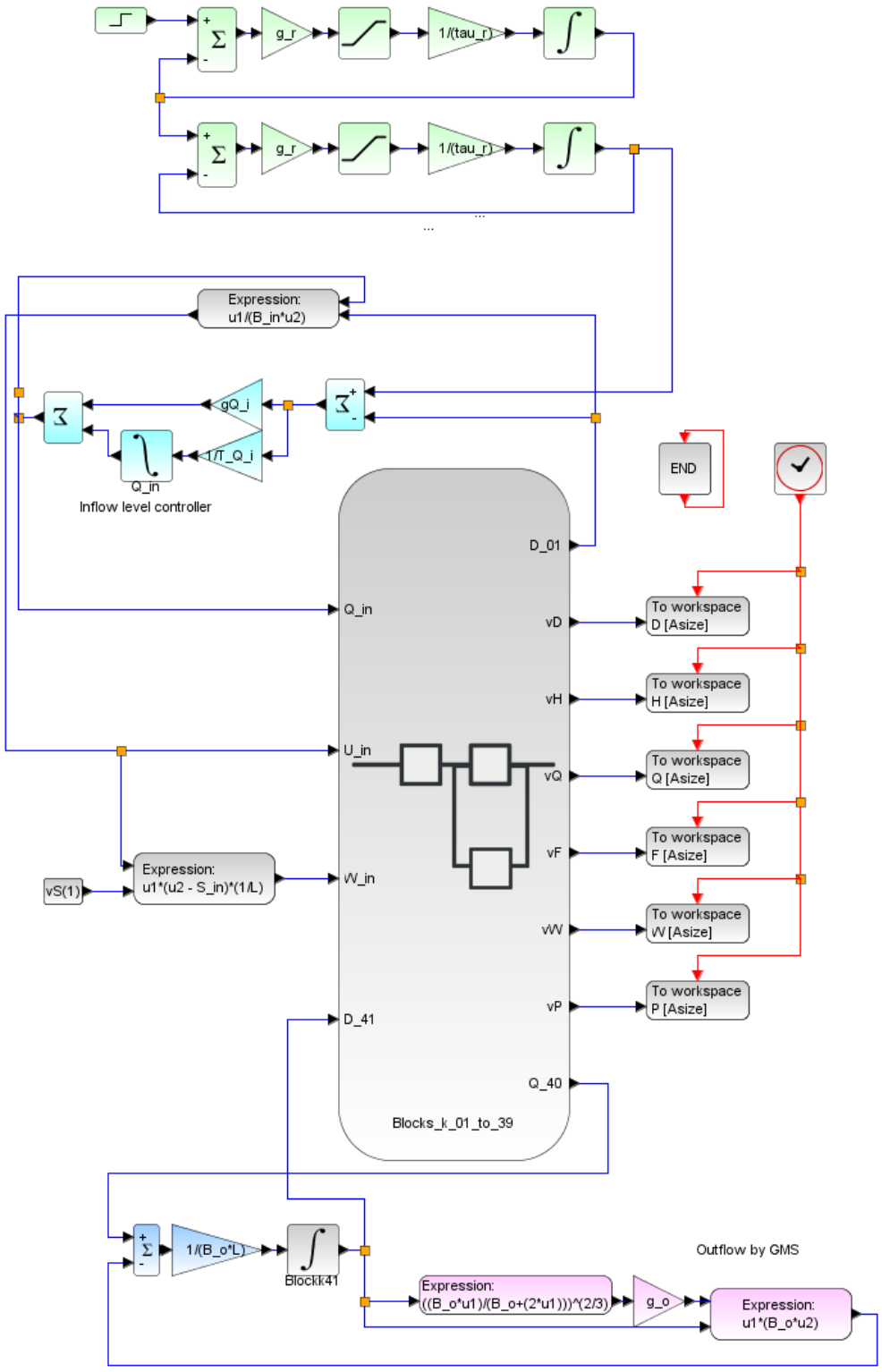


Figure 5.23: Main diagram for Case 2: Spillway with rising inflow level and outflow flap position fully open

```
// s_c5_02_b_context.sce
// Glf  2015_04_27
// outflow GMS
// Inflow with PI-LevelContr, setpoint ramp-up

g = 10.; Q_0 = 50.0; H_0 = 2.5;

lambda0 = 0.40; kap = 1.0;
//******************
//lambda = (1.25)*lambda0;
  lambda = (1.00)*lambda0;
//lambda = (0.80)*lambda0;
//lambda = (0.64)*lambda0;
//lambda = (0.50)*lambda0;
//******************
L = lambda*H_0;
S_0 = 0.0*(-1.0);
D_0 = H_0 - S_0;
B_0 = 10.;
U_0 = 2.0;
U_F = 5.0;

N= 20;      // no of Vol+Momentum-Segments

// channel inlet geometry
//*******************************************
vb_1 = ones(1,((2*N)+1)); vb_2 = vb_1;
vs_1 = vb_1; vs_2 = vs_1;

if lambda == (1.25)*lambda0 then
ik_E = 3; ik_R = 6;  delta_b = 0.144;
elseif lambda == 1.0*lambda0 then
ik_E = 4; ik_R = 8;  delta_b = 0.122;
elseif lambda == 0.80*lambda0 then
ik_E = 6; ik_R = 10; delta_b = 0.100;
elseif lambda == 0.64*lambda0 then
ik_E = 8; ik_R = 12; delta_b = 0.089;
else
ik_E = 10; ik_R = 16; delta_b = 0.079;
end
//*************************************

ik_T = ik_E + ik_R;
vib = ones(1,ik_T);
vd_b = ik_R*ones(1,(ik_T));
for ii = 1:ik_R
 vd_b(ii) =  ii;
end
vd_b = delta_b*vd_b/ik_R;
vincr_b  = vib + vd_b;

for ij = 1:(ik_T)
    vib(ij+1) = vib(ij)*vincr_b(ij);
end

vb_0 = vib(ik_T);
vb = vb_0*ones(1,ik_T);

for ijk = 1:ik_T
    nijk = ik_T+1 - ijk;
    nvib(nijk) = vib(ijk);
end

for k8 = 1:ik_T
    vb_2(k8) = nvib(k8);
    vs_2(k8) = nvib(k8);
end

vb_2 = vb_2 - vb_1;
vB = B_0*(lambda*H_0*vb_2 + vb_1);
vs_2 = vs_2 - vs_1;
vS = -(lambda*H_0*vs_2);
//*******************************************

// setting initial conditions
 D_min = +0.0001; D_max = 40*D_0;
 Q_min = +0.00001; Q_max = 40*Q_0;

h0 = 1.52; delh0 = 0.80; vh0 = ones(1,(2*N+1));
vH0 = delh0*vh0;  vD0 = vH0 - vS;

 q0  = 0.24; vq0 = ones(1,(2*N+1));
vQ0 = q0*Q_0*vq0;

vDdot0 = 0*vh0; ome = 100;
P_max = 100.0; P_min = 5.0;

// Inflow-(_i) boundary cond.
    B_in = vB(1) + (vB(1) - vB(2))*(1+delta_b);
    S_in = vS(1) + (vS(1) - vS(2))*(1+delta_b);
    D_in = vH0(1) - S_in;
    Q_in = vQ0(1);

// GMS-coefficient
  k_s = 100.;

// Friction-inclination of bottom
vD_E = H_0*vh0; vQ_E = U_F*B_0*H_0*vq0;
vdelSf = zeros(1,(2*N+1)); vSf = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vRtilda(kk) = ((vB(kk)*vD_E(kk))/ ...
                 (vB(kk)+2*vD_E(kk)))^(2/3);
  vI_E(kk) = (vQ_E(kk)/(vB(kk)* ...
              vD_E(kk)*k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI_E(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS(k4)  + 1.0*vSf(k4);
    vH0(k4) = vH0(k4) + 1.0*vSf(k4);
    vD0(k4) = vH0(k4) - 1.0*vS(k4);
end

//  outflow-(_o) boundary cond
    B_o = vB(kk+1);  S_o = vS(kk+1);
    D_o = vD0(kk+1); H_o = vH0(kk+1);
    Q_o = vQ0(kk+1);


// inflow level controller (PI)
  gQ_i = 20.; T_Q_i = 0.01;
// setpoint ramper
    t_r_1 = 100.0;
    r_1_0 = -vS(1)+delh0; r_1_1 = h0*H_0-vS(1);
    g_r = 10.0; u_up_r = +1.0; u_dn_r = -1.0;
    tau_r = 30.0; r_f_0 = r_1_0;
    r_hi = r_1_1 + 0.05; r_lo = r_1_0 - 0.05;
  // ************************************

  // Outflow by GMS
 g_o = k_s*(vI_E(40)^(1/2));

T_fin = 300.;

// Datatransfer to Plots
CC = 21;           // no of channels 20 + 1 for time
CN = 300;          // no of clockticks to Tfin
delT = T_fin/CN;   // readout-interval for clock ticks
Asize = 1.01*CC*CN; // size of data arrays
```

```
// s_c5_02_b_crunplot
//  Glf  2015_04_27

stacksize('max');
exec('s_c5_02_b_context.sce',-1);
importXcosDiagram('s_c5_02_b.zcos');
typeof(scs_m); scs_m.props.context;
Info = list();
Info = scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:11,  clf(kfig); end
vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor, ...
      rect=[0.,0.0,300,150.]);
xtitle("Q_2 to Q_40");xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor, ...
      rect=[0.,0.0,300,5.5]);
xtitle("D_1 to D_39");xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor, ...
      rect=[0.,0.0,300,4.0]);
xtitle("H_1 to H_39");xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor, ...
       rect=[0.,0.0,300,1.2]);
xtitle("F_2 to F_40");xgrid(1);

f5 = scf(5);
plot2d(W.time,W.values,vcolor, ...
      rect=[0.,-1.0,300,+1.0]);
xtitle("W_1 to W_39");xgrid(1);

f6 = scf(6);
plot2d(P.time,(P.values-g*D.values), ...
       vcolor,rect=[0.,-5.0,300,5.0]);
xtitle("deltaP*_1 to  deltaP*_39");xgrid(1);

//*************************************
eX = H_0*lambda*(1:1:N); eXend = 22;
for k6= 1:1:N, vSr(k6) = vS(2*k6-1); end
//
f7 = scf(7);clf();yD =D.values;
iD = yD(90,:); oD = yD(290,:);
iH = iD + vSr'; oH = oD + vSr';
vcolorP=[2,5];
plot2d(eX', [iH', oH'], vcolorP, ...
      rect=[0.,0.0,eXend,+4.0]);
xtitle("Longit.Profile H at 90s (bl), 290s (rd)");
xgrid(1);
//
f8 = scf(8);clf();yF =F.values;
iF = yF(90,:); oF = yF(290,:);
plot2d(eX', [iF', oF'], vcolorP, ...
      rect=[0.,0.0,eXend,+1.2]);
xtitle("Longit.Profile F at 90s (bl), 290s (rd)");
xgrid(1);
//
f9 = scf(9);clf();yW =W.values;
iW = yW(90,:); oW = yW(290,:);
plot2d(eX', [iW', oW'], vcolorP, ...
     rect=[0.,-1.0,eXend,+1.0]);
xtitle("Longit.Profile W at 90s (bl), 290s (rd)");
xgrid(1);
//
f10 = scf(10);clf();yP =P.values;
iP = yP(90,:)-g*yD(90,:); oP = yP(290,:)-g*yD(290,:);
plot2d(eX', [iP', oP'], vcolorP, ...
      rect=[0.,-5.0,eXend,+5.0]);
xtitle("Longit.Profile (P*-gD) at 90 s (bl), 290 s (rd)");
xgrid(1);
//
f11 = scf(11);clf();eY = 0.5*H_0*lambda*(1:1:(2*N)+1);
plot2d(eY',[0.1*vB', vS'],vcolorP, ...
      rect=[0.,-3.0,eXend,+3.0]);
xtitle("Longit.Channel Geometry, 0.1*B(x)(bl), S(x)(rd)");
xgrid(1);

// for assembling logitudinal contours
if lambda == (1.25)*lambda0 then
ac_125 = [eY',[0.1*vB',vS']]; ...
         aH_125 = [eX',iH']; aF_125 = [eX',iF'];
elseif lambda == 1.0*lambda0 then
ac_100 = [eY',[0.1*vB',vS']]; ...
         aH_100 = [eX',iH']; aF_100 = [eX',iF'];
elseif lambda == 0.80*lambda0 then
ac_080 = [eY',[0.1*vB',vS']]; ....
         aH_080 = [eX',iH']; aF_080 = [eX',iF'];
elseif lambda == 0.64*lambda0 then
ac_064 = [eY',[0.1*vB',vS']]; ....
         aH_064 = [eX',iH']; aF_064 = [eX',iF'];
else
ac_050 = [eY',[0.1*vB',vS']]; ....
         aH_050 = [eX',iH']; aF_050 = [eX',iF'];
end
```

5.5.3 Overview of simulations for Case 2

Three subcases shall be investigated.
The first is the nominal case with the transients of $Q, D, H, F, W, \Delta P^*$ due to the inflow level run-up. The second is checking longitudinal profiles of $H, F, W, \Delta P^*$ at full flow equilibrium for any differences to the second subcase of case 1.
The third subcase is to look into more detail of the initial low flow equilibrium. The longitudinal profiles of B, S and H, F are given with different compartment lengths L. Here a further value $L = 0.50\ m$ is inserted, as the shape of the level surface is shorter over x. The similarity rule from case 1 would indicate a factor of $U\_F\_p/U\_f\_f = 2.30/5.0 \approx 0.46$.

5.5.4 Simulation results for Case 2

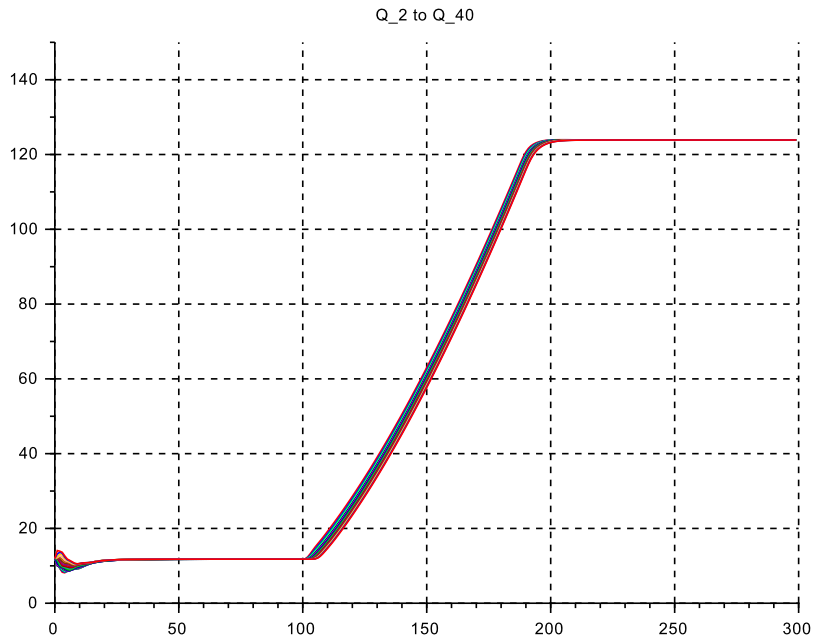


Figure 5.24: C 2: Transient Q for $L = 1$ $[m]$

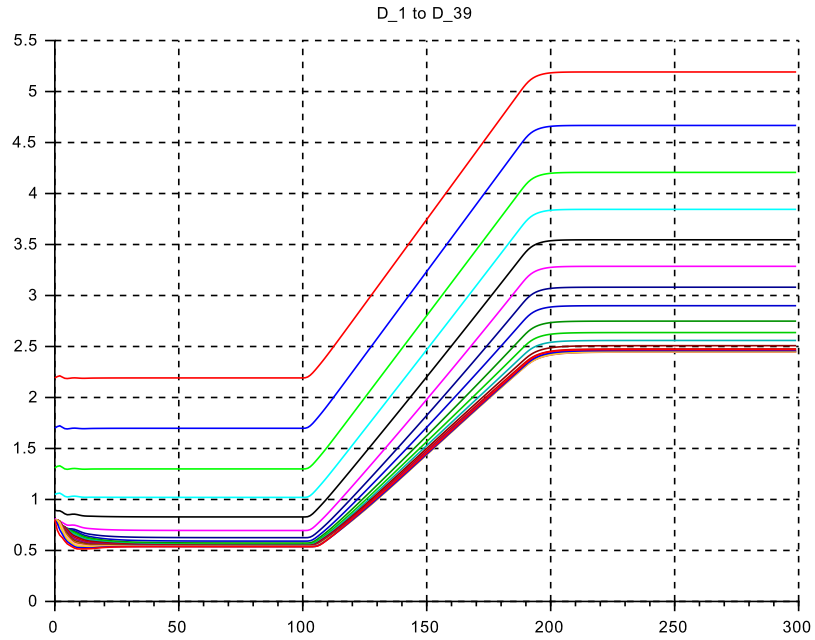


Figure 5.25: C 2: Transient D for $L = 1$ $[m]$

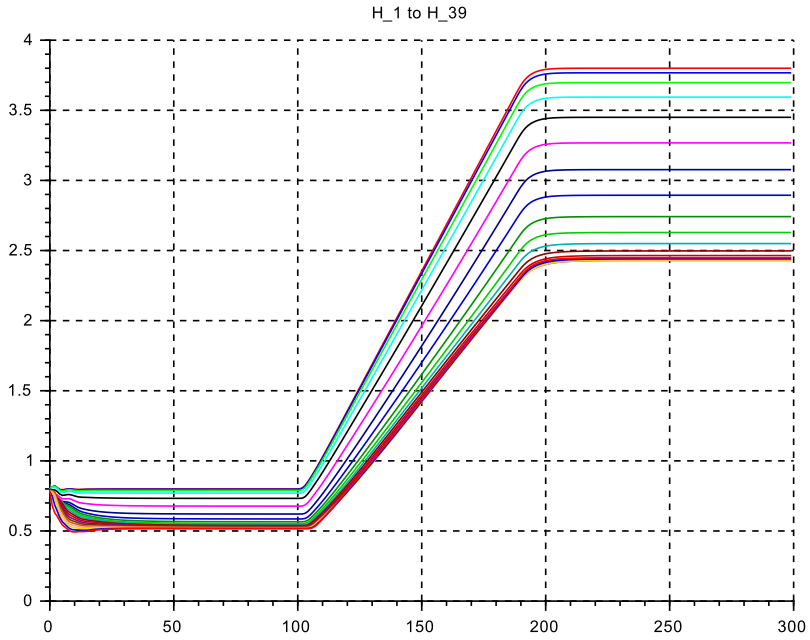


Figure 5.26: C 2: Transient H for $L = 1$ $[m]$

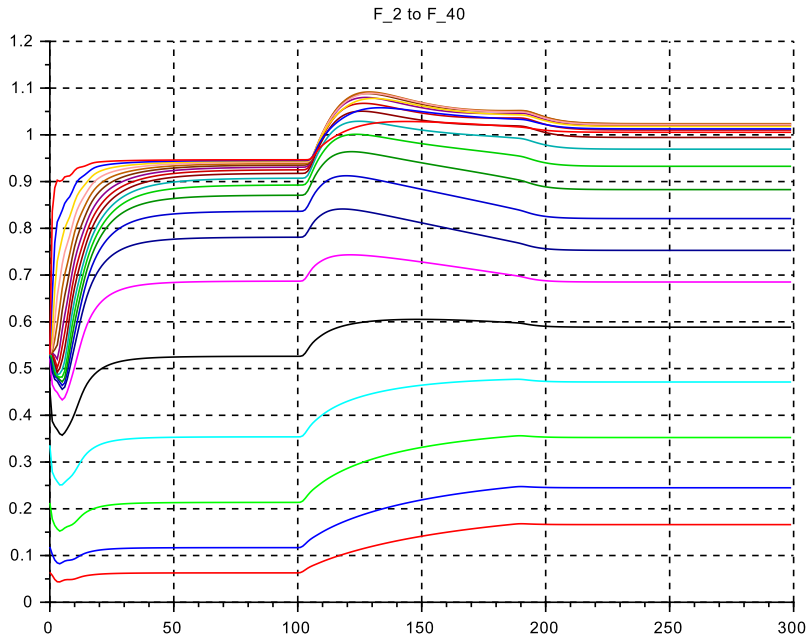


Figure 5.27: C 2: Transient F for $L = 1$ $[m]$

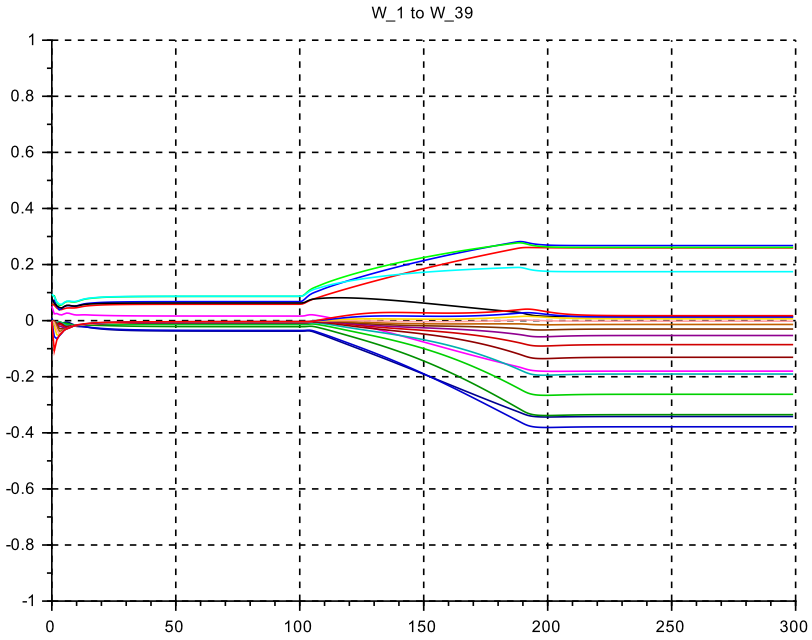


Figure 5.28: C 2: Transient W for $L = 1$ $[m]$

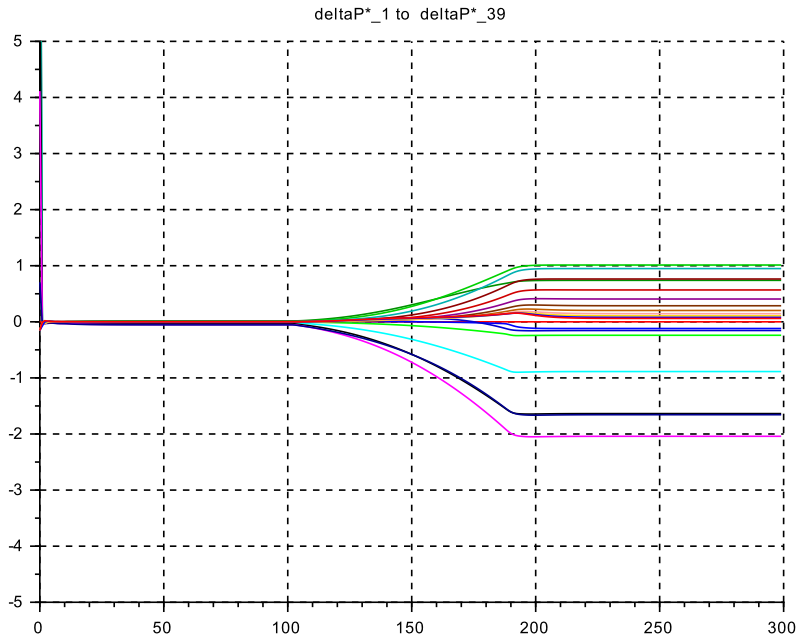


Figure 5.29: C 2: Transient $P^* - gD$ for $L = 1$ $[m]$

Longitudinal profiles at full flow (time 290 s)

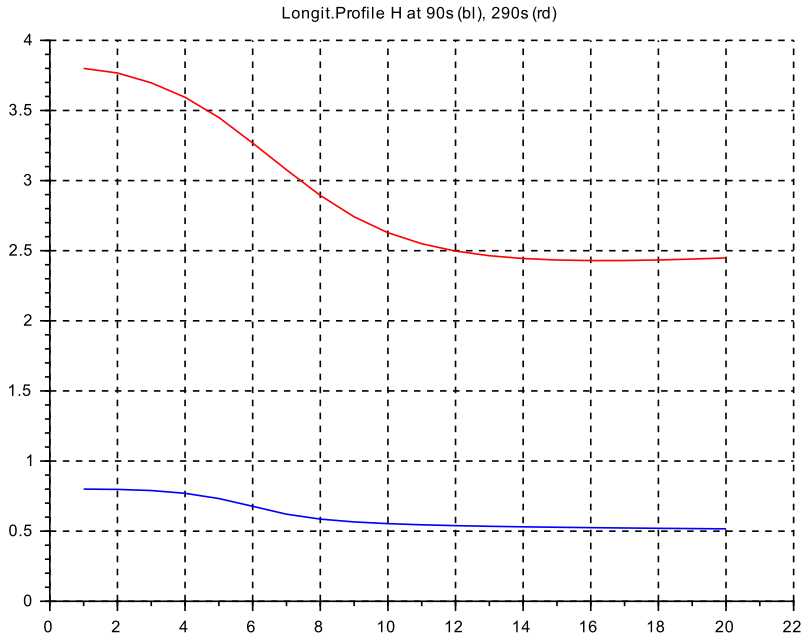


Figure 5.30: C 2: Profile for H for $L = 1$ $[m]$

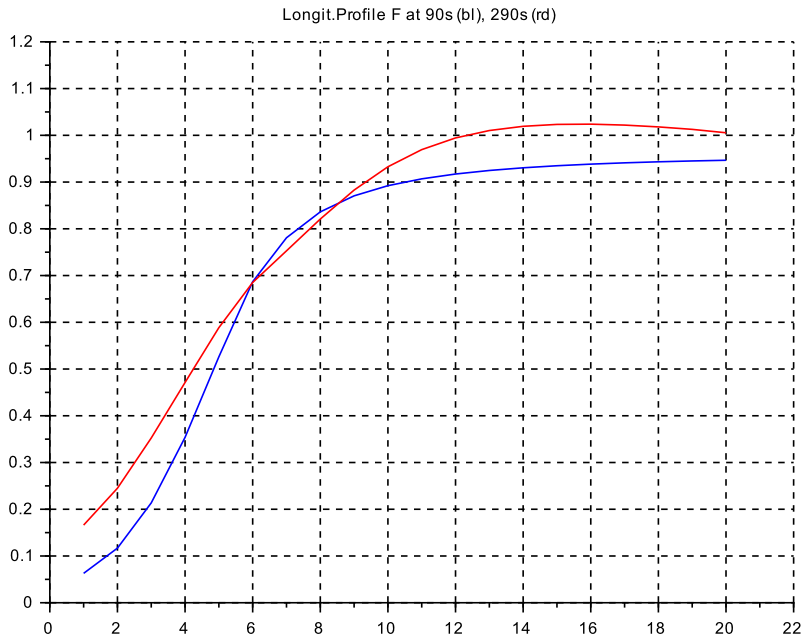


Figure 5.31: C 2: Profile for F for $L = 1$ $[m]$

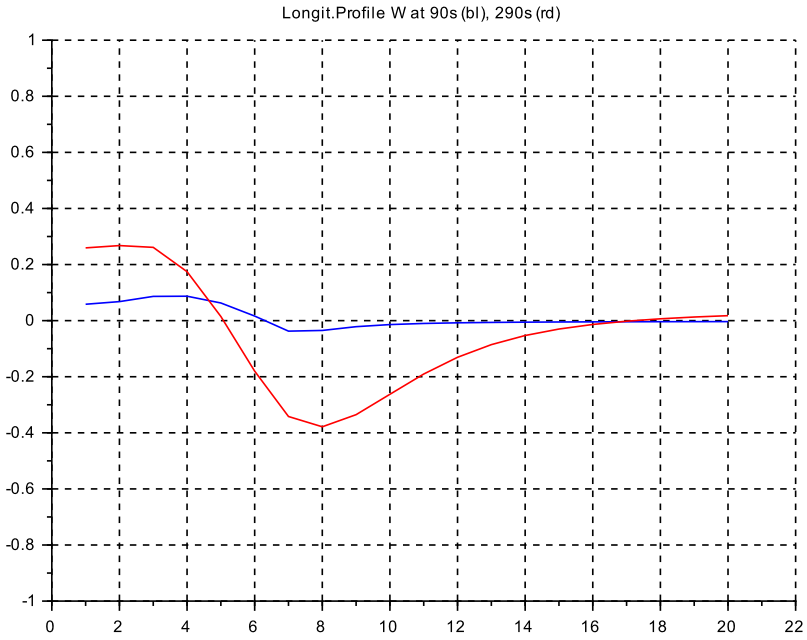


Figure 5.32: C 2: Profile for W for $L = 1$ $[m]$

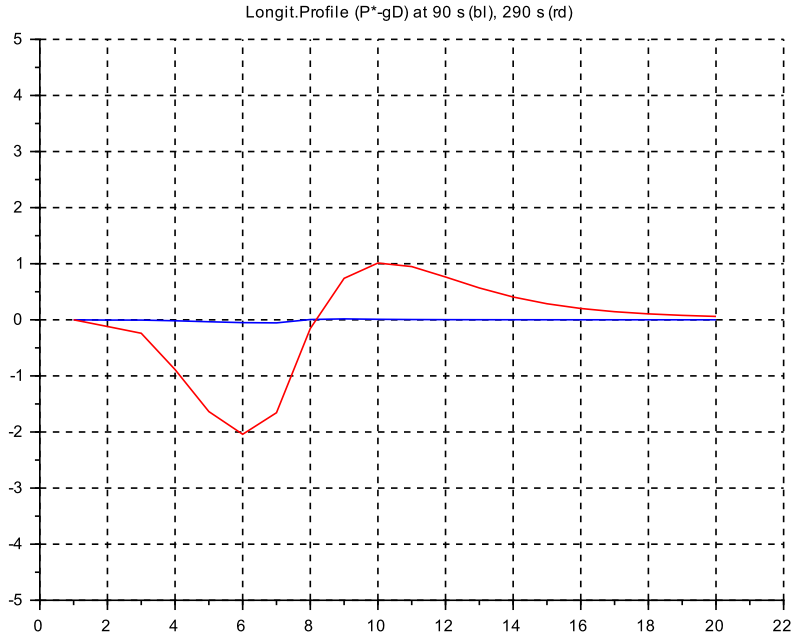


Figure 5.33: C 2: Profile for $P^* - gD$ for $L = 1$ $[m]$

Assembly of longitudinal profiles at time $90\ s$ **for low initial flow** ($Q\_in \approx 12\ m^3/s$)
for $L =\ 0.50,\ 0.64,\ 0.80,\ 1.0,\ 1.25\ m$

```
// s_c5_02_b_contassemb.sce
// Glf 2015_08_13

  for kfig = 12:1:16,  clf(kfig); end
 eXend = 30;

vcolor3 = [ 2, 5, 1, 13, 17];
vcolor4 = [ 2, 5, 1, 13, 17, 2, 5, 1, 13, 17];

f12 = scf(12);
plot2d(...
[ac_125(:,1),ac_100(:,1),ac_080(:,1), ...
ac_064(:,1), ac_050(:,1),...
 ac_125(:,1),ac_100(:,1),ac_080(:,1), ...
 ac_064(:,1), ac_050(:,1)],...
[ac_125(:,2),ac_100(:,2),ac_080(:,2), ...
ac_064(:,2), ac_050(:,2) ...
 ac_125(:,3),ac_100(:,3),ac_080(:,3), ...
 ac_064(:,3), ac_050(:,3)],...
vcolor4,rect=[0.,-2.0,eXend,+3.0]);
xtitle("C 2: geometry contour 0.1*B and S");
legend("B: L = 1.25 m","1.00 m","0.80 m", ...
            "0.64 m", "0.50 m",...
       "S: L = 1.25 m","1.00 m","0.80 m", ...
            "0.64 m", "0.50 m")
xgrid(1);
```

```
f13 = scf(13);
plot2d(...
[aH_125(:,1),aH_100(:,1),aH_080(:,1), ...
   aH_064(:,1),aH_050(:,1)],...
[aH_125(:,2),aH_100(:,2),aH_080(:,2), ...
   aH_064(:,2),aH_050(:,2)],...
vcolor3,rect=[0.,0.4,eXend,+1.0]);
xtitle("C 2: contours H");
legend("H: L = 1.25 m","1.00 m","0.80 m", ...
            "0.64 m", "0.50 m")
xgrid(1);


f14 = scf(14);
plot2d(...
[aF_125(:,1),aF_100(:,1),aF_080(:,1), ...
   aF_064(:,1),aF_050(:,1)],...
[aF_125(:,2),aF_100(:,2),aF_080(:,2), ...
   aF_064(:,2),aF_050(:,2)],...
vcolor3,rect=[0.,0.0,eXend,+1.2]);
xtitle("C 2: contours F");
legend("F: L = 1.25 m","1.00 m","0.80 m", ...
            "0.64 m", "0.50 m")
xgrid(1);
```

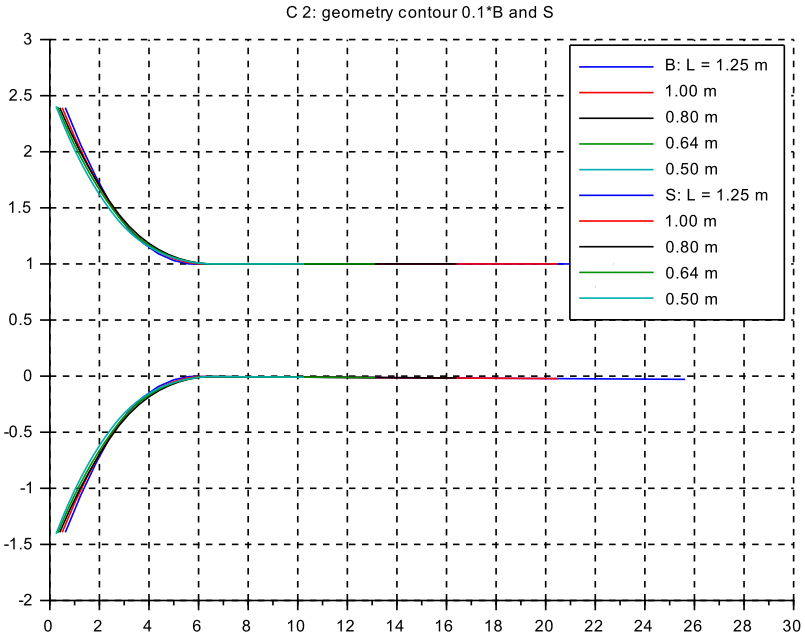


Figure 5.34: C 2: Assembly of Profiles for $0.10 \cdot B$ and S

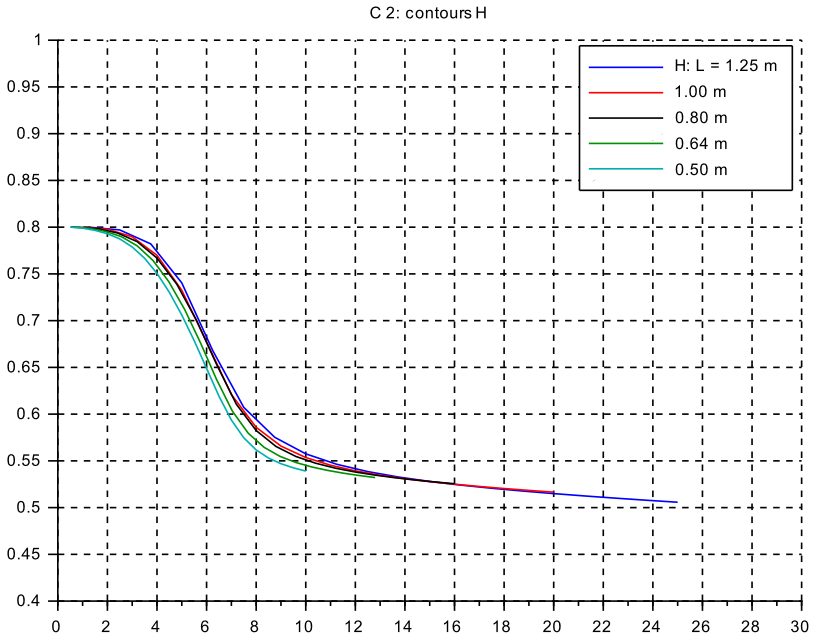


Figure 5.35: C 2: Assembly of Profiles for H

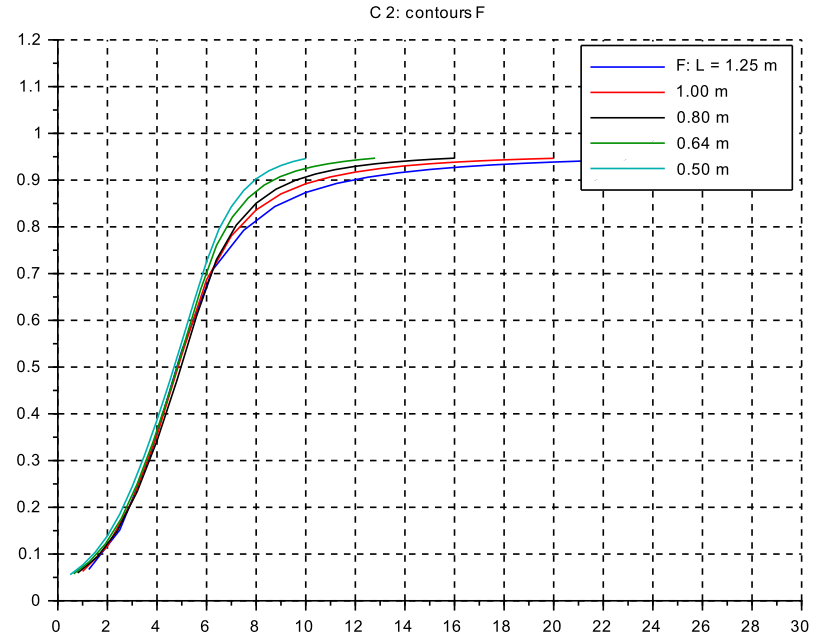


Figure 5.36: C 2: Assembly of Profiles for F

5.5.5 Discussion for Case 2

Transients for the nominal parameter set

Q Initially there is a small settling transient decaying within 10 s to 12.1 m^3/s. Then the flow run-up is as expected. No sloshing transients are visible, due to the first order filtered setpoint trajectory.

D, H as to be expected from $Q(t)$. The initial outflow depth D_o settles at 0.545 m close to what was predicted, and the final value is $D\_o = 2.50$ m

F initially stabilizes at $F\_o = 0.946$, which correlates well with the predicted value. For full flow it rises to $F = 1.005$. During the run-up transient the values of $F(t)$ are consistently higher than the final equilibrium values (due to the filling of the channel).
If the initial flow is decreased significantly, for instance to ≤ 6 m^3/s, then the initial equilibrium values of F increase significantly, to $F \geq 2$, as the bottom slope is too steep to allow a subcritical stationary flow.

W and

ΔP^* converging both to zero at both equilibria, as expected.

Longitudinal profiles for the nominal parameter set at full flow

B, S The same confusor geometry contour is applied.

H ,

F ,

W , and

ΔP^* are the same as for subcase 2 of case 1, as expected.

Longitudinal profiles for different compartment lengths L at low flow conditions, $Q = 12$ m^3/s

A further value $L = 0.50$ m is inserted, in order to generate about the same number points from the length discretisation for the steep part of the $H(x)$-descent. From the similarity found in case 1, that would be a factor p_f applied to the nominal length $L = 1.0$ m

$$p\_f = \frac{U\_F\_p}{U\_F\_f} = \sqrt{\frac{D\_p}{D\_f}} := \sqrt{\frac{0.525}{2.50}} \approx 0.50$$

B, S The contour is manually trimmed to a small spread around the nominal contour, see Fig.5.34

H The spread of $H(x)$ around the nominal shape is surprisingly small, even for $L = 1.25$ m with only about 4 data points on the steep part (!) Then there is a long tail to the outflow location, with the flow slowly accelerating to near equilibrium.
For $L = 0.64$ m and 0.50 m, the shapes of $H(x)$ tend to spread more to the left (smaller x-values). Part of it is due to the inflow location of the confusor moving more to the left. And the approach to near equilibrium is faster, similar to subcase 3 in case 1.

F Again two phases are visible on the F-profiles. The first is the confusor part, up to position $x = 6$ m, where no effect of L is discernible. The second is the approach to equilibrium in the constant cross section area, where the profiles $F(x, L)$ tend to spread out (faster rising for smaller L). The same effect is visible in subcase 3 of case 1. However the end-points are nearly identical, and at $F = 0.945$ which closely matches the value pre-calculated above.

.

5.6 Case 3: 'Dam Break'

5.6.1 Motivation

As mentioned in the introduction to this chapter this is the maximum possible incident, which is often studied using standard grid methods for solving the PDE. Thus this may serve to illustrate any deviations with the model used here where spacial grid elements are much longer.
It may serve (with small adaptations) to investigate a less catastrophic incident, when the accumulated sediments in the reservoir are to be flushed in the downstream river by opening a gate at the bottom of the dam.

5.6.2 Modeling and Data

Two subcases A and B are investigated, where A has a compartment length $L = 25\ m$ that is an overall forebay length of $L_{tot} = 525\ m$ which is typical for a small river hydro installation. And subcase B puts the focus on the very first phase $\approx 2\ s$ after a nearly instantaneous dam break (within $0.040\ s$), with compartment length $L = 1.0\ m$, in order to catch the transient of all main variables as well as the longitudinal profiles.
The plant model is taken from Case 1 with nominal depth $D = 2.50\ m$, and constant width $B = 10.0\ m$. The outflow flap shall move rqpidly compared to the filling time constant of the adjacent volume balance compartment: $T\_f = L/U$ with $U \approx 2.5\ m/s$ that is $T\_f = 10\ s$. Thus the flap shall move in $1.0\ s$ through its full stroke of $2.5\ m$ for subcase A, and in $0.040\ s$ for subcase B.
Finally the inflow shall be kept fixed at $Q\_in = 5\ m^3/s$ and this value shall be used to calculate the bottom friction slope, with $D = 2.5\ m$.

5.6.3 Implementation

The main diagram `s_c5_03.zcos` is carried over from case 1 but with the inflow level control loop replaced by a constant flow source Q_in. Then for the `x_.sce`-files:

```
// s_c5_03_context.sce
// Glf  2017_06_15
// dambreak; by fast outflow flap lowering
// Inflow constant, small

g = 10.;
Q_0 = 50.0;
H_0 = 2.5; kap = 1.0;
//*****************************************
// subcase A:
   lambda = 10.0; T_fin = 500.0; CN = 500;tau_f = 0.40;
// subcase B:
// lambda = 0.4; T_fin =  50.0; CN = 500; tau_f = 0.016;
//*****************************************
L = lambda*H_0;
S_0 = 0.0*(-1.0);
D_0 = H_0 - S_0;
B_0 = 10.;
U_0 = 2.0;
N= 20;      // no of Vol+Momentum-Segments

// Channel-Geometry
// rectangular-cross sect with constant (nominal) width
vb_1 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*(vb_1);

//horizontal  bottom
vs = -[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0];
vS = S_0*(vs);
```

```
 D_min = +0.001; D_max = 40*D_0;
 Q_min = +0.001; Q_max = 40*Q_0;

h0 = 1.00;  vh0 = ones(1,(2*N+1));
vH0 = h0*H_0*vh0; vD0 = vH0 - vS;

q0  = 0.10; vq0 = ones(1,(2*N+1));
vQ0 = q0*Q_0*vq0;

vDdot0 = 0*vh0; ome = 100;
P_max = 100.0; P_min = 0.0;

// Inflow-(_in) boundary cond.
    B_in = vB(1);   S_in = vS(1);
    D_in = vD0(1); H_in = D_in + S_in;
    Q_in = vQ0(1);

// GMS-coefficient
  k_s = 100.;
// Friction-inclination of bottomv'
vD_E = D_0*vh0; vQ_E = q0*Q_0*vq0;

vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vRtilda(kk)=((vB(kk)*vD_E(kk))/ ...
     (vB(kk)+2*vD_E(kk)))^(2/3);
  vI_E(kk) = (vQ_E(kk)/(vB(kk)*vD_E(kk) ...
     *k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI_E(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS(k4)  + vSf(k4);
    vH0(k4) = vH0(k4) + vSf(k4);
```

```
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow-(_o) boundary cond
    B_o = vB(kk+1); S_o = vS(kk+1);
    D_o = vD0(kk+1); H_o = vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow fixed to Q_in

// outflow by flap
```

```
  dH_o = (Q_o/(((2*g)^0.5)*(2/3)*B_o))^(2/3);
  t_f_1 = 1.;  f_1_0 = D_o - dH_o; f_1_1 = S_o;
  g_f = 10.0; u_up_f = +1.0; u_dn_f = -1.0;
   K_f_0 = f_1_0;
  gQ_K = 1.0*B_o*((1.0*g)^(0.5));

// Datatransfer to Plots
CC = 21;           //no of channels 20 + 1 for time
delT = T_fin/CN;   //readout-interval for clock ticks
Asize = 1.01*CC*CN;//size of data arrays
```

```
// s_c5_03_crunplot
//  Glf  2017_06_15
// dam-break

stacksize('max'); exec('s_c5_03_context.sce',-1);
importXcosDiagram('s_c5_03.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:11,  clf(kfig); end
vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[0.,0.0,T_fin,50.]);
xtitle("Q_2 to Q_40");
xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[0.,0.0,T_fin,2.6]);
xtitle("D_1 to D_39");
xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[0.,0.0,T_fin,2.6]);
xtitle("H_1 to H_39");
xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[0.,0.0,T_fin,1.2]);
xtitle("F_2 to F_40");
xgrid(1);

f5 = scf(5);
plot2d(W.time,W.values,vcolor,rect=[0.,-2.,T_fin,+0.2]);
xtitle("W_1 to W_39");
xgrid(1);

f6 = scf(6);
plot2d(P.time, (P.values - g*D.values), vcolor,...
       rect=[0.,-10.0,T_fin,+2.0]);
xtitle("deltaP*_1 to  deltaP*_39");
xgrid(1);
//*******************************************************
eX = H_0*lambda*(1:1:N); eXend = L*22;
for k6= 1:1:N,
  vSr(k6) = vS(2*k6-1);
end

//*******************************
if lambda == 10.0 then // subcase A:
```

```
 vT =[75,150,225,300]; vR=[75,150,225,300];
 else // subcase B:
 vT =[1.6,2.4,3.2,50]; vR=[16,24,32,500];end
//*******************************
vs = ['0','0','0','0'];
for k8 = 1:1:4,
  vs(k8) = msprintf('%5.2f',vT(k8));end
vcolorP=[1,2,5,13];

f7 = scf(7); clf(); yQ =Q.values;
vQ=[yQ(vR(1),:)',yQ(vR(2),:)',yQ(vR(3),:)',yQ(vR(4),:)'];
plot2d(eX',vQ, vcolorP, rect=[0.,-0.0,eXend,+50.]);
vtitle =["Longit.Profile Q at",vs(1),"s (bk)",",vs(2),...
"s (bl)",",vs(3),"s (rd)",",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f8 = scf(8); clf(); yD =D.values;
iD = yD(vR(1),:); mD = yD(vR(2),:);
oD = yD(vR(3),:); eD = yD(vR(4),:);
plot2d(eX', [iD', mD', oD', eD'], vcolorP,...
 rect=[0.,0.0,eXend,+2.6]);
vtitle =["Longit.Profile D at",vs(1),"s (bk)",",vs(2),...
"s (bl)",",vs(3),"s (rd)",",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f9 = scf(9); clf(); yF =F.values;
iF = yF(vR(1),:); mF = yF(vR(2),:);
oF = yF(vR(3),:); eF = yF(vR(4),:);
plot2d(eX', [iF', mF', oF', eF'], vcolorP,...
 rect=[0.,0.0,eXend,+1.2]);
vtitle=["Longit.Profile F at",vs(1),"s (bk)",",vs(2),...
"s (bl)",",vs(3),"s (rd)",",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f10 = scf(10);clf(); yW =W.values;
iW = yW(vR(1),:); mW = yW(vR(2),:);
oW = yW(vR(3),:); eW = yW(vR(4),:);
plot2d(eX', [iW', mW', oW', eW'], ...
vcolorP, rect=[0.,-1.00,eXend,+0.20]);
vtitle=["Longit.Profile W at",vs(1),"s (bk)",",vs(2),...
"s (bl)",",vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f11 = scf(11);clf(); yP =P.values;
vP=...
[yP(vR(1),:)'-g*yD(vR(1),:)',yP(vR(2),:)'-g*yD(vR(2),:)',...
 yP(vR(3),:)'-g*yD(vR(3),:)',yP(vR(4),:)'-g*yD(vR(4),:)'];
plot2d(eX',vP, vcolorP, rect=[0.,-5.0,eXend,+5.0]);
vtitle = ["Longit.Profile (P*-gD) at",vs(1),"s (bk)",", ...
  vs(2),"s (bl)",",vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle);xgrid(1);
```

The transients and the longitudinal profiles for each variable Q, D, F are paired on one page.

5.6.4 Simulation results

Subcase A, $L = 25\ m$

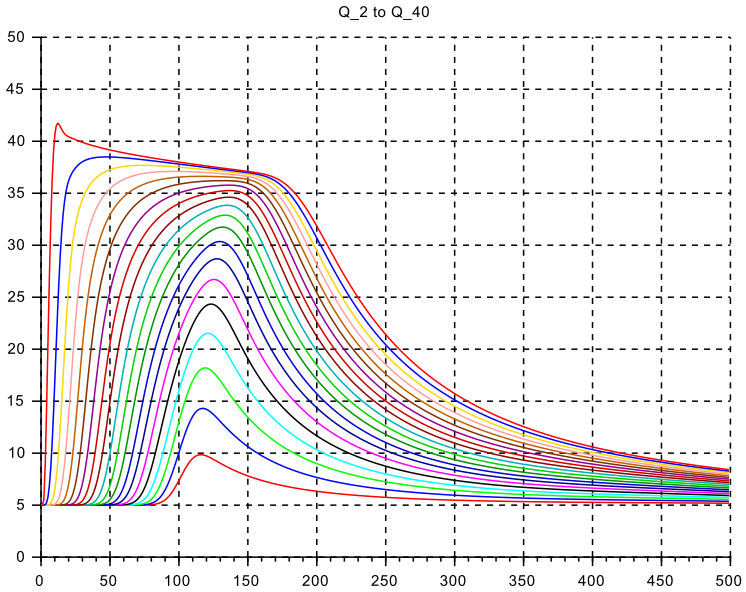


Figure 5.37: C 3: Transient $Q(t)$ for $L = 25\ [m]$

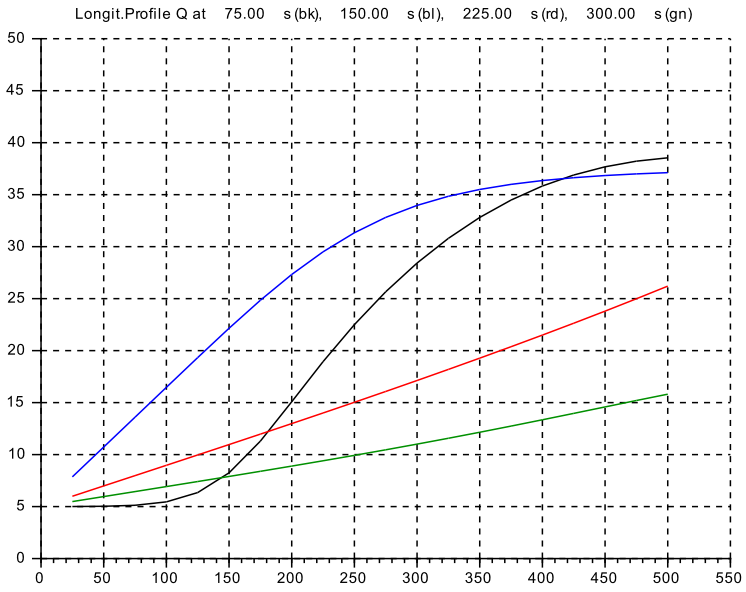


Figure 5.38: C 3: Longit.Profile $Q(x)$ for $L = 25\ [m]$

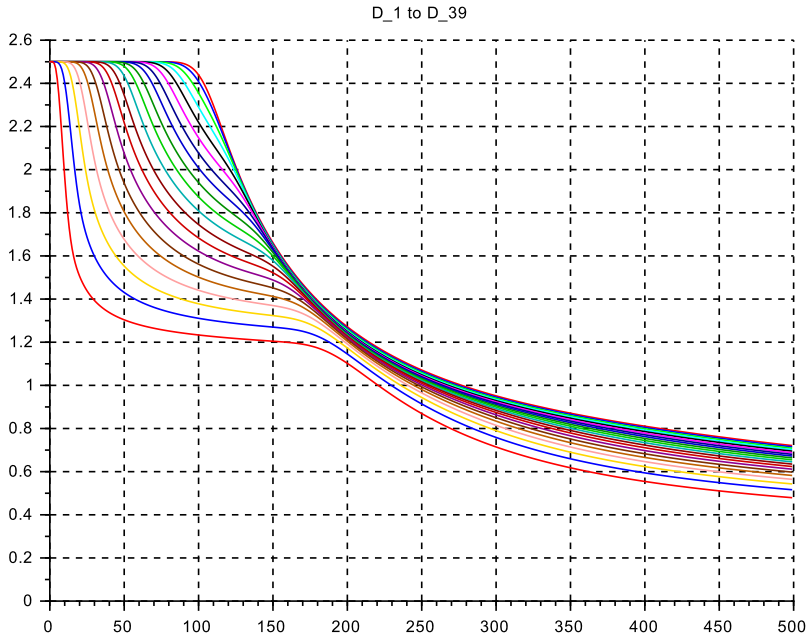


Figure 5.39: C 3: Transient $D(t)$ for $L = 25$ $[m]$

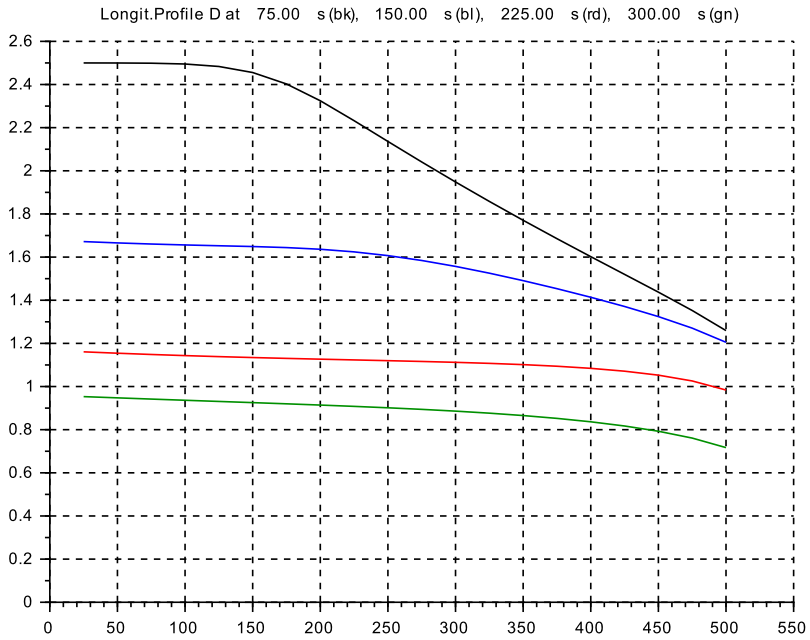


Figure 5.40: C 3: Longit.Profile $D(x)$ for $L = 25$ $[m]$

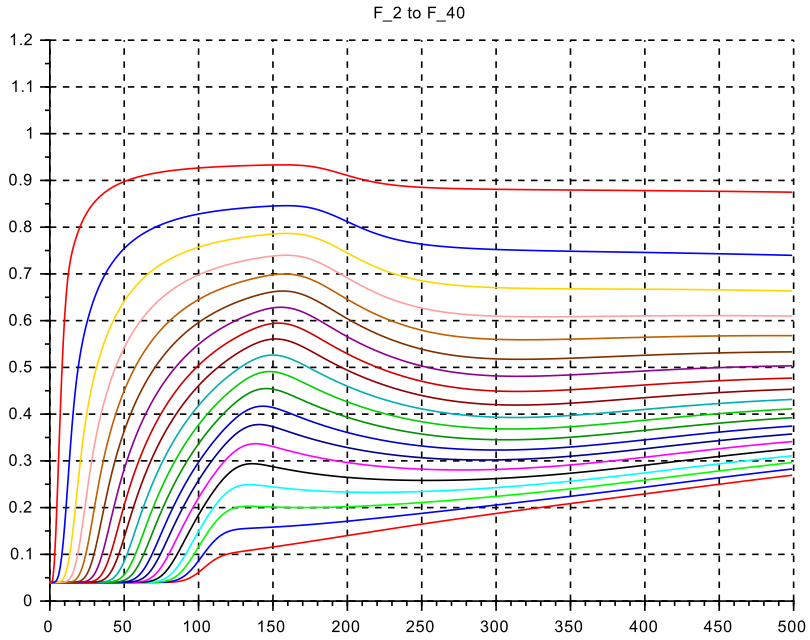


Figure 5.41: C 3: Transient $F(t)$ for $L = 25$ $[m]$

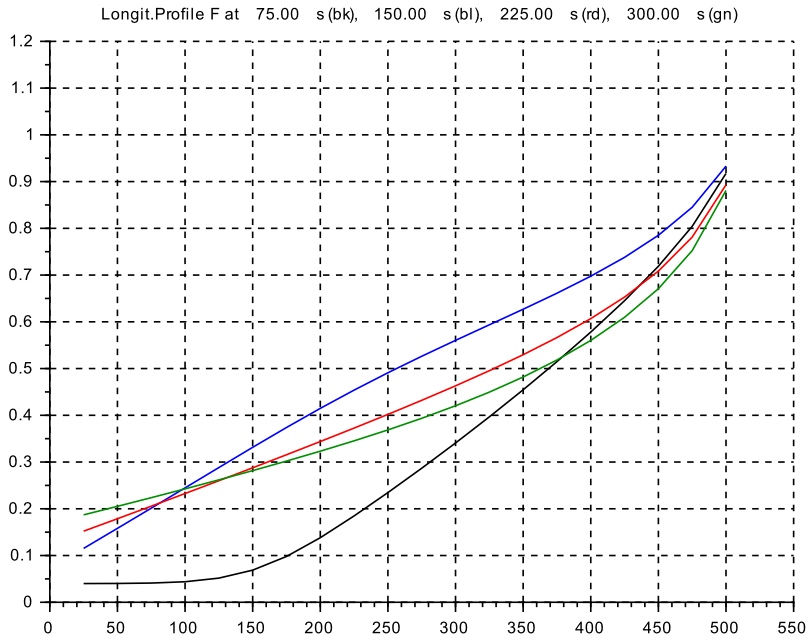


Figure 5.42: C 3: Longit.Profile $F(x)$ for $L = 25$ $[m]$

Subcase B, $L = 1.0\ m$

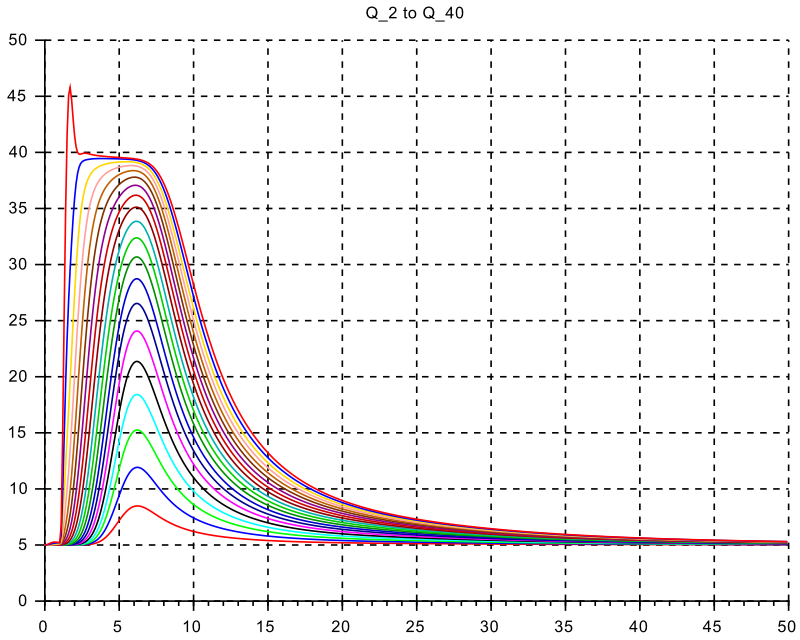


Figure 5.43: C 3: Transient $Q(t)$ for $L = 1.0\ [m]$

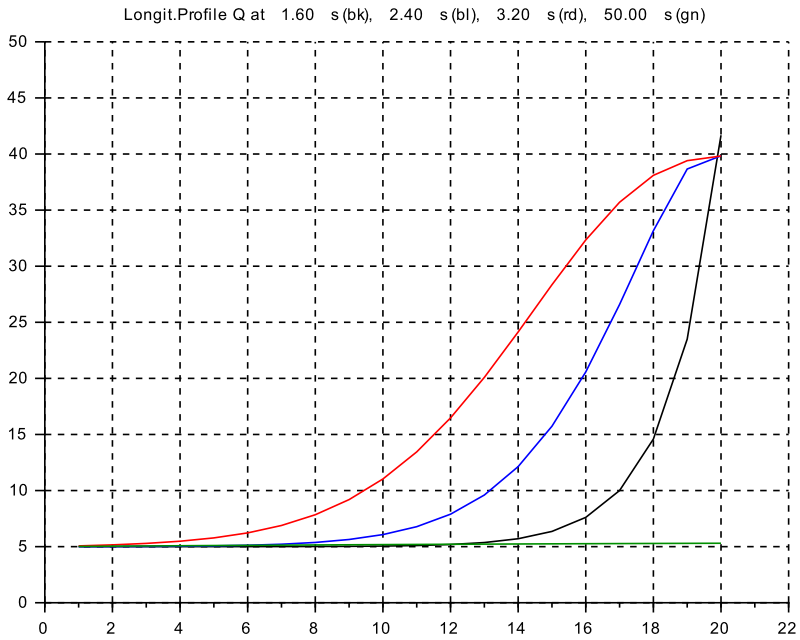


Figure 5.44: C 3: Longit.Profile $Q(x)$ for $L = 1.0\ [m]$

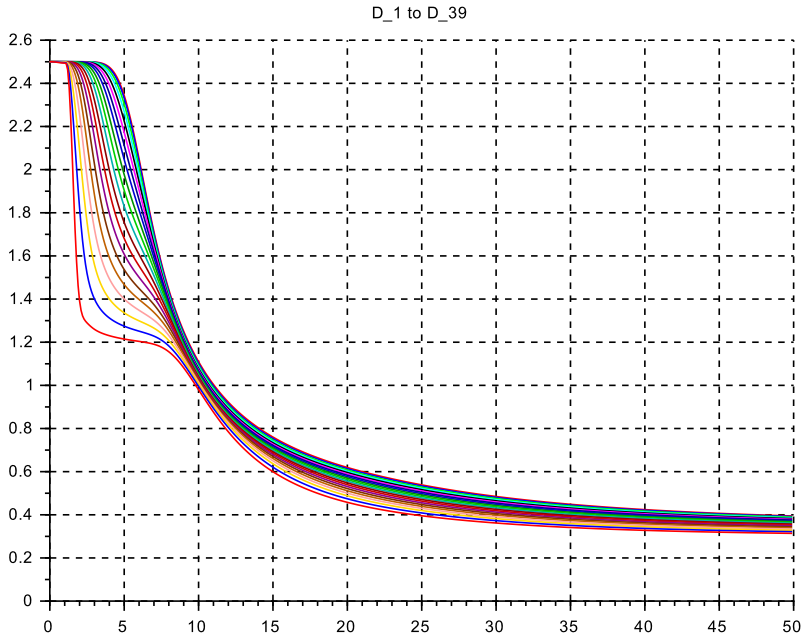


Figure 5.45: C 3: Transient $D(t)$ for $L = 1.0$ $[m]$

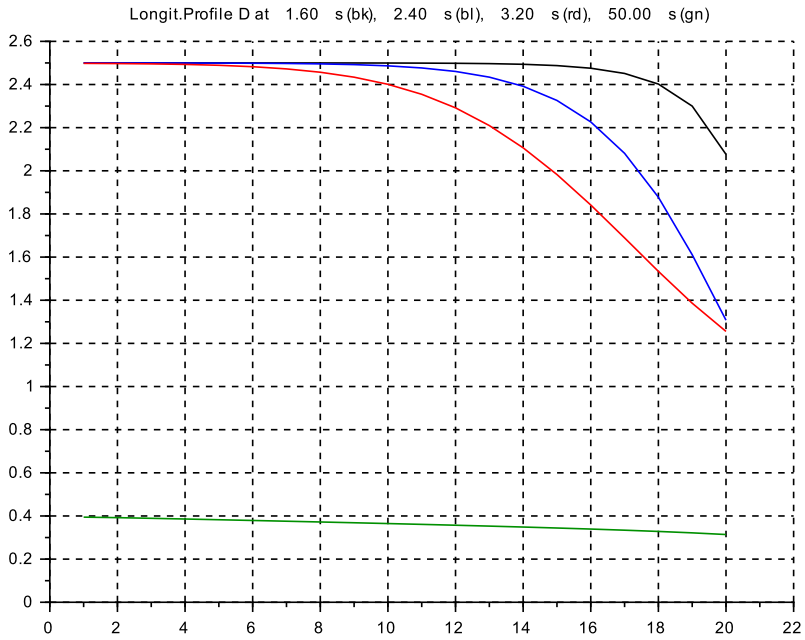


Figure 5.46: C 3: Longit.Profile $D(x)$ for $L = 1.0$ $[m]$

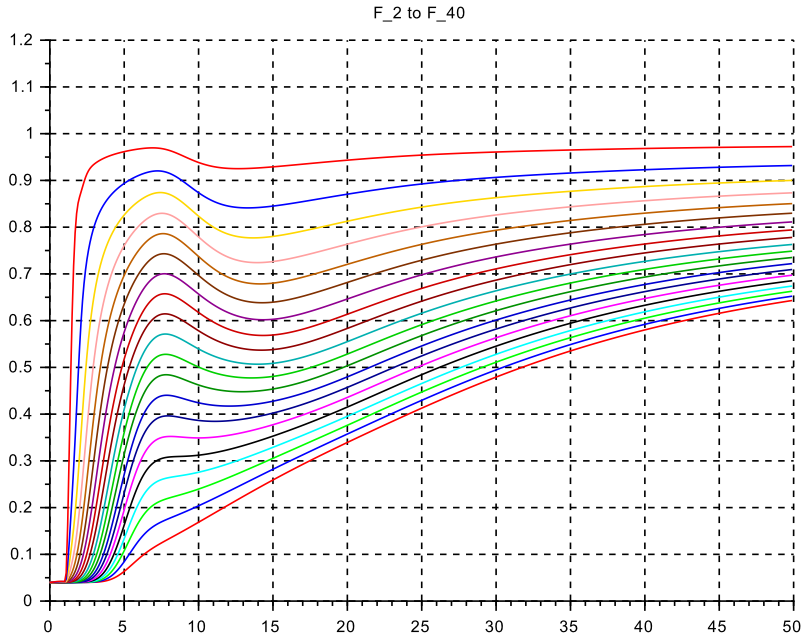


Figure 5.47: C 3: Transient $F(t)$ for $L = 1.0$ $[m]$

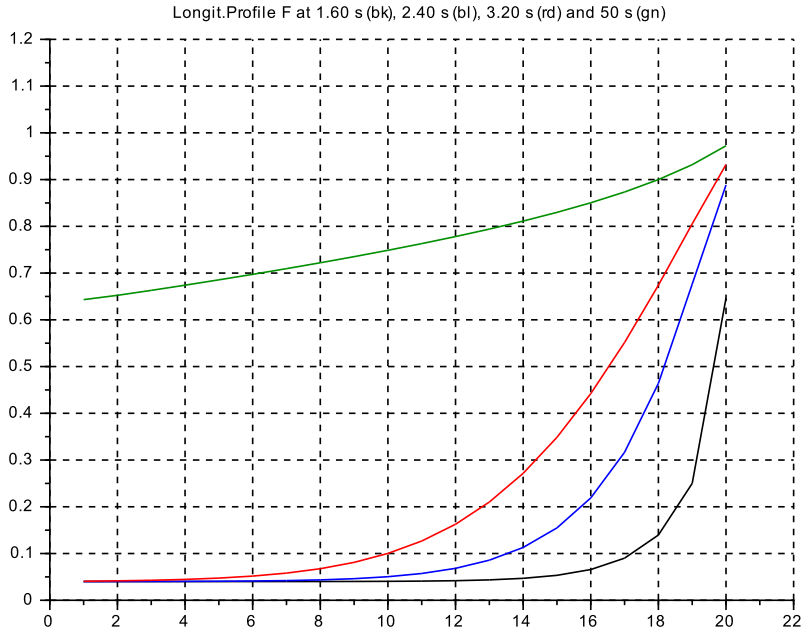


Figure 5.48: C 3: Longit.Profile $F(x)$ for $L = 1.0$ $[m]$

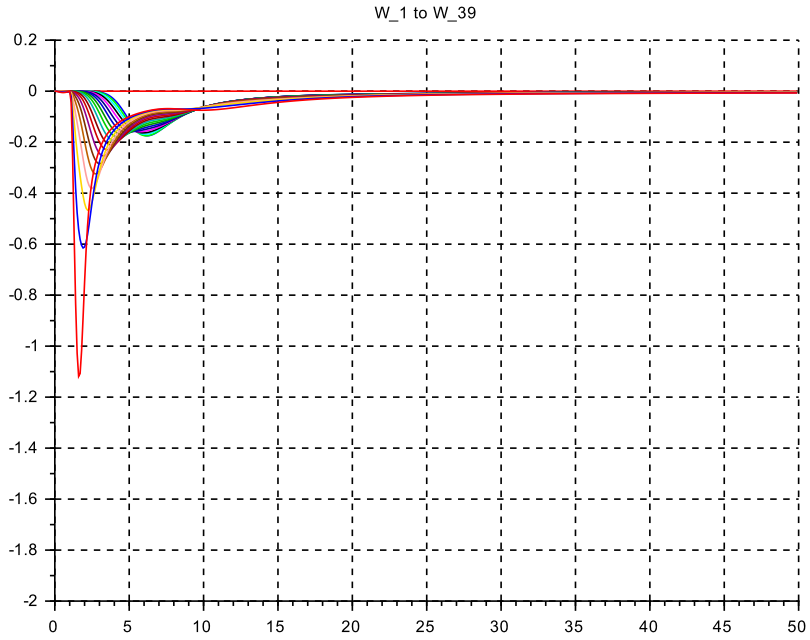


Figure 5.49: C 3: Transient $W(t)$ for $L = 1.0$ $[m]$

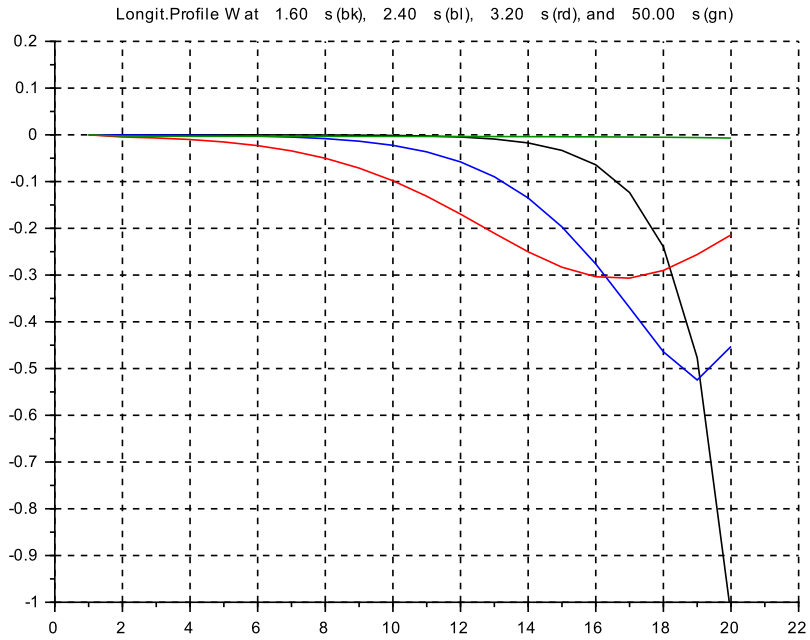


Figure 5.50: C 3: Longit.Profile $W(x)$ for $L = 1.0$ $[m]$

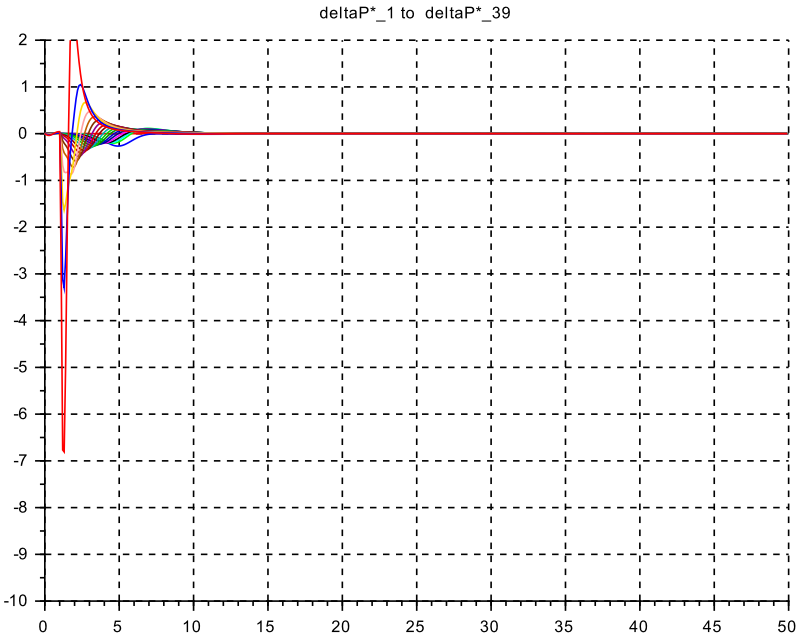


Figure 5.51: C 3: Transient $\Delta P^*(x)$ for $L = 1.0\ [m]$

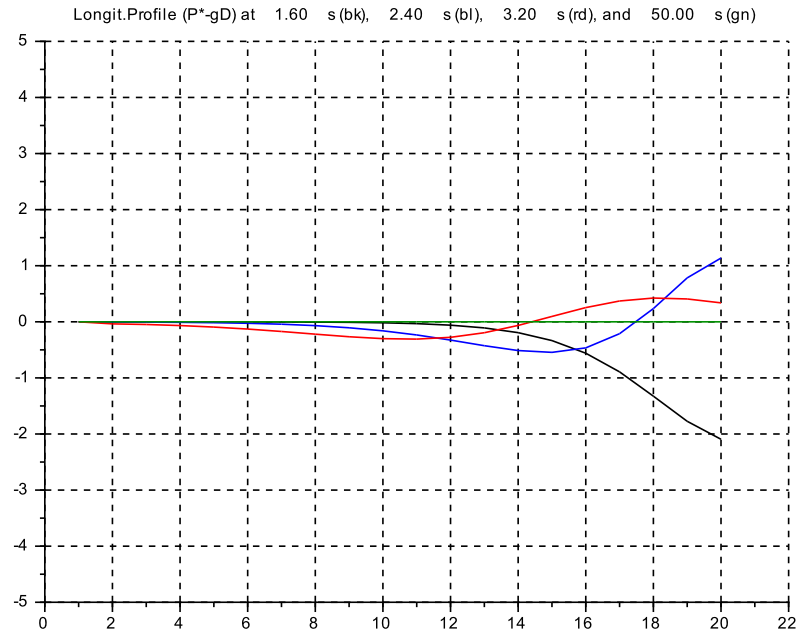


Figure 5.52: C 3: Longit.Profile $\Delta P^*(x)$ for $L = 1.0\ [m]$

5.6.5 Discussion for Case 3

subcase A, $L = 25\ m$

Q At the outflow location the peak value is $\approx 42\ m^3/s$ at $t \approx 10\ s$ and decays to $\approx 37\ m^3/s$ at $t = 100...150\ s$ for the last 5 compartments. There is a typical plateau phase. At the end of the time window $t = 500\ s$, Q decays to $\approx 8\ m^3/s$, which includes the $5\ m^3/s$ from the steady state inflow. In other words the emptying transient is at its end.
At the inflow location, Q is at its steady state value up to $t \approx 90\ s$, then rises to $\approx 10\ m^3/s$ at $t = 120\ s$ and the decays to the steady state.
The longitudinal profiles snapshots show how the emptying transient progresses over the length of the forebay.

D At the inflow location D stays at its initial value $2.5\ m$ up to $t \approx 90\ s$, which corresponds to the Froude wave travelling time over $425\ m$ with $U\_F = 5\ m/s$. At the outflow location the depth D decays with a time constant of $\approx 20\ s$ to a plateau zone of $D \approx 1.25\ m$ up to $t \approx 160\ s$. This would be the 'echo' travelling time, where D is decreasing, but both the flow velocity and thus F are increasing.

F At the outflow location F rises to ≈ 0.90 in $\approx 50\ s$ and stays in this region for the whole time window.
And at the inflow location F stays at its initial value 0.040 up to the arrival of the Froude wave at $\approx 90\ s$ and then rises slowly. Final stabilization (not shown here) will be at $t \approx 1500s$ at $F\_02 \approx 0.37$ and $F\_40 \approx 0.86$.

Subcase B, $L = 1.0\ m$

The overall emptying transient is faster here (by a factor of ≈ 25).
The general shape of the transients and the profiles are much the same. But note that the effects of the vertical dynamics on W and ΔP^* are now strong.

Q At the outflow Q_40 the peak is at $45\ m^3/s$ and the plateau at $40\ m^3/s$ up to $t \approx 7\ s$.

D Again the plateau for D is at $\approx 1.25\ m$ up to $t \approx 7\ s$.

F F_40 rises to ≈ 0.96 and stays in this region, whereas F_02 rises continuously from $t \approx 7\ s$.
Steady state is arrived at after $t = 150\ s$ with values $F\_40 \approx 0.98$ and $F\_02 \approx 0.74$.

5.7 Case 4: Surge waves

5.7.1 Introduction

The motivation and the modeling and the main data of the channel have been discussed in the introduction to this chapter. – The aim here is to show what happens in a channel during the very first seconds after a sharp rise of inflow from near zero flow to nominal flow. The time window is limited to exclude any reflections on the outflow boundary of the model channel (which is here of finite length in contrast to a real river bed of 'infinite' length).

5.7.2 Implementation

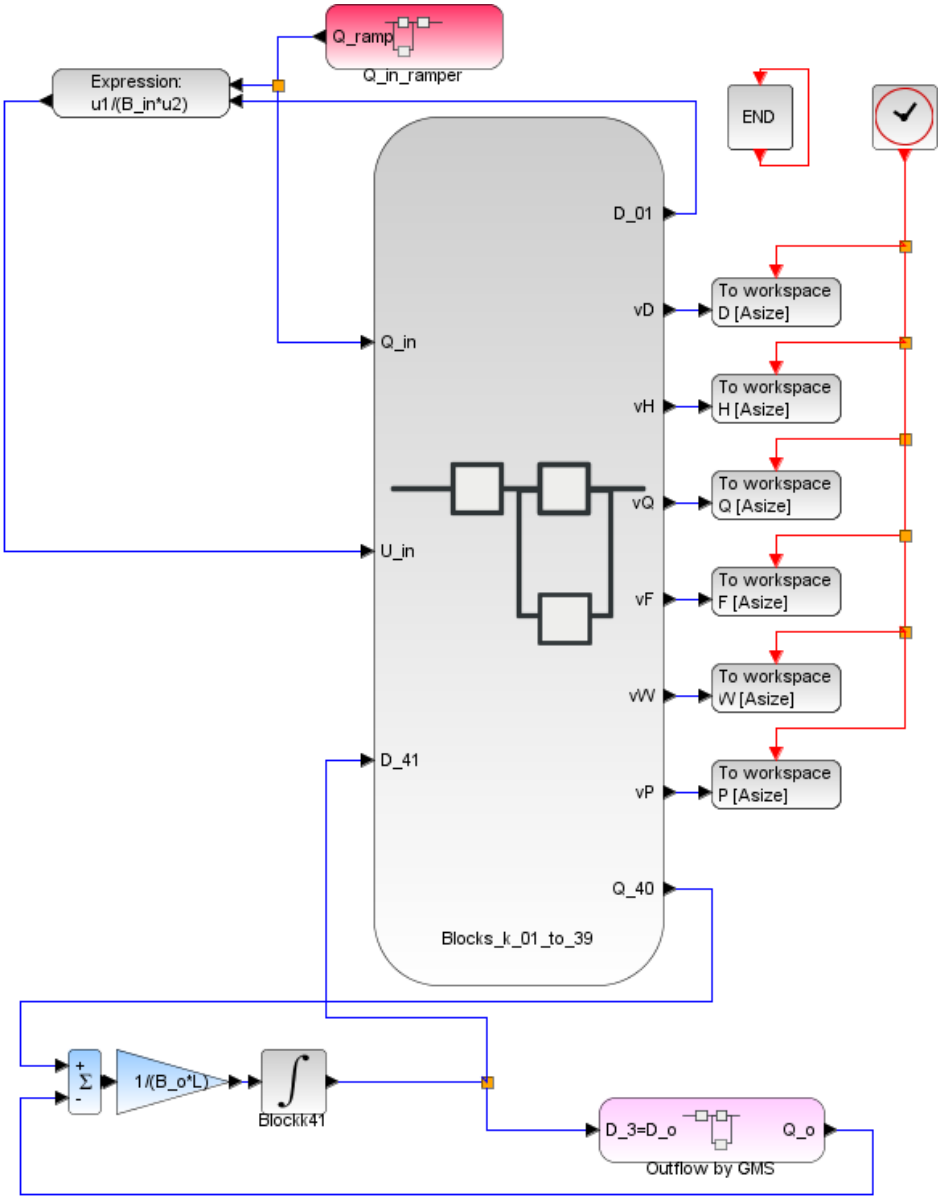


Figure 5.53: Main diagram for Case 4: Surge wave

```
// s_c5_04_context.sce
// Glf  2017_06_15
// surge wave
// by fast inflow rise
// outflow GMS

g = 10.;

Q_0 = 50.0;
H_0 = 2.5; kap = 1.0;
//******************
  lambda = 0.40;
//******************
L = lambda*H_0;
S_0 = 0.0*(-1.0);
D_0 = H_0 - S_0;
B_0 = 10.;
U_0 = 2.0;

N= 20;      // no of Vol+Momentum-Segments

q0 = 0.040; Q_I = q0*Q_0; B_I = B_0; S_I = S_0;

// GMS-friction coefficient
   k_s = 100.0;

// reference bottom slope
  Q_r = 50.0; B_r = 10.0;
  D_r =  2.5; S_r = 0*(-1.0);
  U_r =  2.0;
  R_r = (B_r*D_r)/(B_r + 2*D_r);
  I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// initialzing 'iteration loop'
D_I = D_r;
gainQ = 0.01*(D_I/Q_I);
Q = k_s*(B_I)*(D_I)*(D_I)^(2/3)*(I_r)^(1/2);
eQ  = -Q_I + Q;

// iteration loop
while (eQ<-0.0001)|(eQ>0.0001)  then
    R_I = (B_I*D_I)/(B_I + 2*D_I);
    Q   = (k_s*(I_r)^(1/2)*B_I)*D_I*(R_I)^(2/3);
    eQ  = -Q_I + Q;
    D_I = D_I - gainQ*eQ;
end

Q_I = Q;
D = D_I;

// Channel-Geometry
//*****************************
// rectangular-cross sect with constant (nominal) width
vb_1 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
      1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*(vb_1);
//*****************************

//horizontal  bottom
vs = -[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0];
vS = S_0*(vs);

 D_min = +0.001; D_max = 40*D_0;
 Q_min = +0.001; Q_max = 40*Q_0;

vd0 = ones(1,(2*N+1));
vD0 = D*vd0;
vH0 = vD0 - vS;

vq0 = ones(1,(2*N+1));
vQ0 = q0*Q_0*vq0;

vh0 = ones(1,(2*N+1));
vDdot0 = 0*vh0;
ome = 100;
P_max = 100.0; P_min = 0.0;

// Inflow-(_i) boundary cond.
    B_in = vB(1);
    S_in = vS(1);
    D_in = vD0(1); H_in = D_in + S_in;
    Q_in = vQ0(1);

// Friction-inclination of bottom
vD_E = D_0*vd0;
vQ_E = Q_0*vq0;

vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vRtilda(kk) =((vB(kk)*vD_E(kk))/...
               (vB(kk)+2*vD_E(kk)))^(2/3);
  vI_E(kk)    =(vQ_E(kk)/(vB(kk)*vD_E(kk)...
               *k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI_E(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS(k4)  + vSf(k4);
    vH0(k4) = vH0(k4) + vSf(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow-(_o) boundary cond
    B_o = vB(kk+1);
    S_o = vS(kk+1);
    D_o = vD0(kk+1); H_o = vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
t_st_1 = 0.250;  r_1_0 = q0*Q_0; r_1_1 = Q_0;
g_st = 10.0; u_up_st = +1.0; u_dn_st = -1.0;
tau_st = 0.016; Q_st_0 = q0*Q_0;

// outflow by GMS
g_o = k_s*((I_r)^(1/2));

T_fin = 3.5; CN = 700;

// Datatransfer to Plots
CC = 21;           // no of channels 20 + 1 for time
delT = T_fin/CN;   // readout-interval for clock ticks
Asize = 1.01*CC*CN; // size of data arrays
```

```
// s_c5_04_crunplot
//  Glf  2017_06_15
// surge wave

stacksize('max'); exec('s_c5_04_context.sce',-1);
importXcosDiagram('s_c5_04.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:11,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[0.,0.0,T_fin,65.]);
xtitle("Q_2 to Q_40");
xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[0.,0.0,T_fin,1.6]);
xtitle("D_1 to D_39");
xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[0.,0.0,T_fin,1.6]);
xtitle("H_1 to H_39");
xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[0.,0.0,T_fin,2.0]);
xtitle("F_2 to F_40");
xgrid(1);

f5 = scf(5);
plot2d(W.time,W.values,vcolor,...
rect=[0.,-0.2,T_fin,+1.0]);
xtitle("W_1 to W_39"); xgrid(1);

f6 = scf(6);
plot2d(P.time, (P.values - g*D.values),vcolor,...
rect=[0.,-0.50,T_fin,3.0]);
xtitle("deltaP*_1 to  deltaP*_39"); xgrid(1);
//*********************************************
//
eX = H_0*lambda*(1:1:N); eXend =22;
for k6= 1:1:N, vSr(k6) = vS(2*k6-1); end
vcolorP=[1,2,5,13];
//
vT=[0.75,1.50,2.50,3.40]; vR=[150,300,500,680];
vs = ['0','0','0','0'];

for k8 = 1:1:4,
  vs(k8) = msprintf('%5.2f',vT(k8));end

f7 = scf(7); clf(); yQ =Q.values;
iQ = yQ(vR(1),:); mQ = yQ(vR(2),:);
oQ = yQ(vR(3),:);  eQ = yQ(vR(4),:);
plot2d(eX', [iQ', mQ', oQ', eQ'], vcolorP, ...
 rect=[0.,0.0,eXend,+65.]);
vtitle=...
 ["Longit.Profile Q at",vs(1),"s (bk),",vs(2),...
  "s (bl),",vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f8 = scf(8); clf(); yD =D.values;
iD = yD(vR(1),:); mD = yD(vR(2),:);
oD = yD(vR(3),:); eD = yD(vR(4),:);
plot2d(eX', [iD', mD', oD', eD'], ...
vcolorP, rect=[0.,0.0,eXend,+1.6]);
vtitle=...
 ["Longit.Profile D at",vs(1),"s (bk),",vs(2),...
  "s (bl),",vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f9 = scf(9); clf(); yF =F.values;
iF = yF(vR(1),:); mF = yF(vR(2),:);
oF = yF(vR(3),:); eF = yF(vR(4),:);
plot2d(eX', [iF', mF', oF', eF'], vcolorP, ...
 rect=[0.,0.0,eXend,+2.0]);
vtitle=...
 ["Longit.Profile F at",vs(1),"s (bk),",vs(2),...
  "s (bl),",vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f10 = scf(10); clf(); yW =W.values;
iW = yW(vR(1),:); mW = yW(vR(2),:);
oW = yW(vR(3),:); eW = yW(vR(4),:);
plot2d(eX', [iW', mW', oW', eW'], vcolorP, ...
rect=[0.,-0.20,eXend,+1.0]);
vtitle=...
 ["Longit.Profile W at",vs(1),"s (bk),",vs(2),...
  "s (bl),", vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);

f11 = scf(11); clf(); yP =P.values;
vP=...
[yP(vR(1),:)'-g*yD(vR(1),:)',yP(vR(2),:)'-g*yD(vR(2),:)',...
yP(vR(3),:)'-g*yD(vR(3),:)',yP(vR(4),:)'-g*yD(vR(4),:)'];
plot2d(eX',vP, vcolorP, rect=[0.,-0.5,eXend,+3.0]);
vtitle=["Longit.Profile (P*-gD) at",vs(1),"s (bk),", ...
  vs(2),"s (bl),",vs(3),"s (rd), and",vs(4),"s (gn)"];
xtitle(vtitle); xgrid(1);
```

Again the transients and longitudinal profiles for each variable are paired on one page.

5.7.3 Simulation Results

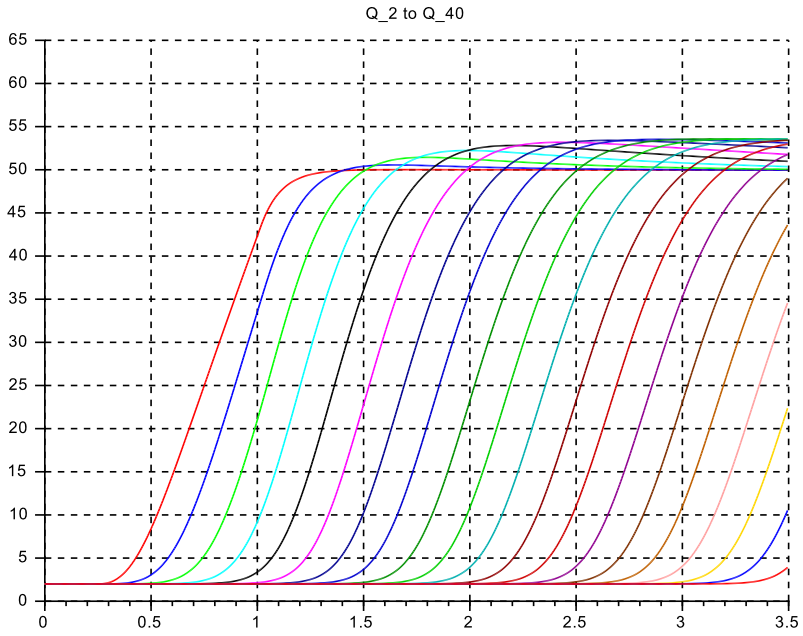


Figure 5.54: C 4: Transient $Q(t)$ for $L = 1.0$ $[m]$

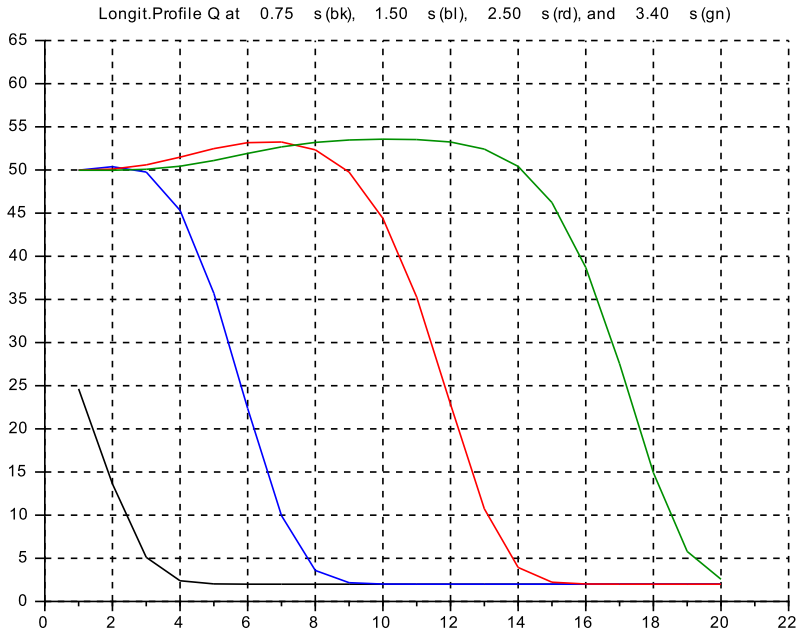


Figure 5.55: C 4: Longit.Profile $Q(x)$ for $L = 1.0$ $[m]$

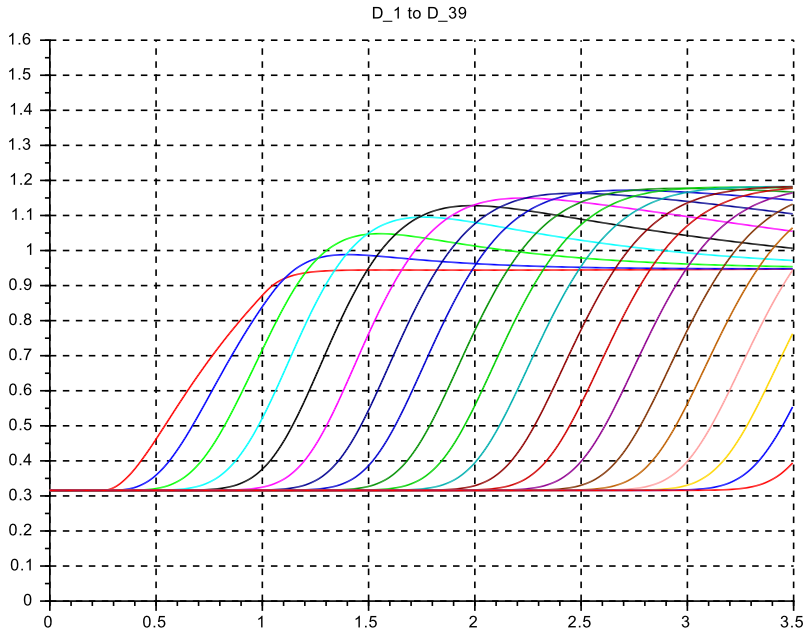


Figure 5.56: C 4: Transient $D(t)$ for $L = 1.0$ $[m]$

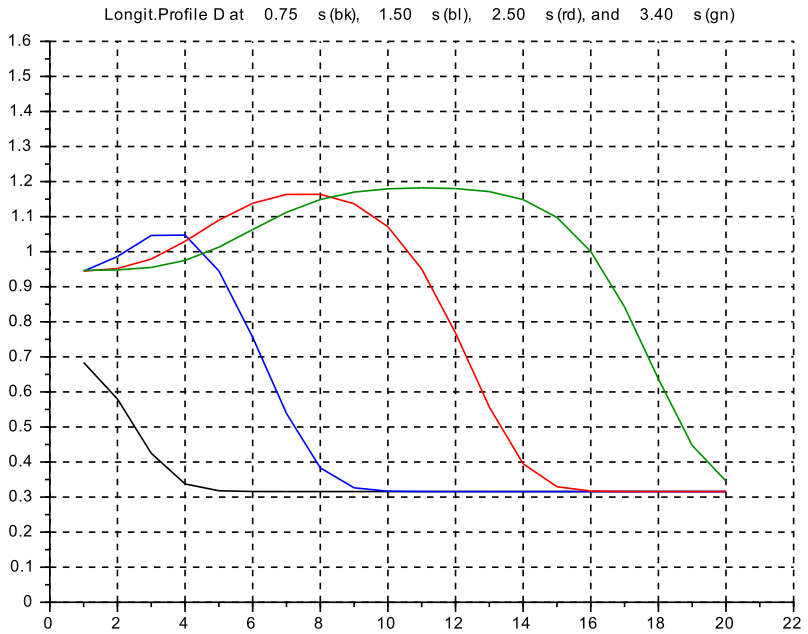


Figure 5.57: C 4: Longit.Profile $D(x)$ for $L = 1.0$ $[m]$

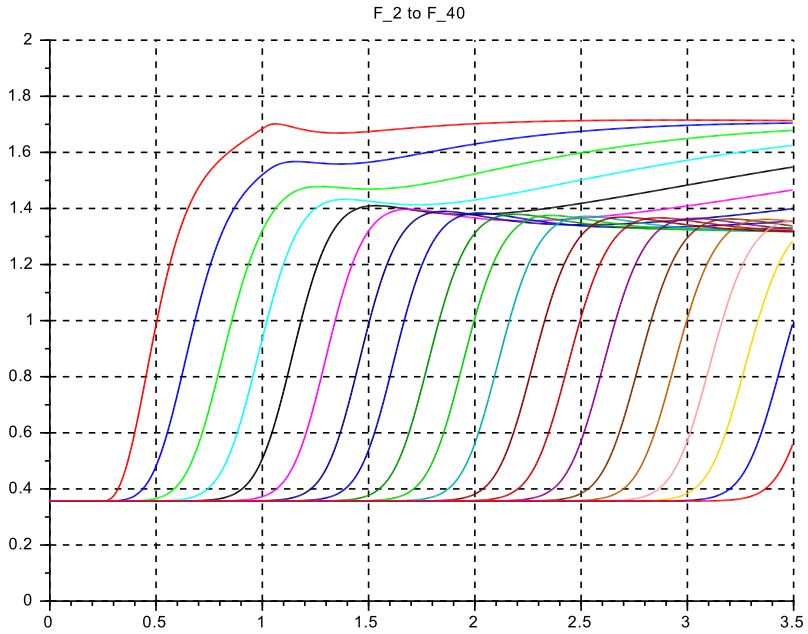


Figure 5.58: C 4: Transient $F(t)$ for $L = 1.0$ $[m]$

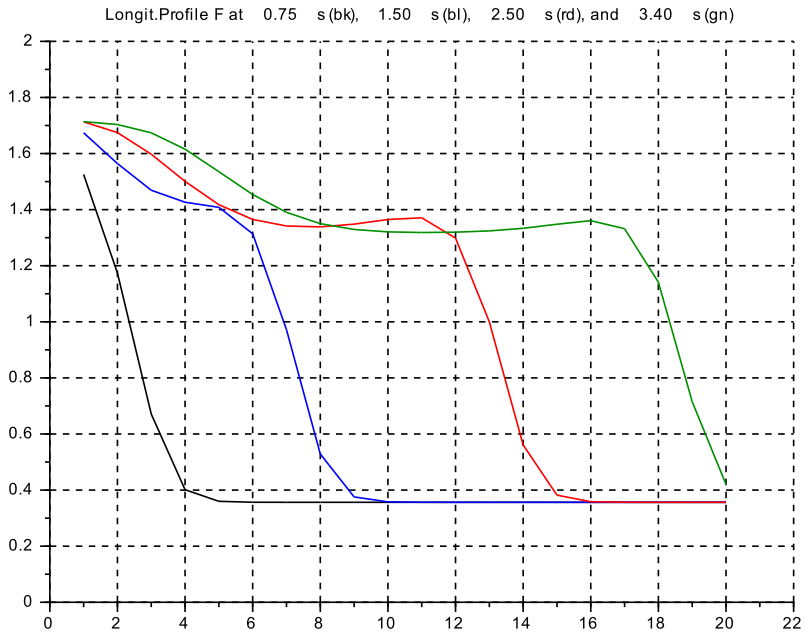


Figure 5.59: C 4: Longit.Profile $F(x)$ for $L = 1.0$ $[m]$

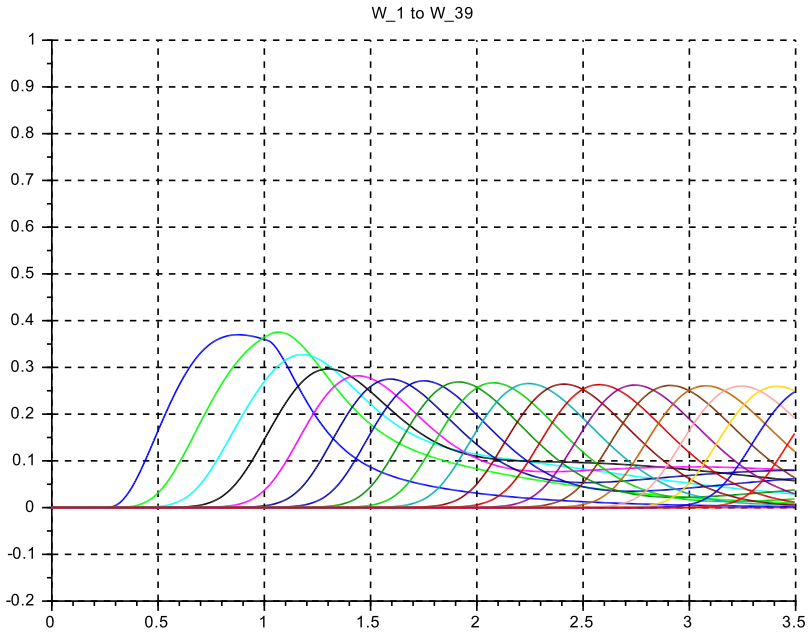


Figure 5.60: C 4: Transient $W(t)$ for $L = 1.0$ $[m]$

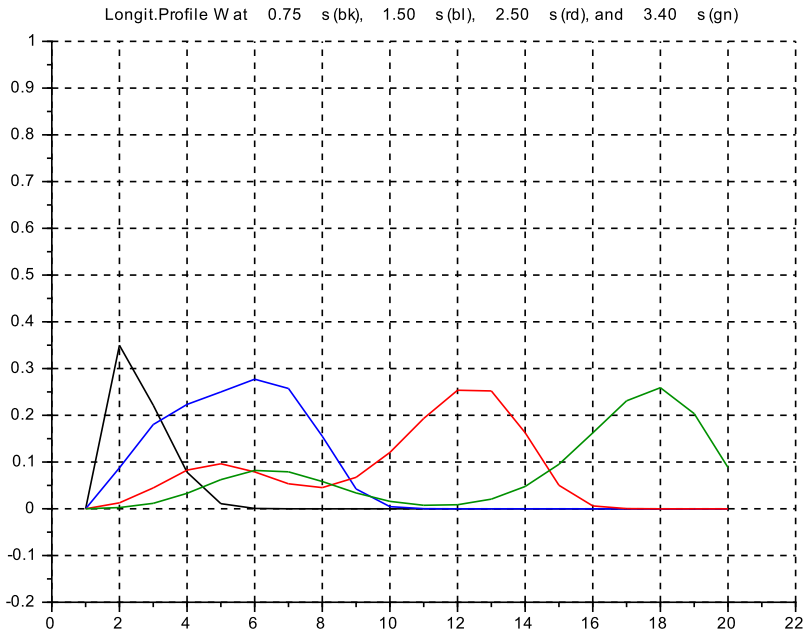


Figure 5.61: C 4: Longit.Profile $W(x)$ for $L = 1.0$ $[m]$

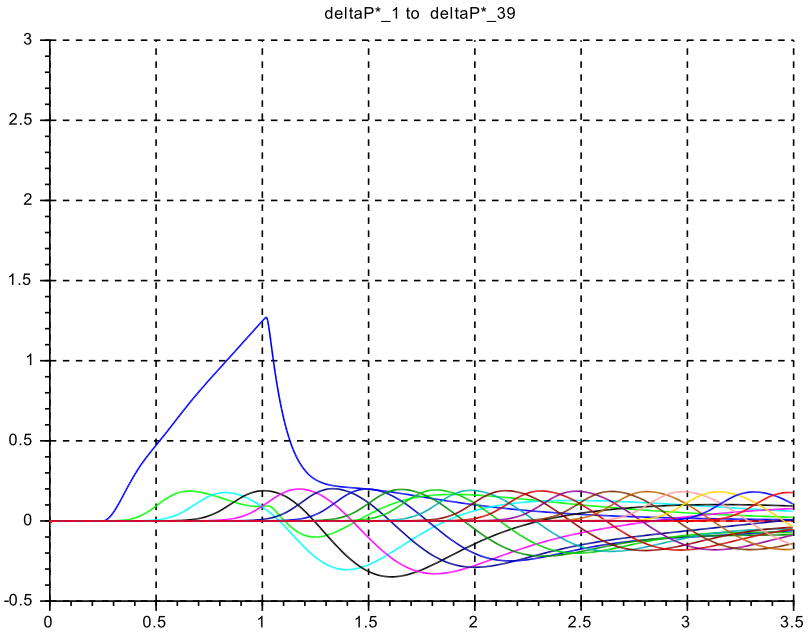


Figure 5.62: C 4: Transient $\Delta P^*(x)$ for $L = 1.0$ $[m]$

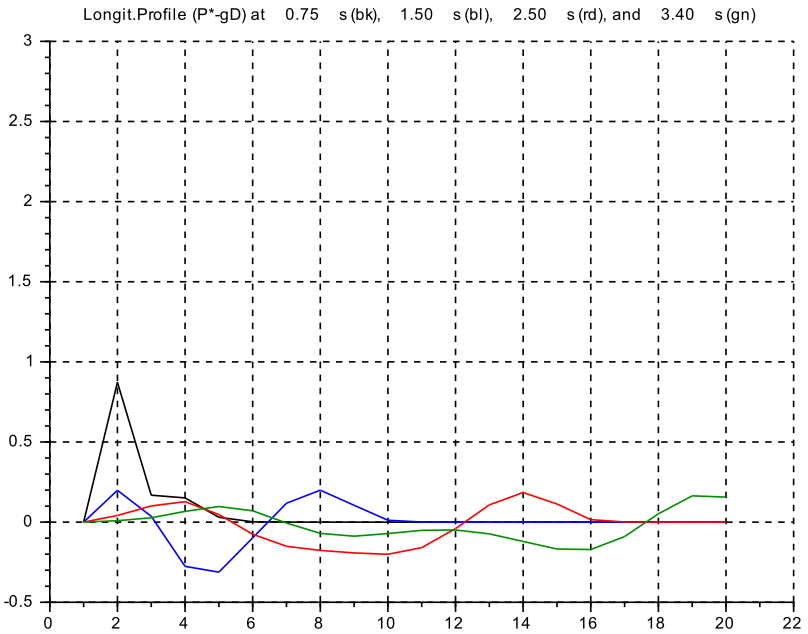


Figure 5.63: C 4: Longit.Profile $\Delta P^*(x)$ for $L = 1.0$ $[m]$

5.7.4 Discussion for Case 4

The friction slope is assumed to be the nominal one from above, with $Q := 50\ m^3/s,\ D := 2.5\ m, B := 10\ m, k\_s := 100$, that is $I\_r = 0.2024 \cdot 10^{-3}$.

Q The initial flow is set to $Q\_0 := 2.0\ m^3/s$. This yields $D\_0 := 0.316\ m, F\_0 := 0.356$.
Then the inflow is raised to $50\ m^3/s$ in $\Delta t = 0.80\ s$. No filter is applied to this ramp-up.
Note that Q_10 and further downstream values overshoot to $\approx 54\ m^3/s$ and then slowly fall back to $50\ m^3/s$.
This transient peaking shows up in other variables as well.

D At the inflow location and $50\ m^3/s$ the depth stabilizes at $D\_03 \approx 0.95\ m$. Further downstream D overshoots to $\approx 1.20\ m$ and slowly drops back to $0.95\ m$.

F At the inflow, the Froude number F_02 rises in less than $1\ s$ to ≈ 1.7 and remains in the region. Further downstream, for instance at location $\_10$ and following, there is a plateau in F at ≈ 1.40, from where the local F's rise further to 1.70. The plateau is also clearly visible on the profile $F(x)$.

W and

ΔP^* The profiles show a distinct wave travelling downstream as with Q, D.

To summarize: there is an overshoot in Q and D travelling downstream close after the wave front (the delay is $\approx 1...2\ s$). Also the flow is accelerated to supercritical (up to $f \approx 1.7$) in this short time interval, respectively over a distance of a few m. This is due to the steep initial wave front, which builds up a strong hydrostatic pressure differential.
The transient responses and the profiles correlate reasonably well with observations of such surge waves. However the flow observed in reality is highly turbulent which cannot be modeled with the basic modeling approach used here...

5.8 Case 5: Hydraulic Jump after a Weir

In the next sections two cases of 'Hydraulic Jump' shall be investigated. The first case is 'after a weir', where water flows *below* the shutoff device and then meets a small flow discontinuity.
The second case will be 'after a dam', where water flows over its crown, accelerates on the backside slope of the dam, and finally hits the horizontal bottom.

5.8.1 Modeling and Data set

Another 'Basic Result'

The aim is to find a basic result similar as for the spillway case for checking the simulation result.
The jump position is considered stationary on x. Also the flow shall be stationary. And no information about the spacial shape of the hydraulic jump is asked for.
Denoting the locations far upstream of the jump with index $_{.1}$ and far downstream with $_{.2}$ then, with $B_1 = B_2 = B$

$$\begin{aligned}
&\text{for the volume balance} && B \cdot U_1 \cdot D_1 &&= && B \cdot U_2 \cdot D_2 \\
&\text{for the momentum balance} && \rho B \left[\frac{1}{2}\, g\, D_1^2 + D_1 \cdot U_1^2\right] &&= && \rho B \left[\frac{1}{2}\, g\, D_2^2 + D_2 \cdot U_2^2\right]
\end{aligned}$$

Rearranging

$$\begin{aligned}
D_1 U_1^2 - D_2 U_2^2 &= \frac{1}{2} g \left[D_2^2 - D_1^2\right] \\
D_1 U_1 \left[U_1 - U_2\right] &= g \frac{1}{2} \left[D_2 + D_1\right] \cdot \left[D_2 - D_1\right] \\
\text{with} \qquad D_m &= \frac{1}{2} \left[D_2 + D_1\right] \quad \text{and} \quad D_2 = \frac{U_1 D_1}{U_2} \\
D_1 U_1 \left[U_1 - U_2\right] &= g D_m \frac{D_1}{U_2} \cdot \left[U_1 - U_2\right]
\end{aligned}$$

$$\begin{aligned}
\text{finally} \qquad U_1 \cdot U_2 &= g D_m \quad = \quad U_{F_m}^2 \\
\text{or} \qquad \frac{U_1}{\sqrt{g D_m}} \cdot \frac{U_2}{\sqrt{g D_m}} &= 1.0 \\
\text{that is} \qquad F_{1_m} \cdot F_{2_m} &= 1.0
\end{aligned}$$

Note that the Froude numbers are not those at $_{.1}$, $_{.2}$ locations (as displayed in all graphs so far) but taken with a 'synthetic' Froude velocity $U_{F_M} = \sqrt{g D_m}$!

Experiment setup

Initially the flow down the channel is constant/nominal $Q = 50\ m^3/s$, in inflow/outflow equilibrium and supercritical at $F = 2.0$. Then the inflow is ramped up to $55\ m^3/s$ in $1\ s$. This will generate a wave travelling downstream with flow velocity plus Froude velocity. But the outflow is kept fixed at the initial value, for instance by inserting a weir board. This will cause an echo wave travelling slowly upstream. When this echo wave is about halfway upstream, the inflow is ramped back to its initial value, restoring flow equilibrium again. The aim is to show that a stationary hydraulic jump is produced, what its longitudinal profile is, and whether is complies to the basic result from above.
This simple clinical experiment may seem somewhat artificial, as in real situations such hydraulic jumps are generated mostly by discontinuities of the bottom (or sidewalls), see the following case.

Data set

The channel geometry is assumed at constant cross section $B = 10\ m$.
For $Q_0 := 50\ m^3/s, F_0 := 2.0$ results $D_0 = 0.855\ m, U_0 = 5.848\ m/s$ with $U_F = 2.924\ m/s$.
The bottom slope is calculated using the GMS-factor $k_s = 1000$. This amounts to very small friction and requires a very smooth surface. Note that the basic result from above assumes no friction loss at all. Letting $k_s = 100$ (as in cases above) already has a visible effect on the simulation results (not shown here).
Further the compartment length is set to $L = 1.0\ m$ as in the previous cases.

5.8.2 Implementation

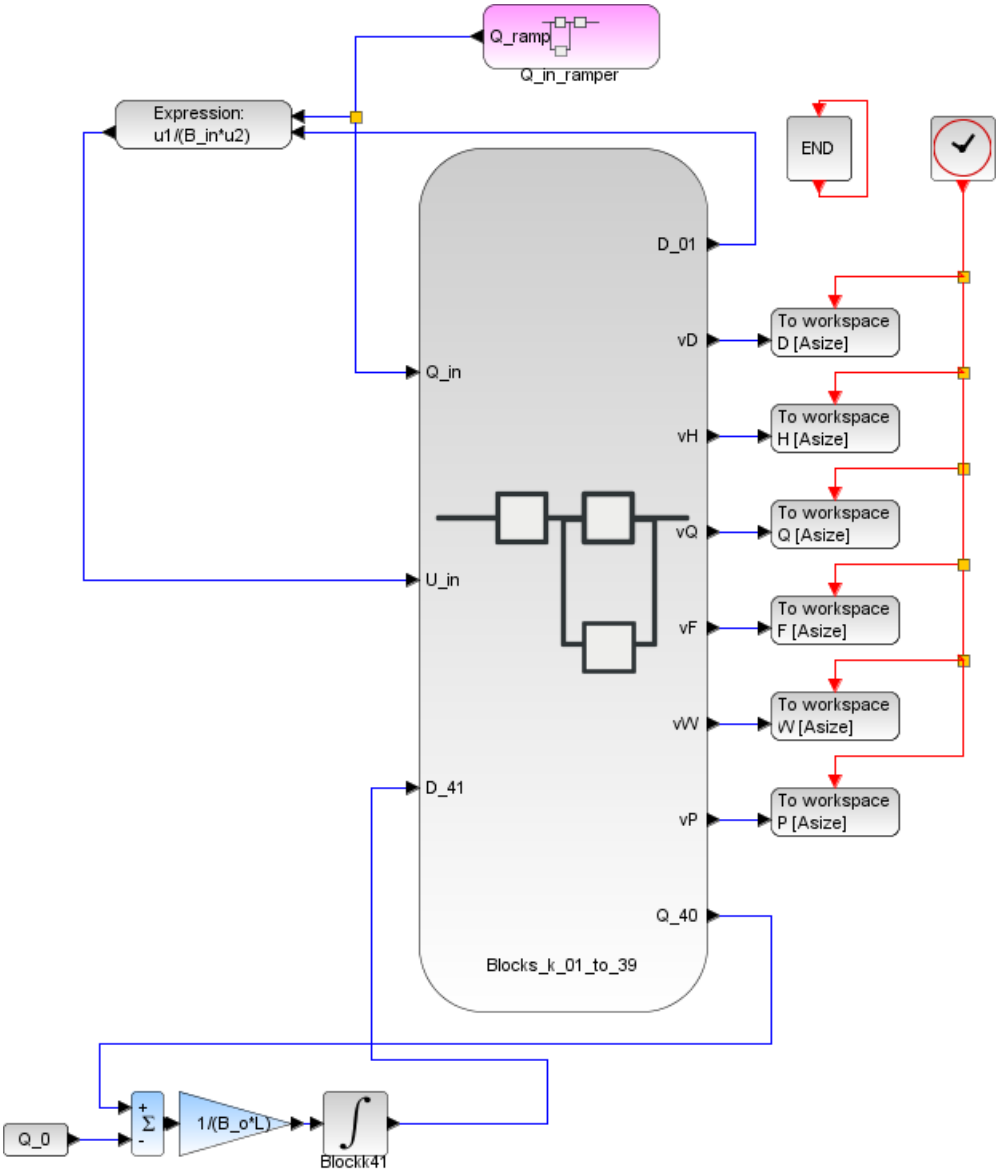


Figure 5.64: Main diagram for Case 5: Hydraulic Jump after a Weir

```
// s_c5_05_context.sce
// Glf  2017_06_15
// hydraulic jump on flat bottom
// by fast inflow rise
// outflow fixed at Q_0

g = 10.;
Q_0 = 50.0; H_0 = 2.5; kap = 1.0;
//******************
  lambda = 0.400;
//******************
L = lambda*H_0;
S_0 = 0.0*(-1.0); D_0 = H_0 - S_0;
B_0 = 10.; U_0 = 2.0;
N= 20;      // no of Vol+Momentum-Segments

//Given: Q_0 and Fr_0; determine D_0
 Fr_0 = 2.0;
 D_0 = ((1/g)*((Q_0/(B_0*Fr_0))^2))^(1/3);

// GMS-friction coefficient
   k_s = 1000;
// reference bottom slope
  Q_r = 50.0; B_r = 10.0;
  D_r =  2.5; S_r = 0*(-1.0);  U_r =  2.0;
  R_r = (B_r*D_r)/(B_r + 2*D_r);
  I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// Channel-Geometry
//*****************************
// rectangular-cross sect with constant (nominal) width
vb_1 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
        1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_0*(vb_1);
//*****************************

//horizontal  bottom
vs = -[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,...
       0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0];
vS0 = S_0*(vs);

 D_min = +0.001; D_max = 40*D_0;
 Q_min = +0.001; Q_max = 40*Q_0;

vd0 = ones(1,(2*N+1));
vD0 = D_0*vd0; vD_r = D_r*vd0;

vH0 = vD0 - vS0;
q0  = 1.00;
vq0 = ones(1,(2*N+1));
vQ0 = q0*Q_0*vq0;

vh0 = ones(1,(2*N+1));
vDdot0 = 0*vh0; ome = 100;
P_max = 100.0; P_min = 0.0;

// Inflow-(_i) boundary cond.
    B_in = vB(1);  S_in = vS0(1);
    D_in = vD0(1); H_in = D_in + S_in;
    Q_in = vQ0(1);  W_in = 0.;

// Friction-inclination of bottom
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vRtilda(kk) = ((vB(kk)*vD0(kk))/ ...
     (vB(kk)+2*vD0(kk)))^(2/3);
  vI0(kk) = (vQ0(kk)/(vB(kk)*vD0(kk)...
  *k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4) + vSf(k4);
    vH0(k4) = vH0(k4) + vSf(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow-(_o) boundary cond
  B_o = vB(kk+1);  S_o = vS(kk+1);
  D_o = vD0(kk+1); H_o = vH0(kk+1); Q_o = vQ0(kk+1);

// inflow generation
t_st_1 =  10.0;  r_1_0 = Q_0; r_1_1 = (1.0+0.1)*Q_0;
t_st_2 =  40.0;  r_2_0 = 0.;  r_2_1 = -0.10*Q_0;
// inflow slew rate
g_st = 100.0; u_up_st = +1.0; u_dn_st = -1.0;
tau_st = 0.20; Q_st_0 = Q_0;

// outflow by GMS
// g_o = k_s*((I_r)^(1/2));

T_fin  = 100.0; CN = 1000;

// Datatransfer to Plots
CC = 21;           // no of channels 20 + 1 for time
delT = T_fin/CN;   // readout-interval for clock ticks
Asize = 1.01*CC*CN; // size of data arrays
```

```
// s_c5_05_runplot
//  Glf  2017_06_15
// Hydraulic jump on flat bottom

stacksize('max');
exec('s_c5_05_context.sce', -1);
importXcosDiagram('s_c5_05.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:11,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[0.,0.0,T_fin,60.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[0.,0.0,T_fin,3.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[0.,0.0,T_fin,3.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[0.,0.0,T_fin,3.0]);
xtitle("F_2 to F_40"); xgrid(1);

f5 = scf(5);
plot2d(W.time,W.values,vcolor,rect=[0.,-0.4,T_fin,1.4]);
xtitle("W_1 to W_39"); xgrid(1);

f6 = scf(6);
plot2d(P.time, (P.values-g*D.values), vcolor,...
       rect=[0.,-5.0,T_fin,5.0]);
xtitle("deltaP*_1 to  deltaP*_39"); xgrid(1);
//*****************************************
eX = H_0*lambda*(1:1:N); eXend = 22;
for k6= 1:1:N,
  vSr(k6) = vS(2*k6-1);
end
vT = [t_st_1-5.0, t_st_2, T_fin-5.0];
vR = [50, 390, 950]; vs = ['0','0','0'];
for k8 = 1:1:3,vs(k8)=msprintf('%5.1f',vT(k8));end
vcolorP=[13,2,5];

f7 = scf(7); clf(); yQ =Q.values;
vQ=[yQ(vR(1),:)',yQ(vR(2),:)',yQ(vR(3),:)'];
plot2d(eX',vQ,vcolorP,rect=[0.,0.0,eXend,+60.]);
vtitle = ["Longit.Profile Q at",vs(1),"s (bk),",...
          vs(2),"s (bl), and",vs(3),"s (rd)"];
xtitle(vtitle); xgrid(1);
```

```
f8 = scf(8); clf(); yD =D.values;
vD=[yD(vR(1),:)',yD(vR(2),:)',yD(vR(3),:)'];
plot2d(eX',vD,vcolorP,rect=[0.,0.0,eXend,+3.0]);
vtitle = ["Longit.Profile D at",vs(1),"s (gn),",...
          vs(2),"s (bl), and",vs(3),"s (rd)"];
xtitle(vtitle); xgrid(1);

f9 = scf(9); clf(); yF =F.values;
vF=[yF(vR(1),:)',yF(vR(2),:)',yF(vR(3),:)'];
plot2d(eX',vF,vcolorP,rect=[0.,0.0,eXend,+3.0]);
vtitle = ["Longit.Profile F at",vs(1),"s (gn),",...
         vs(2),"s (bl), and",vs(3),"s (rd)"];
xtitle(vtitle); xgrid(1);
```

```
f10 = scf(10); clf(); yW =W.values;
vW=[yW(vR(1),:)',yW(vR(2),:)',yW(vR(3),:)'];
plot2d(eX',vW, vcolorP, rect=[0.,-0.40,eXend,+1.6]);
vtitle = ["Longit.Profile W at",vs(1),"s (gn),",...
         vs(2),"s (bl), and",vs(3),"s (rd) "];
xtitle(vtitle); xgrid(1);

f11 = scf(11); clf(); yP =P.values;
vPP=[yP(vR(1),:)'-g*yD(vR(1),:)',...
yP(vR(2),:)'-g*yD(vR(2),:)',yP(vR(3),:)'-g*yD(vR(3),:)'];
plot2d(eX',vPP, vcolorP, rect=[0.,-5.0,eXend,+5.0]);
vtitle =["Longit.Profile P* - gD at",vs(1),"s (gn),",...
         vs(2),"s (bl), and",vs(3),"s (rd) "];
xtitle(vtitle); xgrid(1);
```

5.8.3 Discussion for Case 5

- The simulations show a strong hydraulic jump both in D and in F for flow being equal before and after the flow disturbance. And the jump is stationary.
  In other words the supercritical flow regime is an unstable equilibrium, and the supercritical/subcritical ('jump') regime represents a stable equilibrium. – With more friction loss the outflow depths tend to drift upward (experiments not shown here)

- There is a substantial overshoot in both transient $vD(t)$ and profile $vD(x)$ and a corresponding undershoot in $vF(t), vF(x)$. This corresponds with observation in real situations.

- A rule of thumb from observations is that the length of the jump zone (in profile $D(x)$ is approx 4...6 times the stationary after-jump depth. This would be 8...12 m for this case. From the simulations the jump zone length is taken as $\approx 8\ m$. The difference may be due to the heavy turbulence observed in the real situations. Note that the steepest flank in the simulations is about 2 m long for 1 m of depth difference, which may well cause a back-rollover (see remarks in the previous chapter on locally upstream flow)...

- The downstream wave velocity would be $U_1 + U_{F_1} \approx 8.8\ m/s$, whilst the upstream wave velocity of the jump peak is about $0.36\ m/s$ (taken from the transient $vD(t)$). It is very slow compared to the downstream propagation.

- Checking the 'basic result' from this section: only the red profile from Fig.5.68 is relevant, as the blue one is at non-stationary flow, see Fig.5.66. Reading from the profile

$$
\begin{aligned}
& & D_1 &:= 0.855\ m; \quad D_2 := 2.033\ m \ \rightarrow\ D_m = 1.444\ m; \ \rightarrow\ U_{F_m} = 3.80\ m/s \\
\text{with} & & \frac{Q}{B} &= 5.0\ m^2/s \quad \text{and} \quad D_1, D_2 \quad \text{from above} \\
\text{yields} & & U_1 &= 5.848\ m/s; \quad U_2 = 2.459\ m/s \\
\text{and with } U_{F_m} & & F_{1_m} &= 1.539 \quad \text{and} \quad F_{2_m} = 0.647 \\
\\
\text{finally} & & F_{1_m} \cdot F_{2_m} &= 1.539 \cdot 0.647 \ = \ 0.996 \ \ (!)
\end{aligned}
$$

- Also note from the transients and profiles for W and ΔP^* that the linear interpolation (standard for scilab plots) produces quite ragged curves. This indicates that the entry of $L := 1.0\ m$ is at the upper limit. It should be shortened to $L = 0.50\ m$ at least, but that would require a longer model $N = 20 \rightarrow 40$ to maintain sufficient windows in time and space. In other words the limit of the current compartment model is encountered again.

5.8.4 Simulation Results

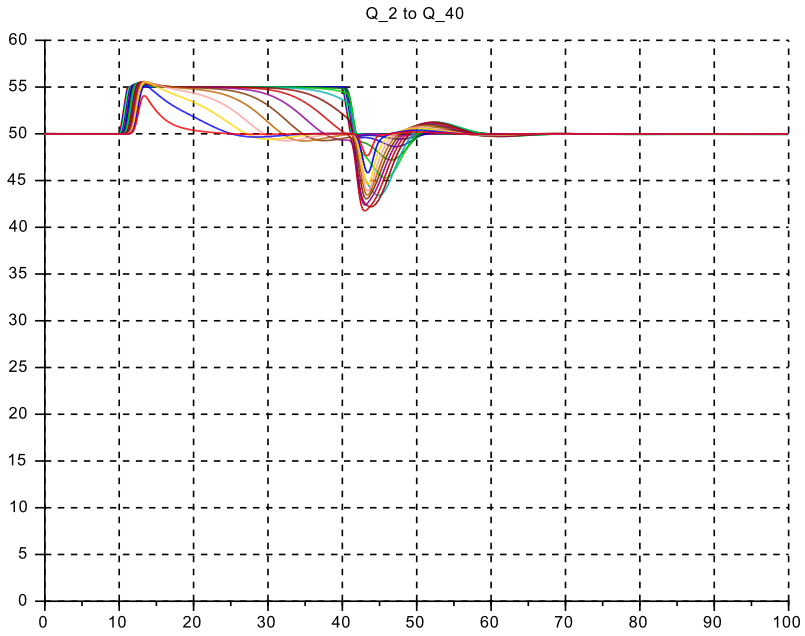


Figure 5.65: C 5: Transient $Q(t)$ for $L = 1.0$ $[m]$

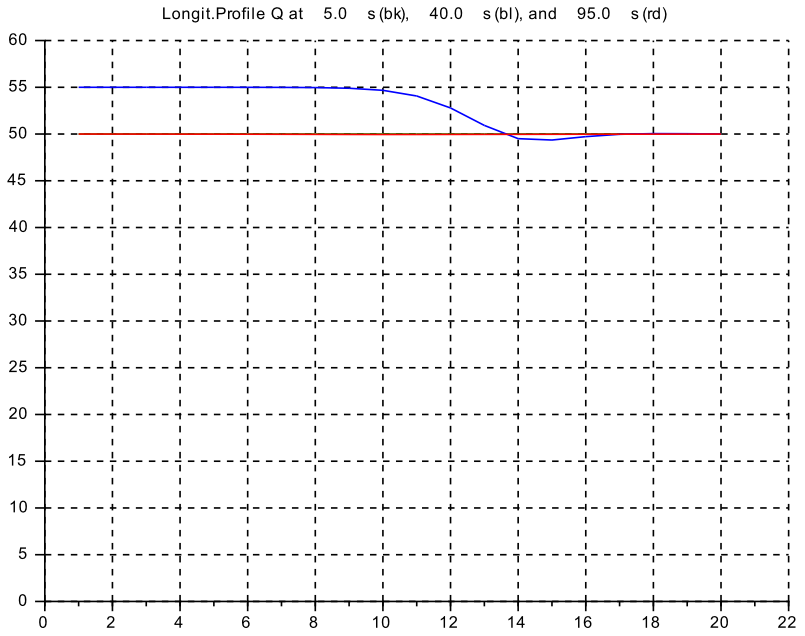


Figure 5.66: C 5: Longit.Profile $Q(x)$ for $L = 1.0$ $[m]$

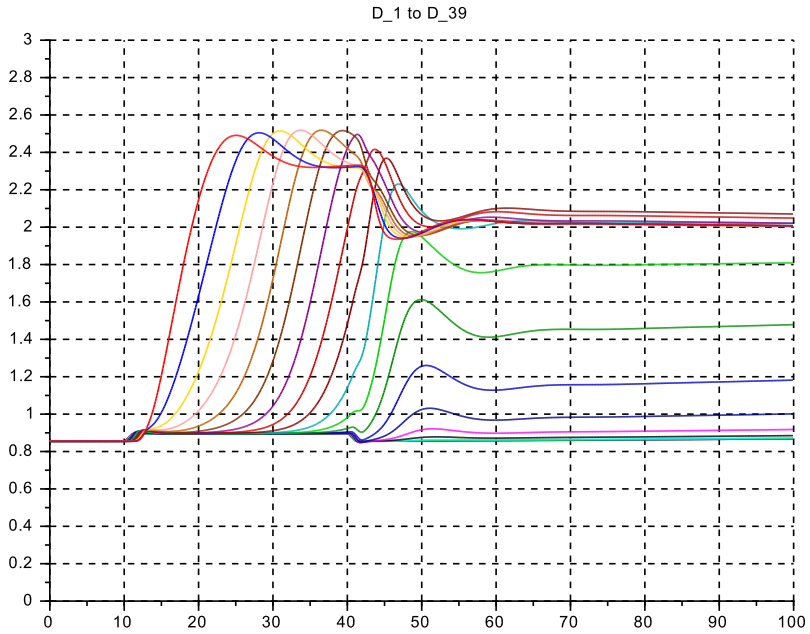


Figure 5.67: C 5: Transient $D(t)$ for $L = 1.0$ $[m]$

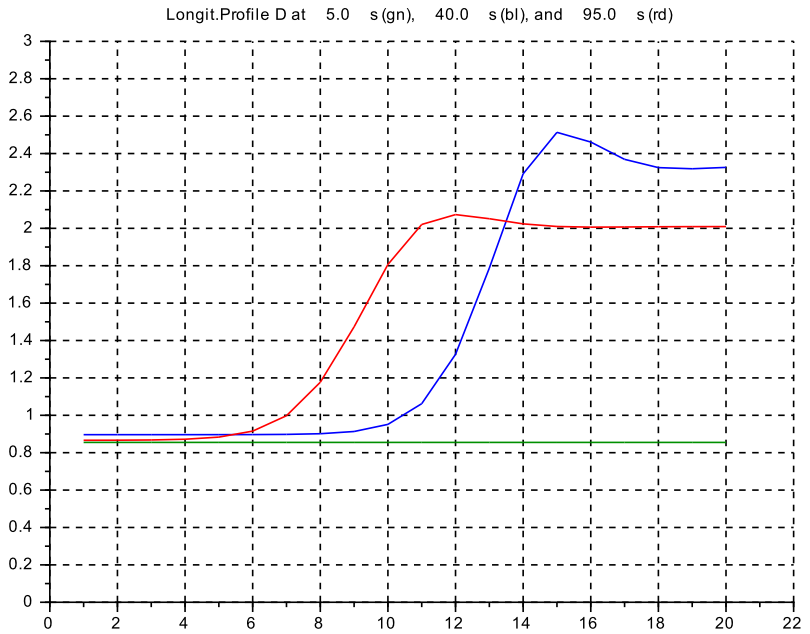


Figure 5.68: C 5: Longit.Profile $D(x)$ for $L = 1.0$ $[m]$

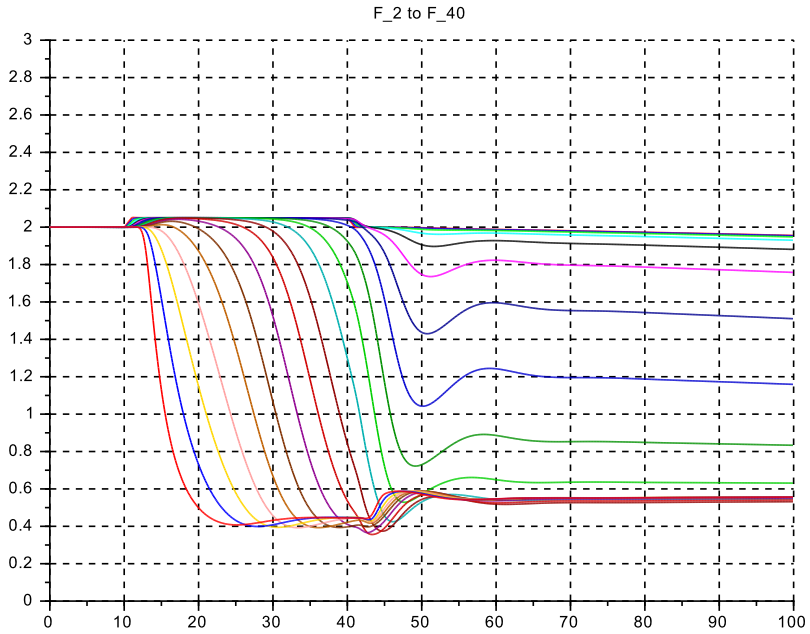


Figure 5.69: C 5: Transient $F(t)$ for $L = 1.0$ $[m]$

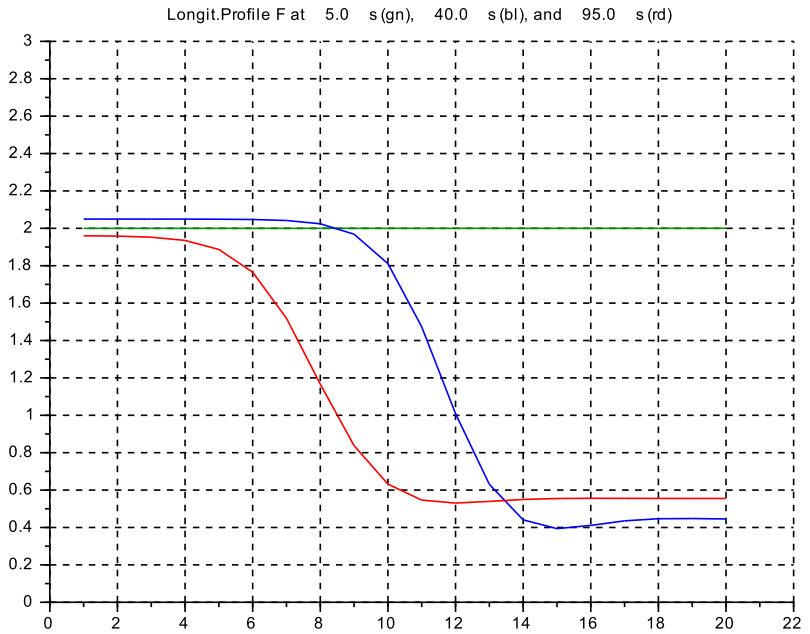


Figure 5.70: C 5: Longit.Profile $F(x)$ for $L = 1.0$ $[m]$

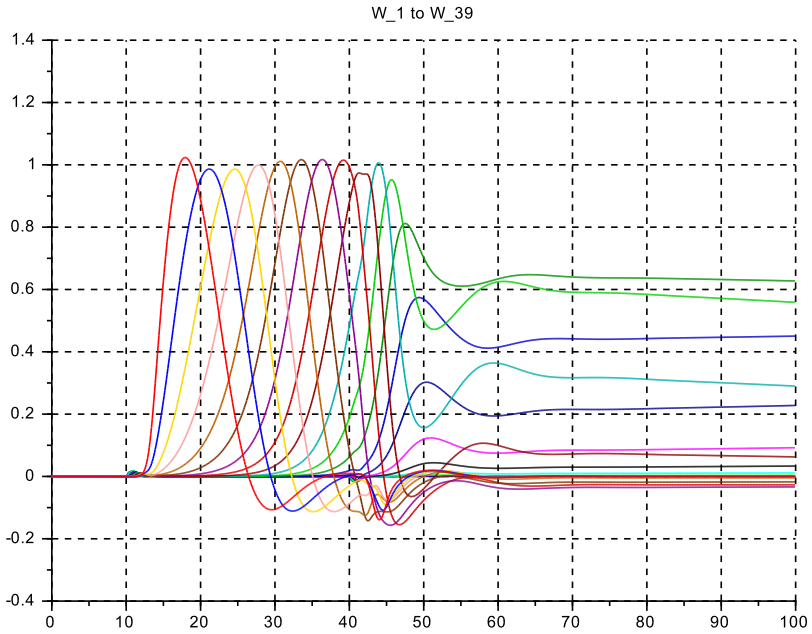


Figure 5.71: C 5: Transient $W(t)$ for $L = 1.0$ $[m]$

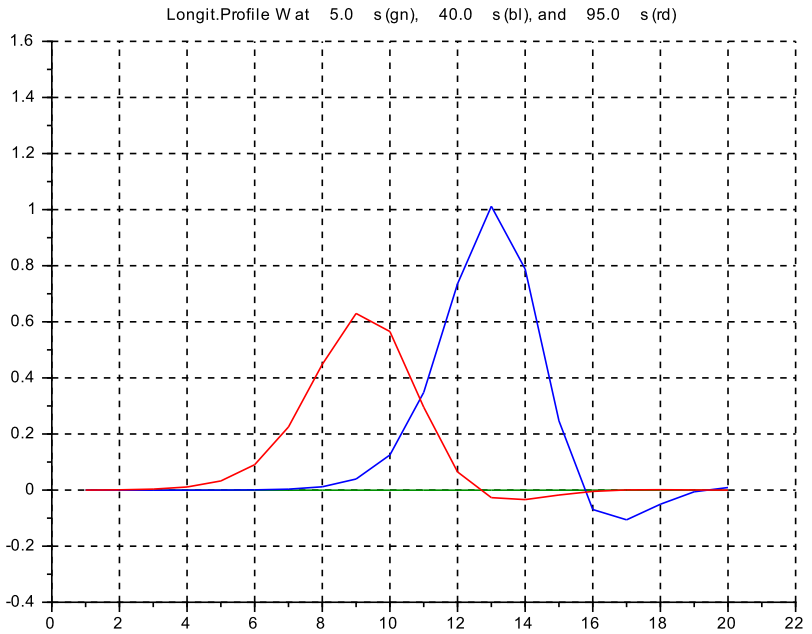


Figure 5.72: C 5: Longit.Profile $W(x)$ for $L = 1.0$ $[m]$

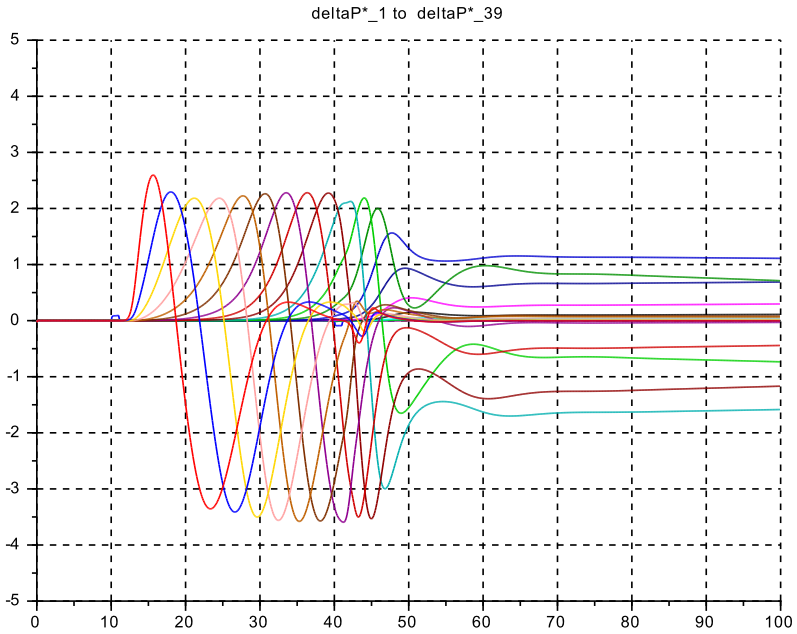


Figure 5.73: C 5: Transient $\Delta P^*(x)$ for $L = 1.0$ $[m]$

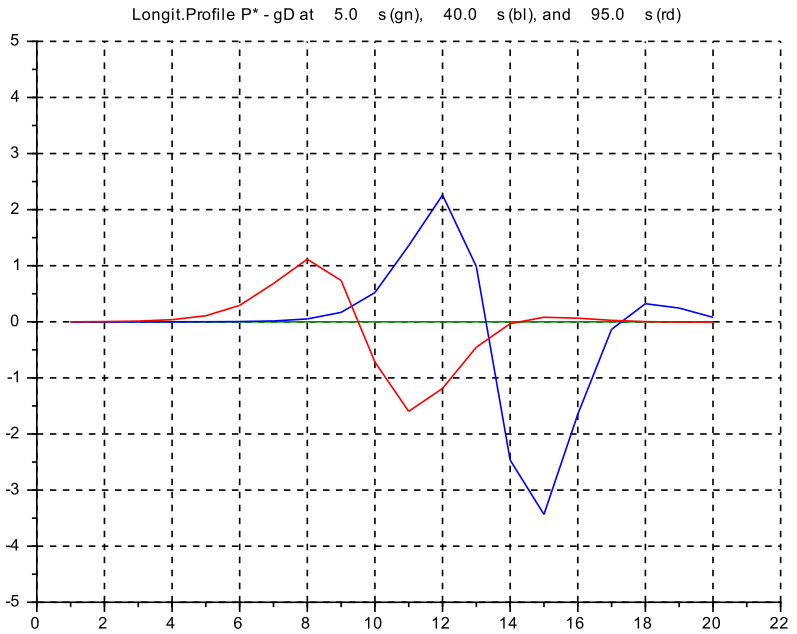


Figure 5.74: C 5: Longit.Profile $\Delta P^*(x)$ for $L = 1.0$ $[m]$

5.9 Case 6: Hydraulic Jump after a Dam

5.9.1 Modeling and Data Set

The bottom shall ramp down from position $x = 2\ m$ to position $x = 8\ m$ nominally from $S = 0.0\ m$ down to $S = -0.45\ m$, that is with a nominal slope of $-75.0 \cdot 10^{-3}$. For $0 \geq x < 2$ and beyond for $8.0 \geq x \leq L_{tot}$ the bottom shall be nominally horizontal (slope zero). To this the friction slope is added, using as reference values (see cases above): $Q_r = 50\ m^3/s;\ D_r = 2.5\ m : B_r = 10\ m; U_r = 2\ m/s$. The GMS-coefficient is set to $k_s := 70$, which is for this particular situation a more realistic value than $k_s = 1000$, as used in Case 5.
The nominal compartment length is set to $L := 1.0\ m$ as above. In order to check the influence of this choice on the shape and position of the hydraulic jump, two values of shorter L will be applied, that is $L = 0.50\ m$ and $L = 0.667\ m$. However the shape of the bottom must remain the same, at least as a good approximation. This is achieved by applying a similar algorithm as with the spillway confusor contour (see the `s_c5_06_context.sce`-listing below).

The inflow is starting at very low flow $Q = 2\ m^3/s$. Then flow is raised in a sequence: $Q \rightarrow 10.0, \rightarrow 20.0, \rightarrow 40.0\ m^3/s$. Each step (rather a fast ramp) is applied after steady state conditions are attained.

The outflow is set to the (nominal) GMS-condition from above.

Further a second value of the GMS-coefficient $k_s := 40$ is applied to check the effect on transients and profiles. This produces a friction effect approx. three times stronger than for the nominal case $k_s = 70$.

5.9.2 Implementation

```
// s_c5_06_context.sce
// Glf  2017_06_15
// hydraulic jump on inclined & flat bottom
// by fast inflow rise
// outflow by GMS

g = 10.;
Q_0 = 50.0;  H_0 = 2.5;  kap = 1.0;
lambda0 = 0.40;
//******************
   lambda = (1.00)*lambda0;
// lambda = (0.6667)*lambda0;
// lambda = (0.500)*lambda0;
//******************
L = lambda*H_0; S_0 = 0.0;  D_0 = H_0 - S_0;
B_0 = 10.; U_0 = 2.0;

N= 20;      // no of Vol+Momentum-Segments

//Given: Q_0 and F_0; determine D_0
 F_0 = 1.0;
 D_0 = ((1/g)*((Q_0/(B_0*F_0))^2))^(1/3);

// GMS-friction coefficient, usual frict.
   k_s = 70.;
// k_s = 40.;

// reference bottom slope
  Q_r=50.0; B_r=10.0;  D_r=2.5; S_r=0.0; U_r=2.0;
  R_r = (B_r*D_r)/(B_r + 2*D_r);
  I_r = (U_r/(k_s*(R_r)^(2/3)))^2;

// channel inlet geometry
//*************************************
vb_1 = ones(1,((2*N)+1)); vb_2 = zeros(1,((2*N)+1));
vs_1 = vb_1; vs_2 = vb_2;

if lambda == (1.00)*lambda0 then
ik_E = 4; ik_R = 12;  delta_b = 0.1000;
elseif lambda == 0.6667*lambda0 then
ik_E = 6; ik_R = 18;  delta_b = 0.06667;
else
ik_E = 8; ik_R = 24; delta_b = 0.0500; end

ik_T = ik_E + ik_R;
vib = zeros(1,ik_T);
vd_b = zeros(1,(ik_T));

for ii = 1:ik_R
 vd_b(ii) =  delta_b; end
vincr_b  = vib + vd_b;

for ij = 1:(ik_T)
  vib(ij+1) = vib(ij) + vincr_b(ij); end

vb_0 = vib(ik_T);

for ijk = 1:ik_T
 nijk = ik_T+1 - ijk; nvib(nijk) = vib(ijk); end

for k8 = 1:ik_T
 vb_2(k8) = nvib(k8); vs_2(k8) = nvib(k8); end

  vB = B_0*(vb_1);
  vs_2 = vs_2 - vb_0*vs_1;
  vS0 = (4.5/12.0)*(vs_2);
//
 eY = 0.5*H_0*lambda*(1:1:(2*N)+1);
//*************************************

 D_min = +0.001; D_max = 40*D_0;
 Q_min = +0.001; Q_max = 40*Q_0;

vd0 = ones(1,(2*N+1));
vD0 = D_0*vd0; vH0 = vD0 - vS0;

q0  = 0.04;
vq0 = ones(1,(2*N+1)); vQ0 = q0*Q_0*vq0;

vh0 = ones(1,(2*N+1));
vDdot0=0*vh0; ome=100; P_max=100.0; P_min=0.0;

// Inflow-(_i) boundary cond.
    B_in = vB(1);  S_in = vS0(1);
    D_in = vD0(1); H_in = D_in + S_in;
    Q_in = vQ0(1);

// Friction-inclination of bottom
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vRtilda(kk) = ((vB(kk)*vD0(kk))/ ...
                 (vB(kk)+2*vD0(kk)))^(2/3);
  vI0(kk) = (vQ0(kk)/(vB(kk)*vD0(kk))...
```

```
                *k_s*vRtilda(kk)))^2;
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4) + vSf(k4);
    vH0(k4) = vH0(k4) + vSf(k4);
    vD0(k4) = vH0(k4) - vS(k4);
end

//  outflow-(_o) boundary cond
    B_o = vB(kk+1);  S_o = vS(kk+1);
    D_o = vD0(kk+1); H_o = vH0(kk+1);
    Q_o = vQ0(kk+1);

// inflow generation
t_st_1 = 260.0; r_1_0=0.04*Q_0; r_1_1=0.20*Q_0;
t_st_2 = 510.0; r_2_0=0.;       r_2_1=0.20*Q_0;
t_st_3 = 760.0; r_3_0=0.;       r_3_1=0.40*Q_0;

T_fin  = 1000.0;

// inflow slew rate
g_st = 100.0; u_up_st = +1.0; u_dn_st = -1.0;
tau_st = 2.0; Q_st_0 = 0.04*Q_0;

// outflow by GMS
g_o = k_s*((I_r)^(1/2));

// Datatransfer to Plots
CC = 21;           // no of channels
CN = 1000;         // no of clockticks to Tfin
delT = T_fin/CN;   // readout-interval for clock ticks
Asize = 1.01*CC*CN;// size of data arrays
```

```
// s_c5_06_crunplot
//  Glf  2017_06_15

// Hydraulic jump on inclined & flat bottom
stacksize('max');
exec('s_c5_06_context.sce', -1);
importXcosDiagram('s_c5_06.zcos');
typeof(scs_m); scs_m.props.context;
Info = list();
Info = scicos_simulate(scs_m,Info);
//***********************************
for kfig = 1:1:10,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

T_st = 200.; T_en = T_fin;

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[T_st,0.0,T_en,50.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[T_st,0.0,T_en,3.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[T_st,-1.0,T_en,2.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[T_st,0.0,T_en,4.0]);
xtitle("F_2 to F_40"); xgrid(1);

f5 = scf(5);
plot2d(W.time,W.values,vcolor,rect=[T_st,-1.,T_en,+1.0]);
xtitle("W_1 to W_39"); xgrid(1);

f6 = scf(6);
plot2d(P.time, (P.values - g*D.values), vcolor,...
       rect=[T_st,-2.5,T_en,2.5]);
xtitle("deltaP*_1 to  deltaP*_39"); xgrid(1);

//*******************************************************
eX = H_0*lambda*(1:1:N); eXend = 22;
for k6= 1:1:N,  vSr(k6) = vS(2*k6-1); end

vT=[t_st_1-10.0,t_st_2-10.0,t_st_3-10.0,T_fin-10.0];
vs = ['0','0','0', '0'];
for k8 = 1:1:4, vs(k8) = msprintf('%5.0f',vT(k8));
end; vcolorP=[1,13,2,5,6];

f7 = scf(7); clf(); yQ =Q.values;
vQ_ = [yQ(250,:)',yQ(500,:)',yQ(750,:)',yQ(990,:)'];
plot2d...
(eX', [vSr,vQ_], vcolorP, rect=[0.,-0.0,eXend,+50.]);
vtitle=["Longit.Profile  Bottom (bk) and Q at",...
vs(1),"s (gn)",",vs(2),"s (bl)",vs(3),"s (rd) and",...
vs(4),"s (mg)"]; xtitle(vtitle); xgrid(1);

f8 = scf(8); clf(); yD =D.values;
vD=[vSr,yD(250,:)',yD(500,:)',yD(750,:)',yD(990,:)'];
plot2d(eX', vD,vcolorP, rect=[0.,-1.0,eXend,+3.0]);
vtitle=["Longit.Profile  Bottom (bk) and D at",...
vs(1),"s (gn)",",vs(2),"s (bl)",",vs(3),"s (rd) and",...
vs(4),"s (mg)"]; xtitle(vtitle); xgrid(1);

f9 = scf(9); clf(); yD =D.values;
vH=[vSr,yD(250,:)'+vSr,yD(500,:)'+vSr,...
        yD(750,:)'+vSr,yD(990,:)'+vSr];
plot2d(eX', vH,vcolorP, rect=[0.,-1.0,eXend,+2.0]);
vtitle=["Longit.Profile  Bottom (bk) and H at",...
vs(1),"s (gn)",",vs(2),"s (bl)",",vs(3),"s (rd) and",...
vs(4),"s (mg)"]; xtitle(vtitle); xgrid(1);

f10 = scf(10); clf(); yF =F.values;
vF=[vSr,yF(250,:)',yF(500,:)',yF(750,:)',yF(990,:)'];
plot2d(eX',vF, vcolorP, rect=[0.,-1.00,eXend,+4.0]);
vtitle =["Longit.Profile Bottom (bk) and F at",...
vs(1),"s (gn)",",vs(2),"s (bl)",",vs(3),"s (rd) and",...
vs(4),"s (mg)"]; xtitle(vtitle); xgrid(1);

f11 = scf(11); clf(); yW =W.values;
vW=[vSr,yW(250,:)',yW(500,:)',yW(750,:)',yW(990,:)'];
plot2d(eX',vW, vcolorP,rect=[0.,-1.0,eXend,+1.0]);
vtitle=["Longit.Profile  Bottom (bk) and W at",...
vs(1),"s (gn)",",vs(2),"s (bl)",",vs(3),"s (rd) and",...
vs(4),"s (mg)"]; xtitle(vtitle); xgrid(1);

f12 = scf(12); clf(); yP =P.values - g*D.values;
vP=[vSr,yP(250,:)',yP(500,:)',yP(750,:)',yP(990,:)'];
plot2d(eX',vP, vcolorP,rect=[0.,-1.0,eXend,+1.4]);
vtitle=["Longit.Profile Bottom (bk) and P-g*D at",...
vs(1),"s (gn)",",vs(2),"s (bl)",",vs(3),"s (rd) and",...
vs(4),"s (mg)"];  xtitle(vtitle); xgrid(1);

// for assembling logitudinal contours
if lambda == (1.000)*lambda0 then
   aD_100=[eX',vD]; aH_100=[eX',vH]; aF_100=[eX',vF];
elseif lambda == 0.6667*lambda0 then
   aD_066=[eX',vD]; aH_066=[eX',vH]; aF_066=[eX',vF];
else
   aD_050=[eX',vD]; aH_050=[eX',vH]; aF_050=[eX',vF];
end
```

```
// s_c5_06_contour plot overlays.sce
// Glf 2017_06_15

  for kfig = 13:1:15,  clf(kfig); end
 eXend = 22;

vcolor5 = [1, 13, 2, 5, 6];

f13 = scf(13);
plot2d([aD_050(:,1)],[aD_050(:,2),aD_050(:,3), ...
        aD_050(:,4),aD_050(:,5),aD_050(:,6)],...
       vcolor5, rect=[0.,-1.0,eXend,+3.0]);
plot2d([aD_066(:,1)],[aD_066(:,2),aD_066(:,3), ...
        aD_066(:,4), aD_066(:,5),aD_066(:,6)],...
       vcolor5, rect=[0.,-1.0,eXend,+3.0]);
plot2d([aD_100(:,1)],[aD_100(:,2),aD_100(:,3), ...
        aD_100(:,4),aD_100(:,5),aD_100(:,6),],...
       vcolor5, rect=[0.,-1.0,eXend,+3.0]);
legend("S","Q_0 = 2.0 m3/s","10 m3/s","20 m3/s","40 m3/s");
xtitle("profiles D"); xgrid(1);

f14 = scf(14);
plot2d([aH_050(:,1)],[aH_050(:,2),aH_050(:,3),...
        aH_050(:,4),aH_050(:,5),aH_050(:,6)],...
       vcolor5, rect=[0.,-1.0,eXend,+2.0]);
plot2d([aH_066(:,1)],[aH_066(:,2),aH_066(:,3),...
        aH_066(:,4),aH_066(:,5),aH_066(:,6)],...
       vcolor5, rect=[0.,-1.0,eXend,+2.0]);
plot2d([aH_100(:,1)],[aH_100(:,2),aH_100(:,3),...
       aH_100(:,4),aH_100(:,5),aH_100(:,6),],...
       vcolor5, rect=[0.,-1.0,eXend,+2.0]);
legend("S","Q_0 = 2.0 m3/s","10 m3/s","20 m3/s","40 m3/s");
xtitle("profiles H"); xgrid(1);

f15 = scf(15);
plot2d([aF_050(:,1)],[aF_050(:,2),aF_050(:,3),...
        aF_050(:,4),aF_050(:,5),aF_050(:,6)],...
       vcolor5, rect=[0.,-1.0,eXend,+4.0]);
plot2d([aF_066(:,1)],[aF_066(:,2),aF_066(:,3),...
        aF_066(:,4),aF_066(:,5),aF_066(:,6)],...
       vcolor5, rect=[0.,-1.0,eXend,+4.0]);
plot2d([aF_100(:,1)],[aF_100(:,2),aF_100(:,3),...
        aF_100(:,4),aF_100(:,5),aF_100(:,6),],...
       vcolor5, rect=[0.,-1.0,eXend,+4.0]);
legend("S","Q_0 = 2.0 m3/s","10 m3/s","20 m3/s","40 m3/s");
xtitle("profiles F"); xgrid(1);
```

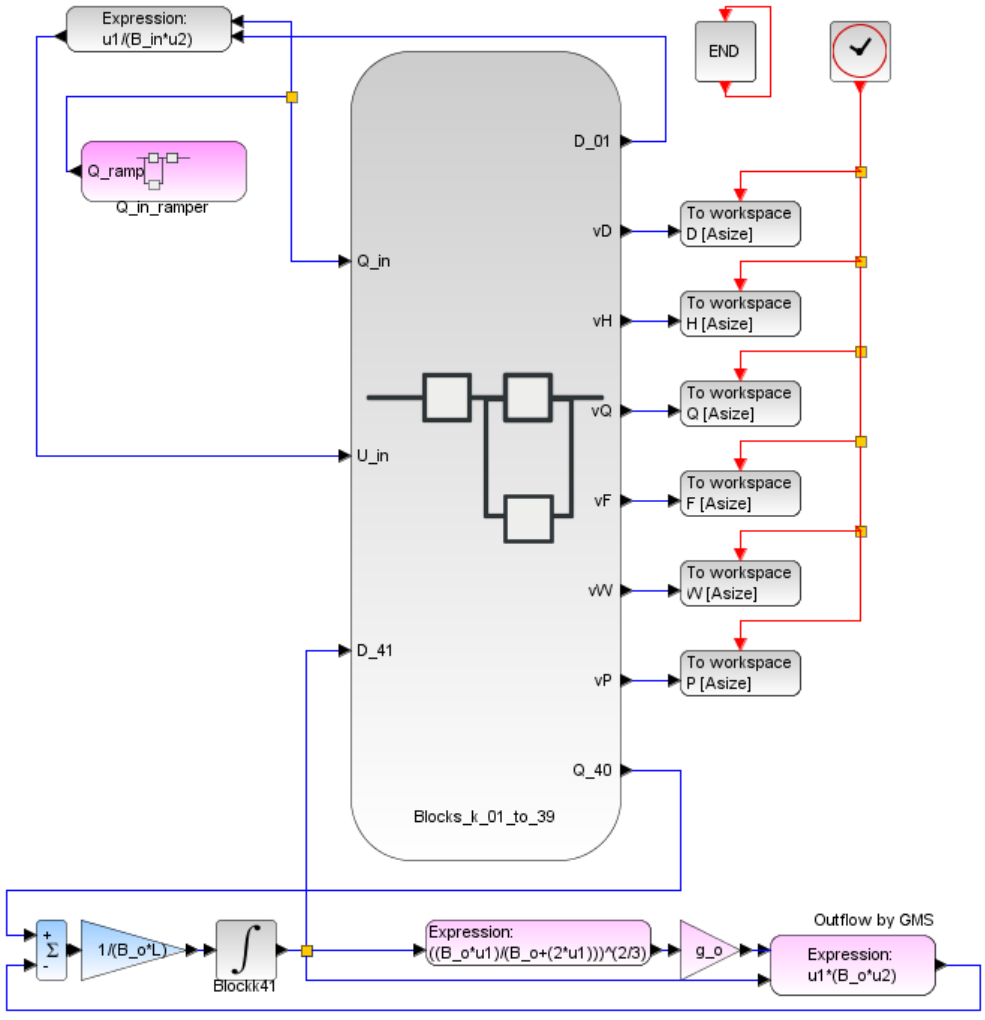


Figure 5.75: Main diagram for Case 6: Hydraulic Jump after a Dam

5.9.3 Simulation Results

Transients for $L = 1.0\ m$

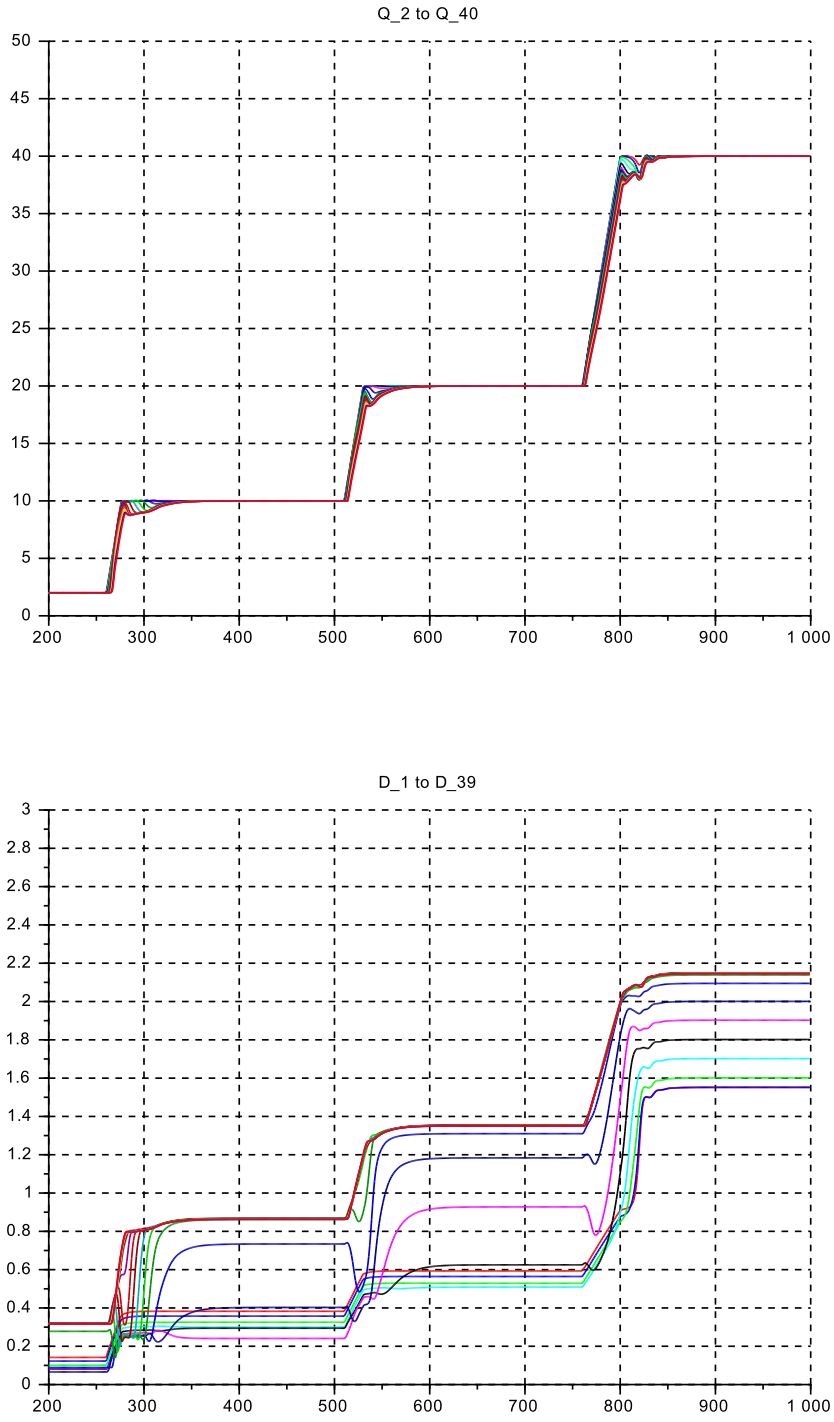


Figure 5.76: C 6: Transient $vQ(t)$ (top) and $vD(t)$ (bottom)

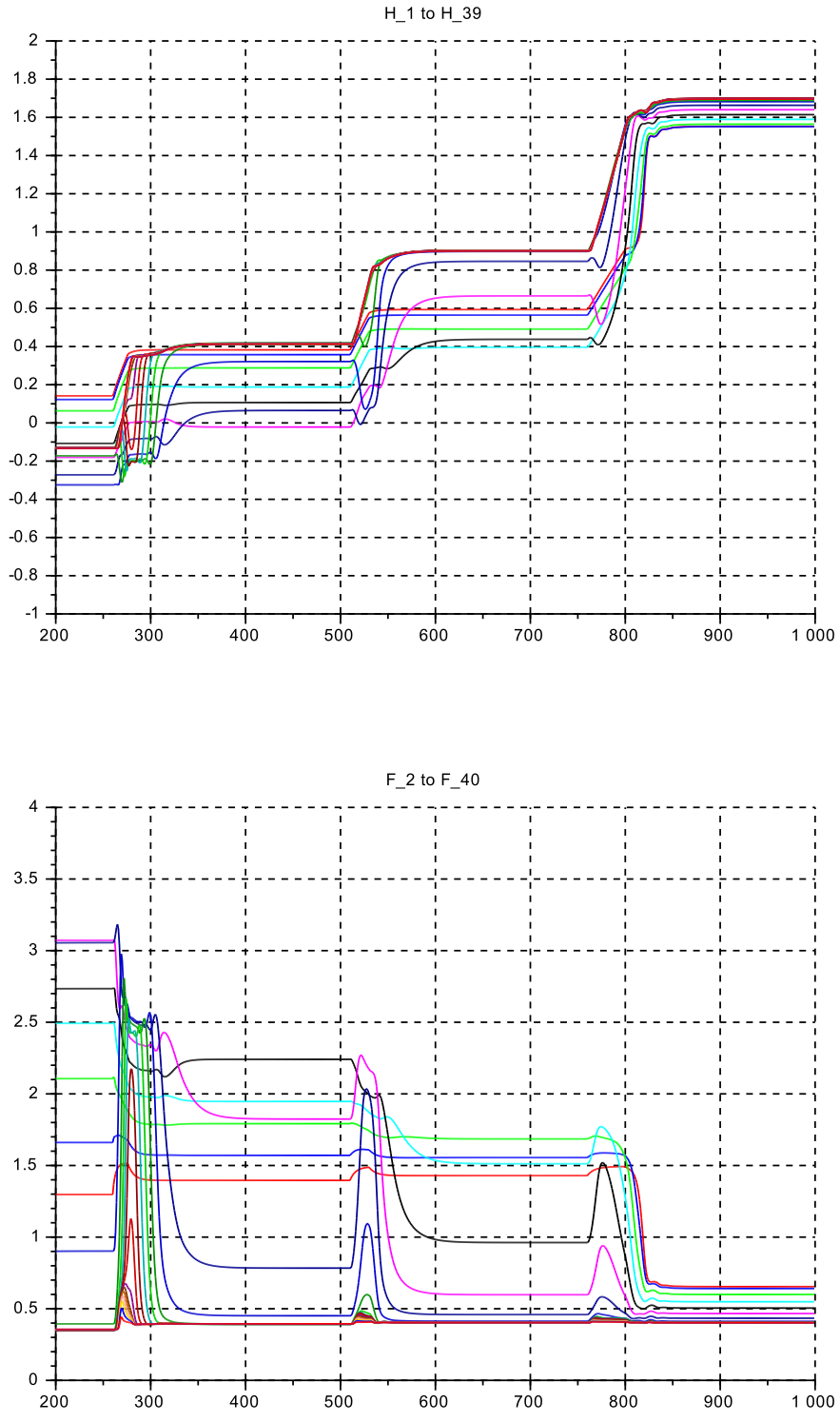


Figure 5.77: C 6: Transient $vH(t)$ (top) and $vF(t)$ (bottom)

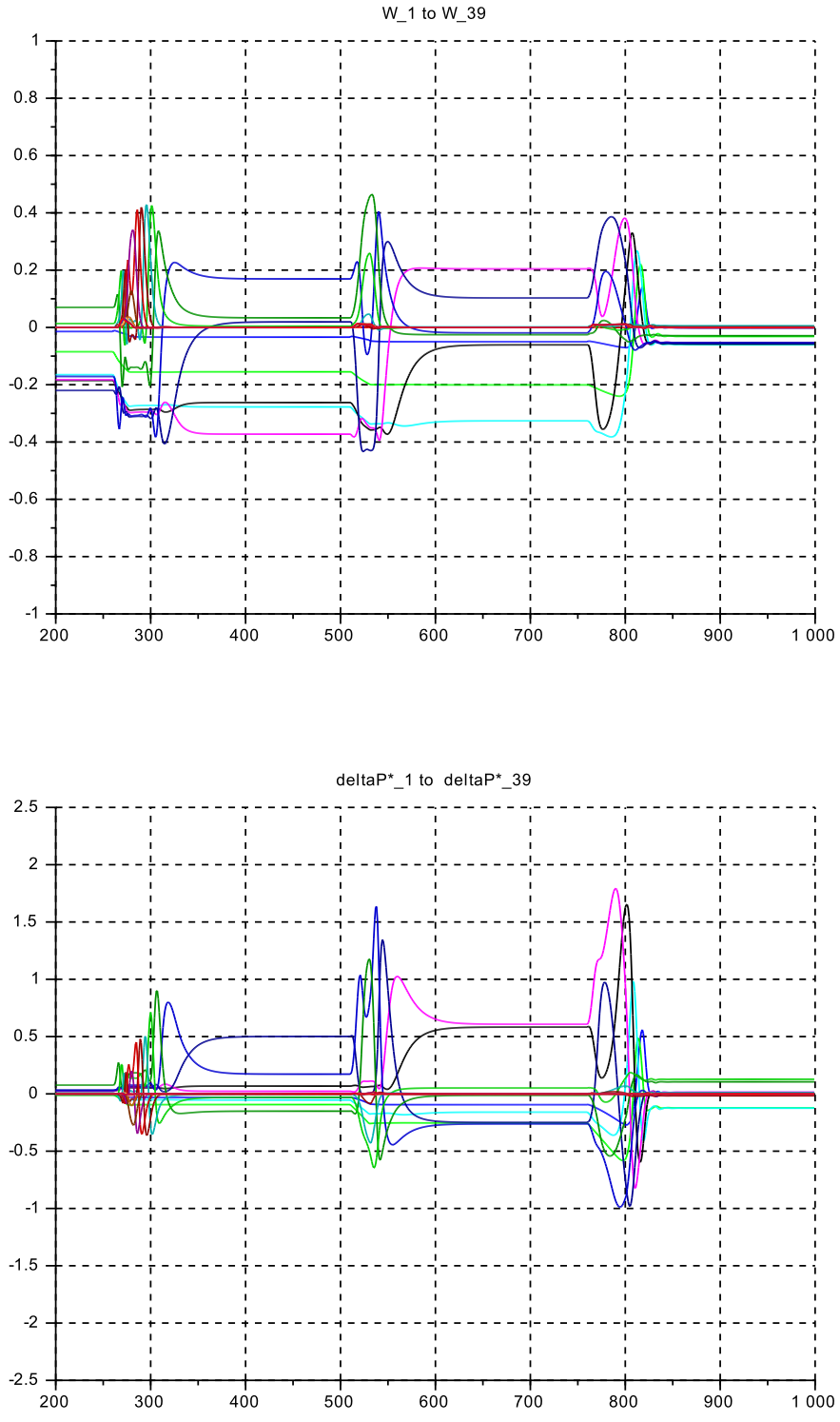
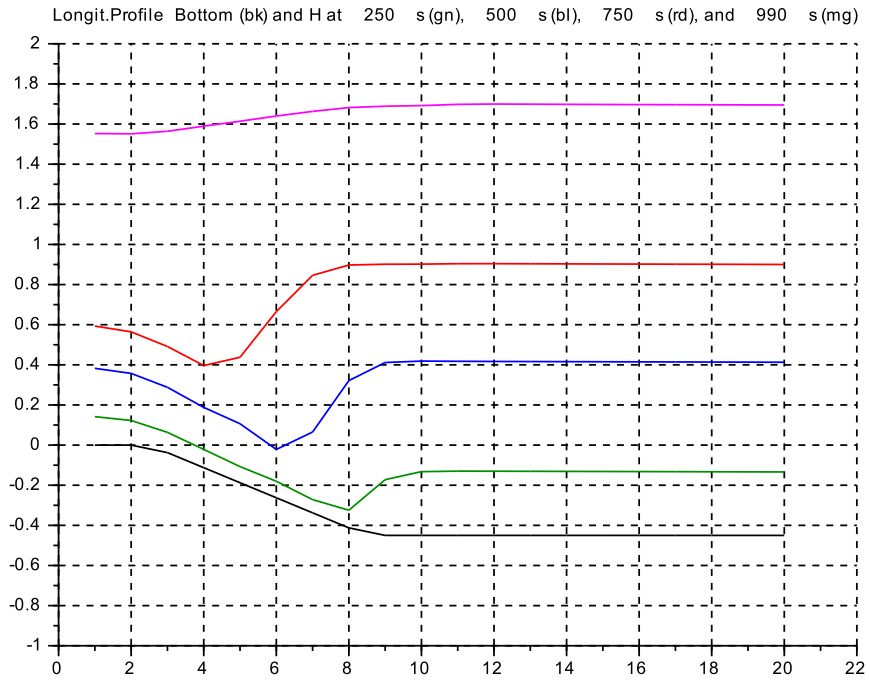


Figure 5.78: C 6: Transient $vW(t)$ (top) and $v\Delta P^*(t)$ (bottom)

Some longitudinal Profiles at $L = 1.0\ m$



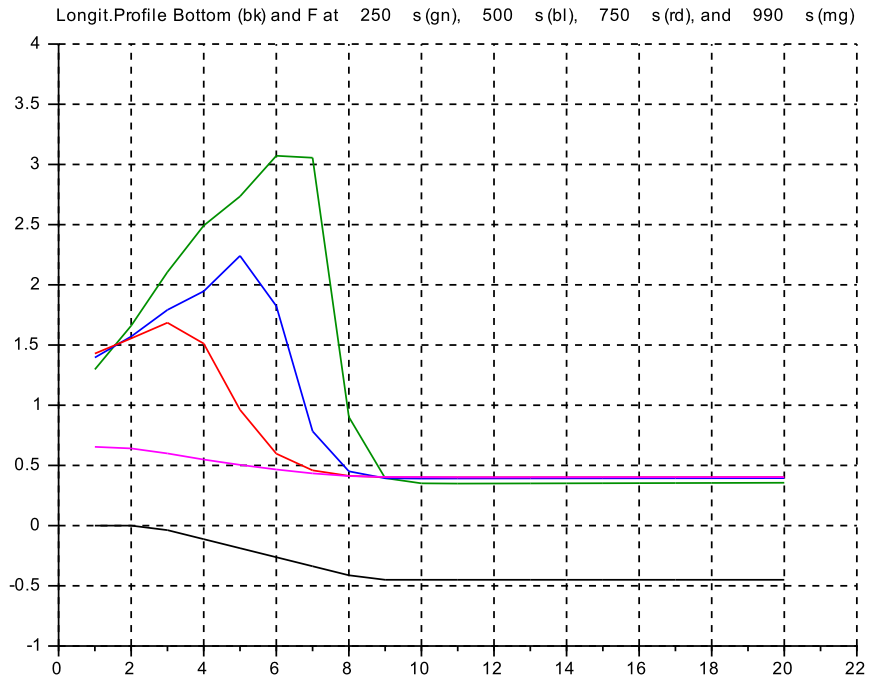


Figure 5.79: C 6: Profiles for $vH(x)$ (top) and $vF(x)$ (bottom)

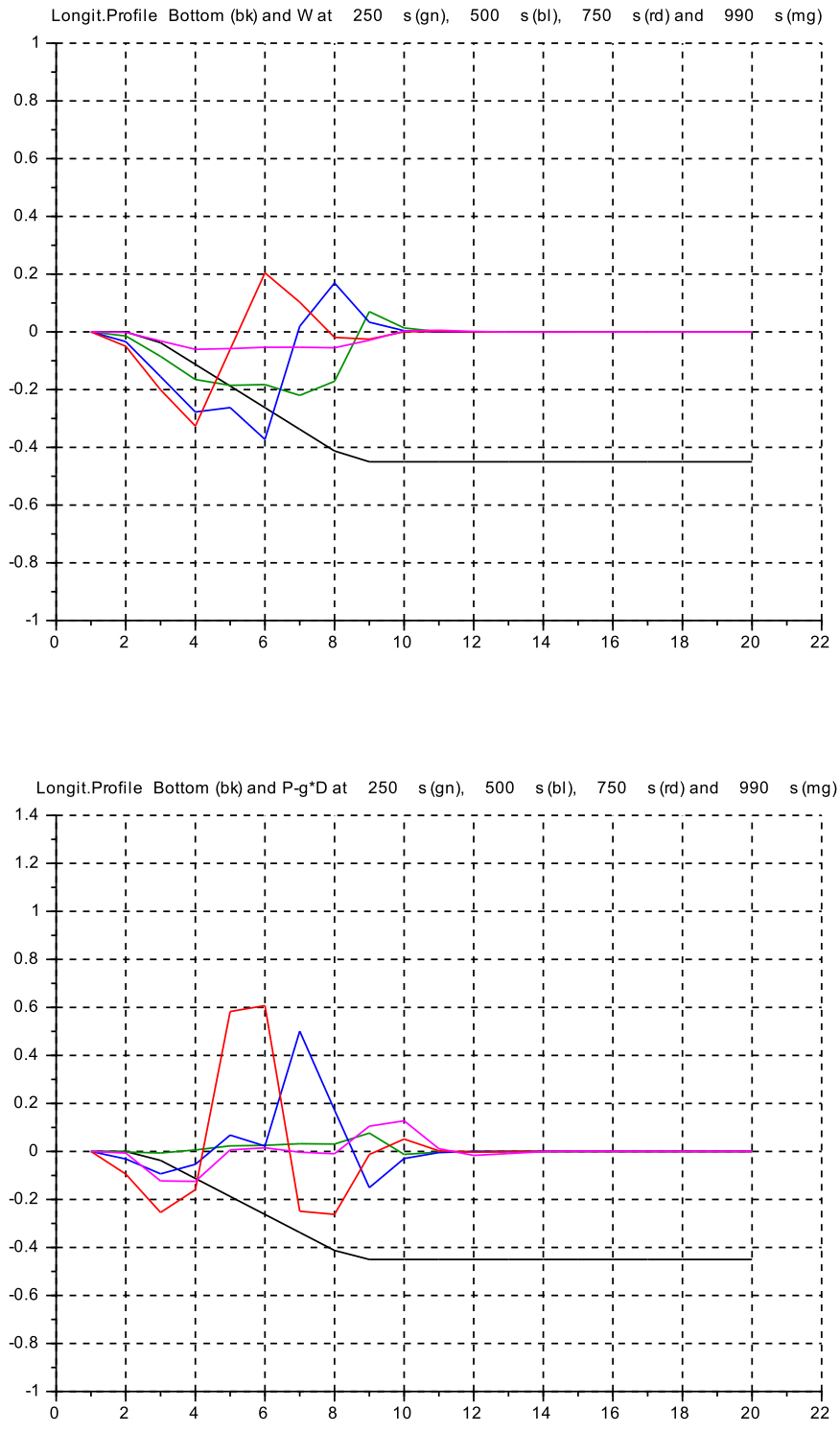


Figure 5.80: C 6: Profiles for $vW(x)$ (top) and $v\Delta P^*(x)$ (bottom)

Selection of Profiles for different compartment lengths

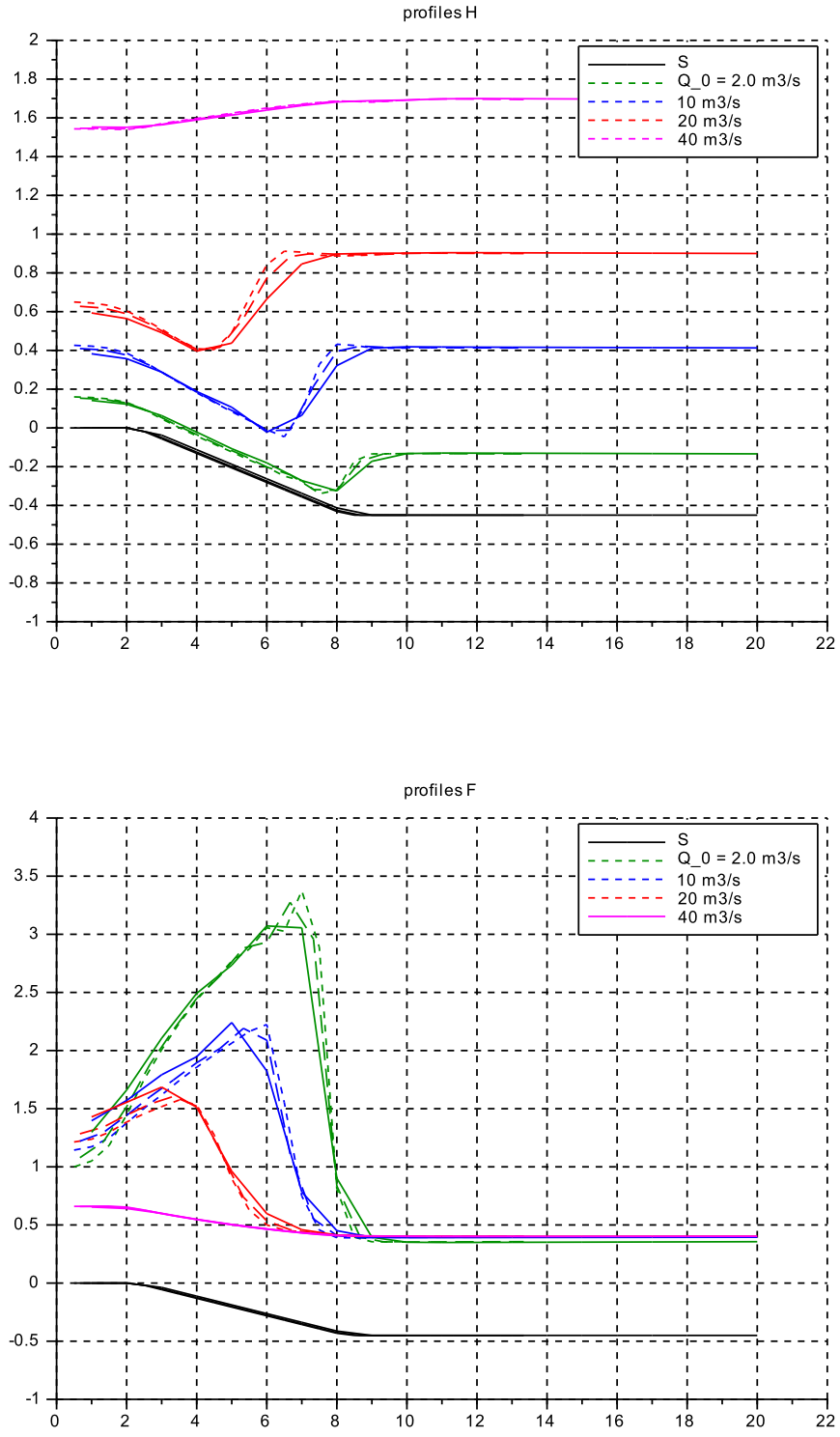


Figure 5.81: C 6: Profiles for $vH(x)$ (top) and $vF(x)$ (bottom)
for $L = 1.0\ m$ (full line) $L = 0.667\ m$ (long dashes) $L = 0.50\ m$ (short dashes)

Profiles for higher friction (GMS-coefficient $k_s := 40$) at different compartment lengths

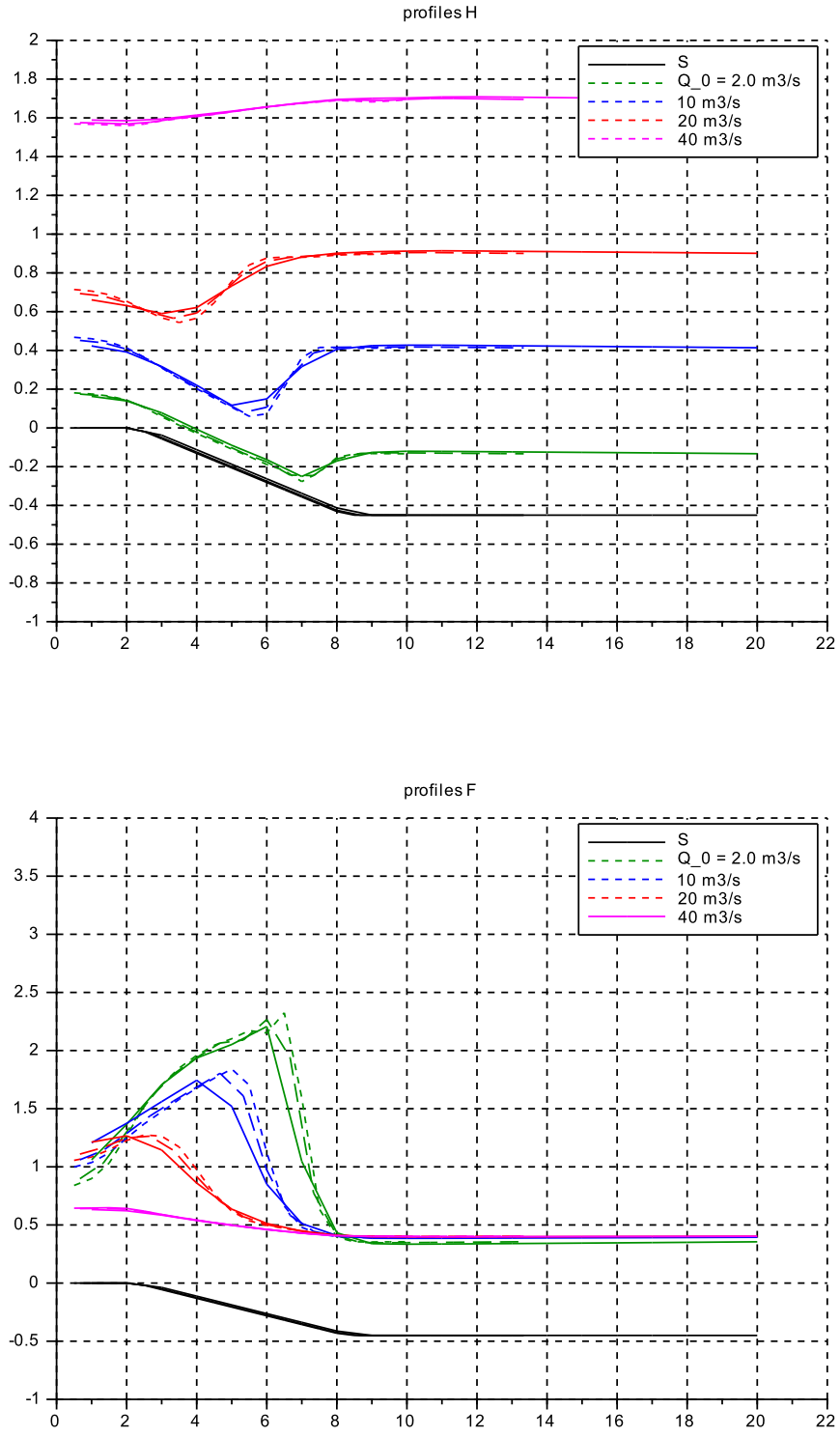


Figure 5.82: C 6: Profiles for $vH(x)$ (top) and $vF(x)$ (bottom) with $k_s := 40$
for $L = 1.0\ m$ (full line) $L = 0.667\ m$ (long dashes) $L = 0.50\ m$ (short dashes)

5.9.4 Discussion for Case 6

Transients

- It is not evident from the traces if a hydraulic jump (abbreviated to 'h-j' in the following) exists and what its shape and location are.
- The traces of $D, H, W, \Delta P^*$ at around $t = 300\ s$ are similar to those in Case 5: There seems to be an upstream moving h-j.
- Following the step-ups on $Q(t)$ there are strong transients in a time window of $\approx 300\ s$, especially on F (with crossings of $F = 1$) and on ΔP^* (up to ± 2.5 units) !

Profiles

- Here the h-j is clearly visible on all variables $D, H, W, \Delta P^*$.
- The h-j shows up for flows at very low, at low and at intermediate level. For high flow $Q(t) = 40\ m^3/s$ the h-j disappears, the dam is submerged in the flow. Both findings correspond qualitatively to observations on real installations.
- Note that the h-j location moves upstream for increasing flow Q. At about $Q \approx 30\ m^3/s$ it reaches the dam crown and then disappears (simulation results not shown here).
- The length of the h-j in vD, vH is $\approx 2\ m$ at its steepest part and about 3...4 m overall. This does not vary perceptibly for the three flow levels.
- vF shows a slow increase along the acceleration zone (over the bottom slope), and a sharp downturn at the h-j start point.
- On $vW, v\Delta\ P^*$ the variations between neighboring data points are very large and abrupt. This indicates that the 'large lumps' spacial discretisation applied here seems to be at its end, again.

Effect of selecting L on Profiles, with nominal friction

- The h-j location does not change significantly and is slightly steeper.
- The abrupt turndown on vF is not rounded off (as one would presume). So the starting of the h-j is still very distinct and only gets more rounded at intermediate flow ($Q = 20\ m^3/s$).
- The F numbers are approx. 1.25 at the inflow for the three lower flow levels, but subcritical (≈ 0.7) at the 'flooding' high flow level. There is no large spread of the F-values over the three values of L. The main effect is from the Q level.
- The basic formula linking the Froude numbers upstream and downstream of the h-j does not apply, as the flow in the upstream zone is still accelerating, especially for $Q = 20\ m^3/s$.

Effect of increased friction $k_s := 40$

- As expected, the rising flank of the h-j is less steep now,
- the starting zone is less abrupt, but not more spread out on x for the L-values.
- The peak values of F are lower, as expected,
- and the effect from L on the size and location of F_peak is still weak.

Chapter 6

Level Control Design

6.1 Overview

The **aim** of this chapter is to use the models developed so far to develop a *systematic design* of level control loops. This means that classic model based design techniques for linear plant models are to be applied, such as pole placement for low order models and frequency domain methods (gain and phase margins) on high order models. In a second step the design result is to be validated on the high order nonlinear model.
This will be done for three typical operational situations that is control configurations, see Fig.6.1:

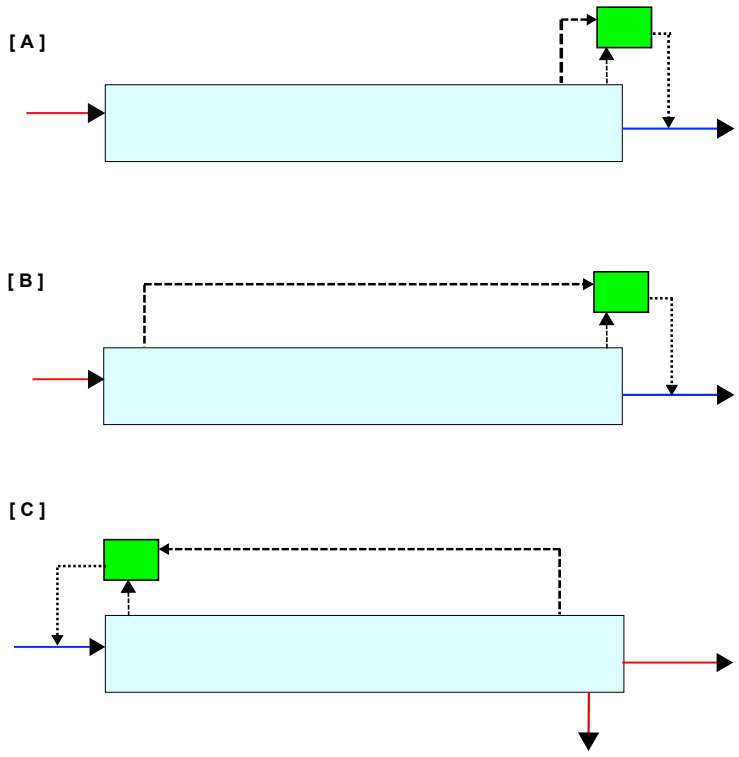


Figure 6.1: The three situations [A], [B], [C], see text

[A] Consider an upstream forebay to a hydropower station, where the level at the outflow end is controlled by acting on the out-flow, that is turbine and weir openings. This situation is known as 'co-located sensor-actuators'. The dominant disturbance response is on the inflow, the level reference

response is mostly not needed in practical cases. This is the typical 'industry standard' case.

[B] Consider the same plant layout, but where now the water level near the inflow location needs to be controlled to a narrow interval, due to specifications from a regulatory agency. From experience this is a more difficult control problem than situation [A].

[C] Consider the situation typical for 'irrigation channels', where there is an additional lateral outflow located near the lower end of the channel section to supply the irrigated fields. There the water level is to be maintained within a narrow interval. The actuator then is placed on the inflow to this channel section, and has to compensate any disturbances on the lateral outflow take. Furthermore the next downstream channel section is regulated in the same manner. Thus a disturbance on its lateral outflow will generate eventually the same disturbance on the longitudinal outflow of the upstream channel section. And this adds to the disturbance on its lateral outflow take.

To keep the investigation focused, the basic channel geometry shall be used, that is with longitudinally constant rectangular cross section, constant width B and constant depth D. The bottom shall be horizontal ($S = const.$), as the friction from the GMS-model shall be neglected [1]. The main operational parameter is flow Q, that is the Froude number F. Only subcritical flow is considered ($0 \leq F < 1$).
The second main parameter is the model order N. First the basic case for $N = 0$ (one compartment) is considered. Then the model with $N \to \infty$ (the partial differential equations 'pde' model) is investigated. Next a linearized state space model with variable order N consisting of $2N + 1$ compartments is looked into. Finally the nonlinear time domain model from chapter 3 is considered with $N = 20$ and with large and comparatively rapid flow variations.

6.2 The one compartment model

6.2.1 Limitations

This model consists of only one compartment for the mass/volume balance. No sloshing modes and no travelling waves are considered. So it is valid only for very low frequencies, that is very slow transients. Nevertheless it is often used in practice, due to its simplicity, see e.g. [10], and therein chap.8.

6.2.2 Modelling

Volume balance: in s.i. units
$$\frac{d}{dt}V = Q_{in} - Q_{out}$$

in 'per units' referenced to the local equilibrium
$$\frac{\overline{V}}{\overline{Q}} \cdot \frac{d}{dt} \frac{V}{\overline{V}} = \frac{Q_{in}}{\overline{Q}} - \frac{Q_{out}}{\overline{Q}}$$

deviations in 'per units' around the local equilibrium
$$\frac{\overline{V}}{\overline{Q}} \cdot \frac{d}{dt} \frac{\delta V}{\overline{V}} = \frac{\delta Q_{in}}{\overline{Q}} - \frac{\delta Q_{out}}{\overline{Q}}$$

finally after re-naming variables
$$\overline{T_f} \cdot \frac{d}{dt} v = q_{in} - q_{out}$$

- Note that the 'filling time constant' $\overline{T_f}$ is not a constant coefficient, but floats with the local operating flow $\overline{Q}$, and tends to infinity for very low local flow, that is for $F \to 0$.
- Also note that the volume content variable v needs to be replaced by the more directly measurable state variable $d = \delta D / \overline{D}$

$$\overline{V} = \overline{B} \cdot \overline{L} \cdot \overline{D}$$
$$\text{and} \quad \delta V = \overline{B} \cdot \overline{L} \cdot \delta D$$
$$\text{into the volume balance} \quad \overline{T_f} \cdot \frac{d}{dt} d = q_{in} - q_{out}$$

[1] This reduces the intrinsic damping in the channel model. Thus the resulting loop design will be on the pessimistic side especially for larger flows Q, and Froude numbers $0 \ll F < 1$.

As a next step the (local, floating) reference variables used above are replaced by the (constant) design values for the channel.

$$V_d \text{ instead of } \overline{V}; \quad D_d \text{ instead of } \overline{D} \quad \text{and} \quad Q_d \text{ instead of } \overline{Q}$$

yielding

$$\begin{aligned} \rightarrow \frac{V_d}{\overline{V}} \cdot \frac{\overline{Q}}{Q_d} \cdot \frac{\overline{V}}{\overline{Q}} \cdot \frac{d}{dt}\left(\frac{\delta V}{\overline{V}}\right) &= \left(\frac{\overline{Q}}{Q_d} \cdot \frac{\delta Q_{in}(t)}{\overline{Q}}\right) - \left(\frac{\overline{Q}}{Q_d} \cdot \frac{\delta Q_{out}(t)}{\overline{Q}}\right) \\ \text{finally} \quad \frac{V_d}{Q_d} \cdot \frac{d}{dt}\left(\frac{\delta D}{D_d}\right) &= \left(\frac{\delta Q_{in}(t)}{Q_d}\right) - \left(\frac{\delta Q_{out}(t)}{Q_d}\right) \end{aligned}$$

with appropriate reference values $_d$:

- D_d is set to the reference value for depth used for the control loop,
- $V_d := L\_tot\ B\ D_d$ with total channel length L_tot,
- $Q_d := B\ D_d\ U_d$ with Q_d specified, and
- $F_d := U_d/\sqrt{g \cdot D_d}$, for instance $F_d := 0.40$,
- and

$$T\_f\_d = \frac{V_d}{Q_d} = \frac{L\_tot\ B\ D_d}{B\ D_d\ U_d} = \frac{L\_tot}{U_F}\ \frac{U_F}{U_d} = \frac{1}{F_d}\ \frac{L\_tot}{U_F}$$

  where L_tot/U_F is the travelling time of the Froude wave at zero main flow over the total channel length

Actuator and Sensor spans
The actuator full span 0 ... 100% shall deliver Q_d and shall have a linear characteristic.
Further the sensor full span 0 ... 100% is set to D_d [2].

6.2.3 Controller Design

The focus is on case [A] of Fig.6.1, that is $u := q_{out}$ and $y := d$.

A PI-controller is used in parallel form with time scaling of the integral action to T_i.

$$u(s) = (y(s) - r(s)) \cdot \left(k_p + \frac{k_i}{s \cdot T_i}\right); \quad \rightarrow \quad G_r(s) = \frac{k_p \cdot sT_i + k_i}{sT_i}$$

Note the sign selection to obtain the negative loop feedback.

The closed loop characteristic equation is

$$0 = 1 + G_u(s) \cdot G_r(s) = (s^2) + \left(s\frac{k_p}{T\_f\_d}\right) + \left(\frac{k_i}{T_i \cdot T\_f\_d}\right)$$

Using pole placement as the design method to targets: bandwidth Ω, damping rate $2D := 2.0$,
and setting $T_i := T\_f\_d$:

$$\begin{aligned} 0 &= s^2 + s \cdot (2D \cdot \Omega) + (\Omega)^2 \\ \text{yields} \quad k\_p\_d &= 2D \cdot (\Omega \cdot T\_f\_d) \\ \text{and} \quad k\_i\_d &= (\Omega \cdot T_i) \cdot (\Omega \cdot T\_f\_d) \quad := \quad (\Omega \cdot T\_f\_d)^2 \end{aligned}$$

Thus both k_p_d, k_i_d have fixed values and do not depend on the equilibrium flow $\overline{Q}$ at the current operating condition. This simplifies the implementation.
From general experience, suitable values are $\Omega \cdot T\_f\_d = 1.0...2.0$, that is $k\_p\_d = 2...4$ and $k\_i\_d = 1.0...4.0$.
Note that sensor and actuator gains are set to unity here, but their actual values must be included in any application.

[2] For practical cases, typically an offset on the sensor zero of up to +50% is implemented to allow measuring depths above the reference/setpoint

6.2.4 Closed Loop Bandwidth Limitation

Theoretically there is no limitation on the closed loop bandwidth Ω with the pole placement method. But in most practical applications there are a number of small time constants from sensors, actuators, and from neglected higher frequency dynamics within the process model. In other words the pure integrator model used so far is only valid for very low frequencies. The higher frequency dynamics will add additional phase lags.
It is common usage in control system design to replace those higher frequency dynamics by one pure delay element with unity gain and delay time T_t and place it in series to the integrator element with time constant T_f_d. This approach is used in such well known techniques as the Ziegler-Nichols rules and related ones.

Putting this delay element in series with the pure integrator model from above yields

$$G_u(s) = k\_u\_d \cdot e^{sT_t} \cdot \frac{1}{s \cdot T\_f\_d}$$

The delay element will add $-90°$ at frequency $\omega_{crit} \cdot T_t = \pi/2$ to the $-90°$ from the pure integrator, and thus produce a zero phase margin. In other words ω_{crit} is the oscillation frequency in rad/sec at the stability limit with a P-controller.
Clearly the design bandwidth Ω must be selected below ω_{crit} to produce a sufficient stability margin (both in gain and phase margins). A practical rule-of-thumb is applying a factor φ in the Bode plot

$$\varphi := \frac{\omega_{crit}}{\Omega} \quad \text{from} \quad \approx 8.0 \quad \text{down to} \quad \approx 4.0$$

Generally, an acceptable closed loop damping is achieved in the given range of φ, where higher values of φ produce better damping.

If this were applied blindly to the specific case from above, intuitively taking the traveling time of the Froude wave at zero flow operation as the delay T_t, and using the relative bandwidth shift rBW from open loop to closed loop defined by $rBW := \Omega \cdot T\_f\_d$:

$$T_t := \frac{L}{U_F} \quad \text{and with} \quad T\_f\_d := \frac{V_d}{Q_d} = \frac{L}{U_d} = \frac{L}{U_F} \cdot \frac{U_F}{U_d} = T_t \frac{1}{F_d}$$

$$\text{from above} \quad \omega_{crit} = \frac{\pi}{2} \cdot \frac{1}{T_t}$$

$$\text{also} \quad \Omega = rBW \cdot \frac{1}{T\_f\_d} = rBW \cdot \frac{1}{T_t} \cdot F_d$$

$$\rightarrow \quad \varphi \cdot rBW = \frac{\omega_{crit}}{\Omega} \cdot \Omega \cdot T\_f\_d = \omega_{crit} \cdot T\_f\_d = \frac{\pi}{2} \cdot \frac{1}{T_t} \cdot \frac{T_t}{F_d} = \frac{\pi}{2} \cdot \frac{1}{F_d}$$

Inserting numerical values would yield with $L = 500$, $U_F = 5.0$ and $F_d := 0.40$:
that is $\varphi \cdot rBW = \pi/0.80 \approx 4.0$,
and also selecting $\varphi := 4$:

$$rBW \approx 1.0; \quad \text{and thus} \quad k\_r\_d \approx 2.0; \quad k\_i\_d \approx 1.0;$$

$$\text{and} \quad T\_i\_d := T\_f\_d = (L/U_F)/F_d = (500/5.0)/0.40 = 250\ s$$

A note of caution:
Note that the basic assumption to this design approach (the delay connected in series to the integrator) does *not* hold here. The situation is quite different: For the industry standard case A the control input is directly located at the level sensor location, so there is 'no' delay. But a large delay is on the echo wave reflected at the inflow end and having twice the distance L_tot to travel. So the controller gains calculated above are far from reliable...
Therefore a new design method is required, based on a much better model. This is done in the next section.

6.3 The infinite number of compartments model, 'sys_inf'

6.3.1 Motivation and overview

This model is based on the partial differential equation (***pde*** for short) for the variables flow Q and depth D with respect to time t and longitudinal coordinate X.

- The main motivation is to provide a model much closer to the real physics of the plant than the simplistic first order one from the last section.
- Further the approach through the *pde* is very often used in the relevant references ([7, 8, 9, 10] and further references therein).
- It will be shown that this approach produces a transparent and simple procedure for selecting the gains of the PI-level controller commonly used in industrial projects.
- In addition it produces simple gain scheduling functions in closed form[3] as functions of the flow operating point.
- It also produces acceptable gain values (being within the range from operating experience with the usual actuator equipment).
- And the procedure covers all three control situations from Fig.6.1.

6.3.2 Modelling

Overview

First the *PDE* is derived, both at different operating points (Froude numbers F), but without GMS-friction, in s.i.-units. The final result is given in 'per unit' form, referenced to the design point $_d$.
Then the PDE is solved as function of F with zero friction, without upper and lower channel end boundaries, first for the water flow $q_d(x,t)$ and then for the water depth $d_d(x,t)$. Next the end boundaries are inserted, again first for q_d and then for d_d. Full reflections are assumed here.
The resulting open loop time responses to inflow and outflow steps are shown by simulations. Finally the corresponding transfer functions are given in symbolic form and frequency responses are shown in Bode-plots.

Deriving the pde

Assumptions for the channel flow: total length L_0, rectangular cross section, constant width B_0, constant depth D_0, bottom horizontal (as for zero friction), Froude celerity $U_0 = \sqrt{g \cdot D_0}$, and flow Q at operating point.

Volume Balance for element with longitudinal length ΔL at position X:

$$\begin{aligned}
\text{z-axis} \qquad Q_z &= \Delta L\ B_0\ U_z = \Delta L\ B_0\ \frac{\partial D(X)}{\partial t} \\
\text{x-axis} \qquad \Delta Q &= -\frac{\partial Q}{\partial X}\ \Delta L \\
\text{volume balance} \qquad Q_z &\stackrel{!}{=} \Delta Q \\
\text{or} \qquad \Delta L\ B_0\ \frac{\partial D(X)}{\partial t} &= -\frac{\partial Q}{\partial X}\ \Delta L \\
\text{finally} \qquad B_0 \frac{\partial D(X)}{\partial t} + \frac{\partial Q}{\partial X} &= 0
\end{aligned}$$

[3] no need to be developed by time-consuming experimentation during plant startup or by simulations

Scaling from 's.i.'-units to 'p.u._d.'-units, using as reference values:
$x = X/L_0$; $d_d = D/D_d = D/D_0$; $q_d = Q/Q_d$; $F_d = U_d/U_0 = Q_d/Q_0$;
and Froude wave travelling time at zero flow regime $T_0 = L_0/U_0$

$$\begin{aligned} B_0\ D_0\ \frac{\partial}{\partial t} d_d \ + \ \frac{1}{L_0}\ Q_d\ \frac{\partial\,(Q/Q_d)}{\partial\,(X/L_0)} &= 0 \\ \frac{L_0\ B_0\ D_0}{Q_d}\ \frac{\partial d_d}{\partial t} \ + \ \frac{\partial q_d}{\partial x} &= 0 \\ \text{where} \quad \frac{L_0\ B_0\ D_0}{Q_d} &= \frac{L_0\ B_0\ D_0}{B_0\ D_0\ U_d} = \frac{L_0}{U_d} = \frac{L_0}{U_0}\ \frac{U_0}{U_d} = \frac{L_0}{U_0}\frac{1}{F_d} = T_0\frac{1}{F_d} \end{aligned}$$

finally

$$T_0\frac{1}{F_d}\ \frac{\partial d_d}{\partial t} = -\ \frac{\partial q_d}{\partial x}$$

Momentum Balance for element with longitudinal length ΔL at position X:

$$\begin{aligned} \text{rate of momentum } \frac{\partial}{\partial t}(m|_{\Delta L} U) &= \varrho \Delta L \frac{\partial Q}{\partial t} \\ \text{pressure force difference } \Delta F_p &= -(\varrho \Delta L) g D_0 B_0 \frac{\partial D}{\partial X} \\ \text{momentum flow difference } \Delta I &= -(\varrho \Delta L)\frac{\partial}{\partial X}\left(\frac{Q^2}{B_0 D}\right) \\ &\quad -(\varrho \Delta L)\frac{1}{(B_0 D)^2}\left[(B_0 D) 2 Q \frac{\partial Q}{\partial X} - Q^2 B_0 \frac{\partial D}{\partial X}\right] \\ \text{momentum balance } (\varrho \Delta L)\frac{\partial Q}{\partial t} &= (\varrho \Delta L)\left(-B_0\ g D_0 \frac{\partial D}{\partial X} - \frac{1}{(B_0 D_0)^2}\left[(B_0 D_0)\ 2\ Q \frac{\partial Q}{\partial X} - Q^2 B_0 \frac{\partial D}{\partial X}\right]\right) \end{aligned}$$

Scaling from 's.i.'-units to 'p.u._d'-units, while taking off $(\varrho \Delta L)$ on both sides

$$\begin{aligned} Q_d\ \frac{\partial q_d}{\partial t} &= -\ B_0\ D_0\ g D_0\ \frac{1}{L_0}\ \frac{\partial d_d}{\partial x} \ - \\ &\quad -\ \frac{Q_d^2}{(B_0\ D_0)^2}\left[(B_0\ D_0)\frac{1}{L_0}\ 2\ \frac{Q}{Q_d}\ \frac{\partial q_d}{\partial x} \ - \ \left(\frac{Q}{Q_d}\right)^2\ B_0 D_0\ \frac{1}{L_0}\ \frac{\partial d_d}{\partial x}\right] \\ L_0\ B_0\ D_0\ U_d\ \frac{\partial q_d}{\partial t} &= -\ B_0\ D_0\ g D_0\ \frac{\partial d_d}{\partial x} \ - \\ &\quad -\ \frac{(B_0\ D_0)^2\ U_d^2}{(B_0\ D_0)^2}\ (B_0\ D_0)\left[2\ \frac{Q}{Q_d}\ \frac{\partial q_d}{\partial x} \ - \ \left(\frac{Q}{Q_d}\right)^2\ \frac{\partial d_d}{\partial x}\right] \\ \text{or} \quad \frac{L_0\ U_d}{g D_0}\ \frac{\partial q_d}{\partial t} &= -\ \frac{\partial d_d}{\partial x} \ - \ \frac{U_d^2}{(g D_0)}\left[2\ \frac{U}{U_d}\ \frac{\partial q_d}{\partial x} \ - \ \left(\frac{U}{U_d}\right)^2\ \frac{\partial d_d}{\partial x}\right] \\ \text{with coefficients } \frac{L_0\ U_d}{g D_0} &= \frac{L_0}{U_0}\ \frac{U_d}{U_0} \ = \ \frac{L_0}{U_0}\ F_d = \ T_0\ F_d \\ \text{and with} \quad \frac{U_d^2}{g D_0} &= F_d^2; \qquad \frac{U}{U_d} = \frac{U}{U_0}\ \frac{U_0}{U_d} = \frac{F}{F_d}; \qquad \left(\frac{U}{U_d}\right)^2 = \left(\frac{F}{F_d}\right)^2 \end{aligned}$$

finally

$$T_0\ F_d\ \frac{\partial q_d}{\partial t} = -\ \frac{\partial d_d}{\partial x} \ - \ 2\ F\ F_d\ \frac{\partial q_d}{\partial x} \ + \ F^2\ \frac{\partial d_d}{\partial x} \ = \ -\,(1 - F^2)\ \frac{\partial d_d}{\partial x} \ - \ 2\ F\ F_d\ \frac{\partial q_d}{\partial x}$$

Remarks:
With nonzero friction the momentum balance turns into

$$T_0\ F_d\ \frac{\partial q_d}{\partial t} + c_f\ q_d = -\ \frac{\partial d_d}{\partial x}\ -\ 2\ F\ F_d\ \frac{\partial q_d}{\partial x}\ +\ F^2\ \frac{\partial d_d}{\partial x}\ =\ -\ (1-F^2)\ \frac{\partial d_d}{\partial x}\ -\ 2\ F\ F_d\ \frac{\partial q_d}{\partial x}$$

And setting both friction and the Froude number to zero, the momentum balance reduces to

$$T_0\ F_d\ \frac{\partial q_d}{\partial t} = -\ \frac{\partial d_d}{\partial x}$$

Together with the volume balance equation from above this is the basic (lateral) wave equation pair well known from textbooks on pde's.

Solving the PDE's, with open ends

The two first order PDE's for the dependent variables d_d, q_d are

$$\begin{aligned} T_0 \frac{1}{F_d}\ \frac{\partial d_d}{\partial t}\ &=\ -\ \frac{\partial q_d}{\partial x} \\ T_0\ F_d\ \frac{\partial q_d}{\partial t}\ &=\ -\ (1-F^2)\ \frac{\partial d_d}{\partial x}\ -\ 2\ F\ F_d\ \frac{\partial q_d}{\partial x} \end{aligned}$$

The initial conditions at time $t = 0$ and along the longitudinal variable x are set to zero. And from the 'open ends on x', there are no boundary conditions.

Step 1 is to eliminate one of the two dependent variables to obtain one PDE of second order. Here d_d is eliminated, by partial differentiation of the first equation w.r.t. x, and by partial differentiation of the second equation w.r.t. t, yielding:

$$\begin{aligned} T_0 \frac{1}{F_d}\ \frac{\partial^2 d_d}{\partial t\ \partial x}\ &=\ -\ \frac{\partial^2 q_d}{\partial x^2} \\ T_0\ F_d\ \frac{\partial^2 q_d}{\partial t^2}\ &=\ -\ (1-F^2)\ \frac{\partial^2 d_d}{\partial x\ \partial t}\ -\ 2\ F\ F_d\ \frac{\partial^2 q_d}{\partial x\ \partial t} \end{aligned}$$

Inserting the first equation in the second one,

$$T_0\ F_d\ \frac{\partial^2 q_d}{\partial t^2}\ =\ +(1-F^2)\ \frac{1}{T_0}\ F_d\ \frac{\partial^2 q_d}{\partial x^2}\ -\ 2\ F\ F_d\ \frac{\partial^2 q_d}{\partial x\ \partial t}$$

and taking off the common factor F_d, then finally

$$T_0^2\ \frac{\partial^2 q_d}{\partial t^2}\ +\ 2\ F\ T_0\ \frac{\partial^2 q_d}{\partial x\ \partial t}\ -\ (1-F^2)\ \frac{\partial^2 q_d}{\partial x^2}\ =\ 0$$

Step 2 is a first Laplace transformation with respect to time, $t \to s$ and $q_d \to \widetilde{q_d}$. Note that initial conditions are zero. The partial derivatives transfer to normal derivatives.

$$(1-F^2)\ \frac{d^2\widetilde{q_d}}{dx^2}\ -\ 2\ F\ (sT_0)\ \frac{d\widetilde{q_d}}{dx}\ -\ (sT_0)^2\ \widetilde{q_d} = 0$$

Step 3 is a second Laplace transformation with respect to length position x, $x \to r$ and $\widetilde{q_d} \to \widetilde{\widetilde{q_d}}$. Note that there are no boundary conditions due to the open ends assumption.

$$(1-F^2)\ r^2\widetilde{\widetilde{q_d}}\ -\ 2\ F\ (sT_0)\ r\widetilde{\widetilde{q_d}}\ -\ (sT_0)^2\ \widetilde{\widetilde{q_d}} = 0$$

Step 4: The characteristic equation for the roots of r then is

$$r^2\ -\ \frac{2\ F}{1-F^2}(sT_0)\ r\ -\ \frac{1}{1-F^2}\ (sT_0)^2\ =\ 0$$

and the roots are

$$\begin{aligned}
r_{1,2} &= \frac{F}{1-F^2}\ (sT_0) \pm \sqrt{\left(\frac{F^2}{(1-F^2)^2}\right)\ (sT_0)^2 + \left(\frac{1-F^2}{(1-F^2)^2}\right)(sT_0)^2} \\
&= \left[\frac{F}{1-F^2} \pm \frac{1}{1-F^2}\sqrt{F^2+(1-F^2)}\right]\ (sT_0) \\
&= \left[\frac{F}{1-F^2} \pm \frac{1}{1-F^2}\right]\ (sT_0) \\
r_1 &= -\frac{1-F}{(1-F)(1+F)}\ (sT_0) \quad = \quad -s\frac{T_0}{1+F} = -sT_1 \\
\text{with } T_1 &= \frac{T_0}{1+F} \le T_0 \quad \text{in downflow direction at position } x=+L \text{ from the inflow point at } x=0 \\
r_2 &= +\frac{1+F}{(1-F)(1+F)}\ (sT_0) \quad = \quad +s\frac{T_0}{1-F} = +sT_2 \\
\text{with } T_2 &= \frac{T_0}{1-F} \ge T_0 \quad \text{in upflow direction at position } x=-L \text{ from the inflow point at } x=0
\end{aligned}$$

Step 5 is the inverse L-Transformation from $r \to x$, and $\widetilde{\widetilde{q}}_d \to \widetilde{q}_d$ with the roots from above

$$\widetilde{q}_{d1} \ = \ e^{r_1} \ = \ e^{-sT_1}; \qquad \widetilde{q}_{d2} \ = \ e^{r_2} \ = \ e^{+sT_2}$$

Step 6 is the inverse L-Transformation from $s \to t$, and $\widetilde{q}_d \to q_d(t \pm L)$. The transfer function from Step 5 is a pure delay. Thus the time response is a pure time shift of the inflows in the downstream and upstream direction.

The Froude wave moves downstream with speed $C_1 = U + U_F = U + U_0$, reaching the point at $X = L$ at time

$$X/C_1 \to L/C_1 = L/(U_0+U) = (L/U_0)/(1+F) = T_0/(1+F) := T_1.$$

In the upstream direction, the Froude wave moves with speed $C_2 = +U - U_F$, and thus reaches the point at $X = -L$ at time

$$-L/C_2 = (-L/-U_F)/(1-F) = T_0/(1-F) := T_2.$$

Thus the responses of $q_d(X,t)$ at a distance $\pm X = \pm x \cdot L$ from the inflow point (at $X = 0$) are

$$\begin{aligned}
q_{d_1} &= q_{d_{in_1}}(t)(t - \frac{X}{C_1}) = 0.5 \cdot q_{d_{in_1}}(t)\ (t - x \cdot T_1); \\
q_{d_2} &= 0.5 \cdot q_{d_{in_2}}(t)\ (t - (-x \cdot T_2))
\end{aligned}$$

Remarks

Observe that for $F \ \to \ 1.0$ the upstream travelling time goes to infinity, $T_2 = T_0/(1-F) \to \infty$, which is to be expected...

Note that for $F \ \to \ 0$ this is the d'Alembert case, which is well known from textbooks on the (lateral) wave equation. There the total inflow is split equally to the downstream and upstream directions, $q_{d_{in_2}} = -q_{d_{in_1}}$.

Step 7 is to calculate the step increase of d_d due to the step increase in flow q_d. This is done for the downstream case by way of the volume balance at time T_1:

$$\begin{aligned}
\Delta V|_{t=T_1} &= \Delta D_1\ B\ L = \Delta Q_1\ T_1 \quad \to \quad \frac{\Delta D_1}{D_d}\ B\ L\ D_d \ = \ \Delta Q_1\ T_1 \\
\frac{\Delta D_1}{D_d} &= \frac{\Delta Q_1}{B\ L\ D_d} \cdot \left(\frac{L}{U_F}\ \frac{1}{1+F}\right) \\
&= \frac{\Delta Q_1}{Q_d}\frac{Q_d}{B\ D_d} \cdot \left(\frac{1}{U_F}\ \frac{1}{1+F}\right) \\
&= \frac{\Delta Q_1}{Q_d}\frac{B\ D_d\ U_d}{B\ D_d} \cdot \left(\frac{1}{U_F}\ \frac{1}{1+F}\right) \\
&= \frac{\Delta Q_1}{Q_d}\frac{U_d}{U_F} \cdot \frac{1}{1+F} \quad = \quad \frac{\Delta Q_1}{Q_d} \cdot F_d \cdot \frac{1}{1+F}
\end{aligned}$$

that is $d_{d_1} = q_{d_1}\dfrac{F_d}{1+F}$; and correspondingly $d_{d_2} = q_{d_2}\dfrac{F_d}{1-F}$

Step 8 Fig.6.2 illustrates the results for $F = 0$ and $F = F_d = 0.40$.

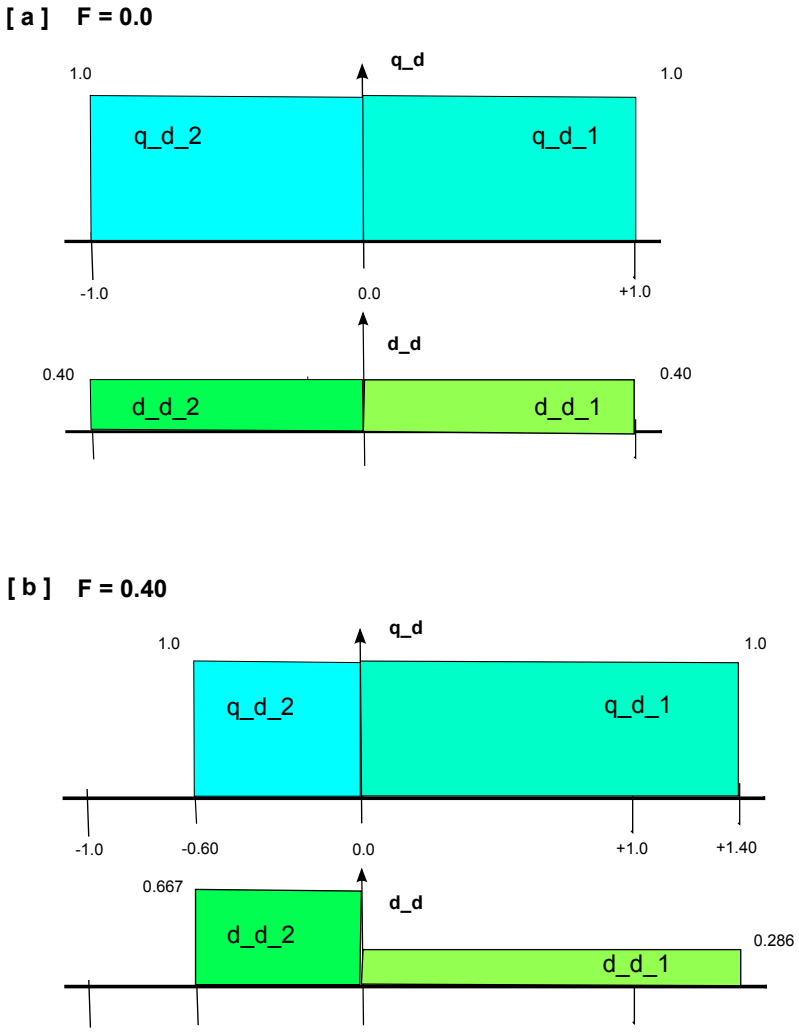


Figure 6.2: Flows and Depths downstream (to the right) and upstream (to the left) snapshot taken at time $t = T_0$
[a] for $F = 0$ (top), [b] for $F = 0.40$ (bottom)

Introducing the boundary conditions at the upper and lower channel end

First the flow and volume content will be investigated, and then the resulting depth changes.

For the **flow** the analysis starts at the upper end, and continues with the flow to the lower end.

At the inflow boundary it is assumed that up to $t = -0$ all flow deviations q_{d_1}, q_{d_2} and the disturbance inflow q_{1_z} are zero. At $t = +0$ the disturbance inflow q_{1_z} changes stepwise, and continues at the same step height for all times. The disturbance inflow is placed at the upper end of the channel. Thus no upstream travelling wave is possible (in contrast to the open-end model of the preceding subsection), and the *full* inflow disturbance will travel downstream. It reaches the lower end of the channel delayed by T_1, that is at time $t = T_1$, now as $q_{1_{out}}$.

Consider first the full flow disturbance being reflected fully (no leakage, no other outflows) and instantaneously (all of it at the same time instant) at the lower end boundary. Then, in the same manner as the upstream flow in the open-end model from above, it will travel upstream as a moving step function, until it reaches the upper end, delayed by T_2, that is at $t = T_1 + T_2 = T_E$ (echo delay time T_E). There it gets reflected fully and instantaneously (same reflection properties here as at the lower end boundary) in the downstream direction. It meets the continuing constant disturbance inflow q_{1_z} and adds itself to it. (It shall have zero back-effect on the inflow). Thus the flow deviation travelling downstream is $2 \cdot q_{1_z}$.

Thus the cycle repeats itself, and after a second echo delay time, the next flow deviation will be of size $3 \cdot q_{1_z}$, and so on. In other words the volume content deviation $v_d(t)$ of the channel will increase by time steps of length T_E and step height $T_E \cdot q_{1_z}$. This is similar to the output of a discrete time integrator module $I(z)$ with a step input. This basic notion will be very helpful to start the systematic design of a feedback controller.

Consider now a nonzero outflow deviation q_{2_u} at the lower boundary, typically from an actuator movement. Then the flow into the upstream leg of the model above is modified to $q_{2_{in}} = q_{1_{out}} - q_{2_u}$
Note the sign convention here: For an increased actuator outflow ($q_{2_u} > 0$) the inflow to the upstream leg is reduced. — Then the argument runs as above. Now the travelling wave delivers $q_{2_{in}}$ delayed by T_2, and as $q_{2_{out}}$ to the upper boundary, where it is added to the inflow deviation $q_{1_{in}}$, and so on...

Next the **depth** changes d_d at the inflow and the outflow boundaries are considered. Both the GMS-friction and the channel bottom slope have been set to zero. Thus the deviations in depth d_{d_1}, d_{d_2} turn into the deviations of surface level, and therefore can be used as 'measured process outputs' y_{d_1}, y_{d_2} for control purposes [4]
Consider the 'industry standard' situation, case [A] from Fig.6.1. The relevant level signal is y_{d_2} at the outflow end. And the level sensor shall be placed a comparatively short distance ΔL upstream of the outflow boundary. Typically this would be $\Delta L \approx B$.
Then the first step input to the sensor is the downstream travelling wave front with deviation d_{d_1} in 'p.u._d'. After a short delay $\Delta L/(U_0 + U)$ the wave front gets reflected instantaneously at the lower boundary wall. There the outflow q_{2_u} gets subtracted. And the resulting flow deviation $q_{2_{in}}$ travels upstream with a level deviation d_{d_2}. This is detected by the level sensor after the additional delay $\Delta L/(U_0 - U)$.
Now let $\Delta L \to 0$. Then the time delays vanish, and the deviation sensor signal y_{d_2} in 'p.u._d' is

$$\begin{aligned} y_{d_2} &= d_{d_1} + d_{d_2} = q_{1_{out}} \cdot \frac{F_d}{1+F} + q_{2_{in}} \cdot \frac{F_d}{1-F} \\ &= q_{1_{out}} \cdot \frac{F_d}{1+F} + [q_{1_{out}} - q_{2_u}] \cdot \frac{F_d}{1-F} \end{aligned}$$

Correspondingly for the sensor y_{d_1} at the inflow end, to be used in situations [B] and [C] from Fig.6.1:

$$\begin{aligned} y_{d_1} &= d_{d_1} + d_{d_2} = q_{1_{in}} \cdot \frac{F_d}{1+F} + q_{2_{out}} \cdot \frac{F_d}{1-F} \\ &= q_{1_{in}} \cdot \frac{F_d}{1+F} + (q_{1_{in}} + q_{1_z}) \cdot \frac{F_d}{1-F} \end{aligned}$$

[4] Note that scaling back into 's.i.'-units is required in practical applications

Illustrating the results by simulation

First the system layout is visualized by signal flow graphs, Fig.6.3, and then by implementation in a Scilab/Xcos diagram, Fig.6.4, with simulation results in Fig.6.5, Fig.6.6

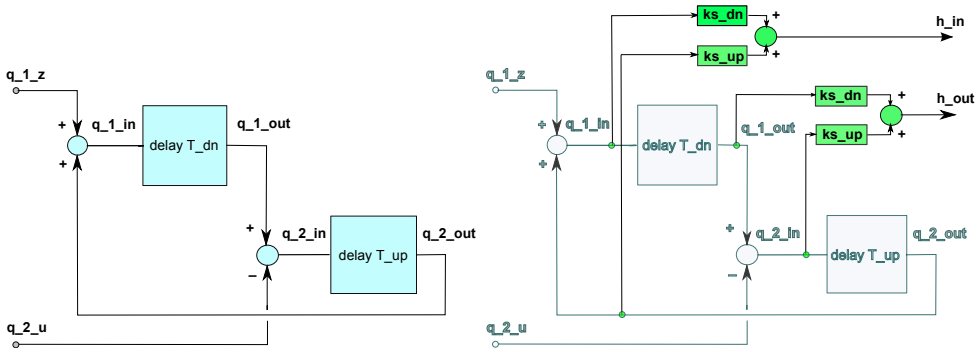


Figure 6.3: Signal flow graphs for flows q_d (left) and with added measured outputs for control y_d (right)

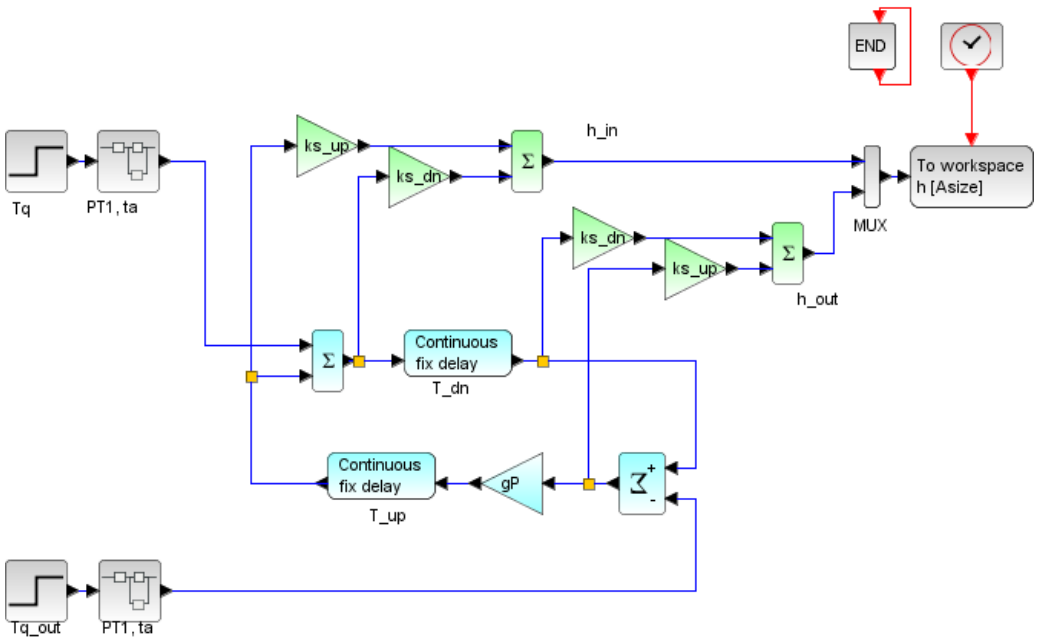


Figure 6.4: Scilab diagram of the process with no control, `s_c6_01_0_01.zcos`

Remarks: Note that this basic layout delivers the measurable depths/levels at both ends of the channel. And that is where the level sensors are placed in nearly all application cases.
If a sensor is placed in an intermediate position X with $0 < X < L$, because the level signal is needed there for control purposes, then the delay blocks in both the downstream and upstream paths must each be broken into two separate delay blocks where distance L is broken into X and $L-X$ and corresponding delay times T_X and $T\_(L-X)$, both for the *_up* and *_dn* paths. The actual level signal at position X then is generated by superposition with coefficients *ks_dn* and *ks_up* (as for the channel end positions). And if further the travelling wave fronts shall be observed over the full length of the channel at intermediate positions $k \cdot \Delta L$ for $k = 0, ..., L/\Delta L$, then the downstream and upstream blocks must be broken into a corresponding number of delay elements with $\Delta T = \Delta L/(U_F \cdot (1 \pm F_d))$[5].

[5]Note the relation to the finite difference solution of the pde (re. upper limit of time step), see [1],[2]

.sce file for ‘context’

```
// s_c6_01_0_1_context.sce
// levels response to inflow & outflow steps
// all control loops open
// Glf   2017.06.15
L = 500;
U_F = 5.0;
F_d = 0.40;

// select Froude number
      U = 2.0;
      U = 0.02;

      F = U/U_F;

T_up = (L/U_F)/(1-F);
T_dn = (L/U_F)/(1+F);
T_E = T_up + T_dn;

ks_dn = F_d/(1+F);
ks_up = F_d/(1-F);

gP = 1.0 - 0.000;

ta = 0.10;

// inflow
// Tq = 1.0; zq0 = 0.0; zq1 = 1.0;
   Tq = 1.0; zq0 = 0.0; zq1 = 0.0;
// outflow
// Tq_out = 1.; qout0 = 0.0; qout1 = 0.0;
   Tq_out = 1.; qout0 = 0.0; qout1 = 1.0;

T_fin = 450;
 CN = 3000;
 delT = T_fin/CN; // readout clock ticks
bufsize = 90000;

// Datatransfer to Plots
CC = 3;     // no of channels 2+1 for time
Asize = 1.01*CC*CN; // size of data arrays
```

.sce file for ‘run’ and ‘plot’

```
// s_c6_01_0_01_crunplot
//  Glf     2017.06.15
//
stacksize('max');
exec('s_c6_01_0_01_context.sce', -1);
importXcosDiagram('s_c6_01_0_01.zcos');
typeof(scs_m); scs_m.props.context;
Info = list();
Info = scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:1,  clf(kfig); end

vcolor = [ 5, 2];

f1 = scf(1);
plot2d(H.time,H.values,vcolor,...
      rect=[0.,-2.4,T_fin,0.1]);
xtitle(['h (t) , response to q_out =',...
msprintf('%3.1f',qout1),...
'and to q_in =',msprintf('%3.1f',zq1),...
' ,   Froude F = ',msprintf('%5.3f',F)]);
  xgrid(1);
legend('h_in', 'h_out',4)
```

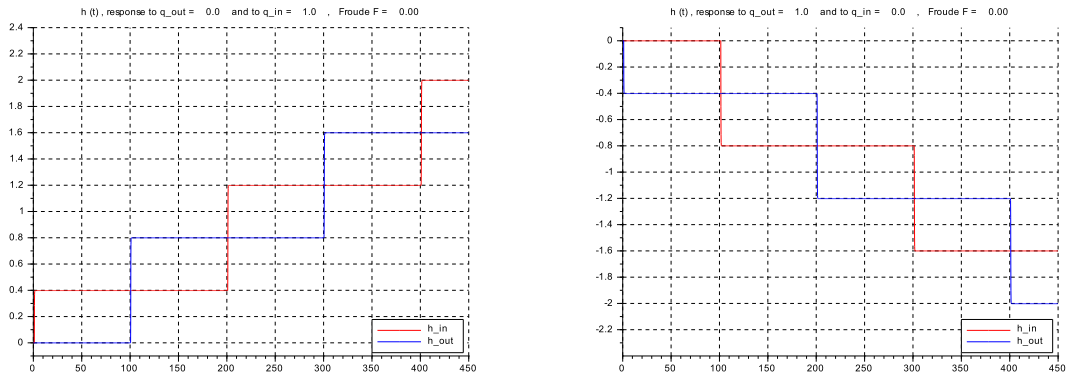


Figure 6.5: level time responses of the controlled process
for $F = 0.004$, to inflow z (left), to outflow u (right)

The delay times from the plots are $T_{dn} = 99.5s$ and $T_{up} = 100.4s$ with $T_E = 200s$, and the step heights are for $h_{in} = 0.3984$ and $h_{out} = 0.8000$, as expected.

For $F = F_d = 0.40$ the delay times from the plots are $T_{dn} = 71.4s$, $T_{up} = 166.7s$ and $T_E = 238.1s$, with step heights for $h_{in} = 0.286$ and $h_{out} = 0.286 + 0.666 = 0.952$, again as expected.

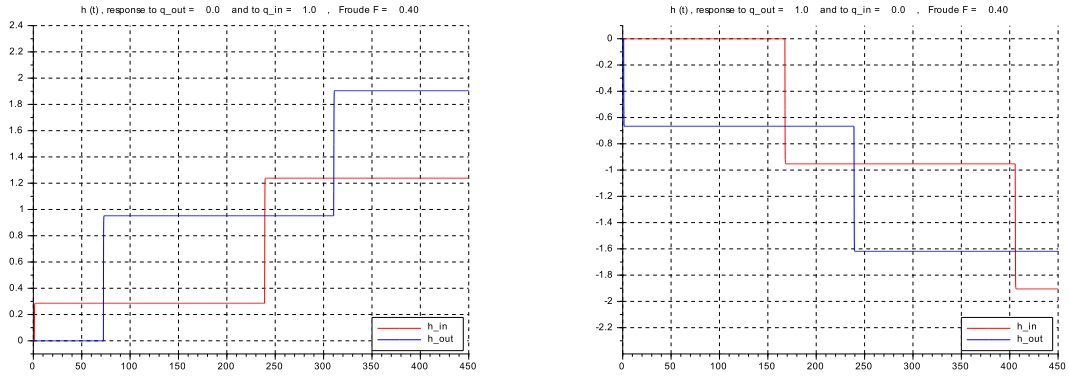


Figure 6.6: level time responses of the controlled process
for $F = F_d = 0.4000$, to inflow z (left), to outflow u (right)

Transfer functions and frequency responses for the plant without controllers

The focus here is on the situation **case [A]**. The delay elements shall be characterised by their transfer functions, $G_{dn}(s), G_{up}(s)$. Then reading from the positive feedback loop in Fig.6.3 yields for the transfer function (TF for short) from the control outflow q_{2_u} to the outflow from the downstream leg $q_{1_{out}}$:

$$q_{1_{out}} = \frac{G_{up} \cdot G_{dn}}{1.0 \ - \ G_{up} \cdot G_{dn}} \cdot (- \ q_{2_u})$$

And for the 'measured output' h_{out}

$$\begin{aligned} h_{out} &= ks_{dn} \cdot q_{1_{out}} + ks_{up} \cdot (q_{1_{out}} - \ q_{2_u}) \\ &= (ks_{dn} + ks_{up}) \cdot q_{1_{out}} + ks_{up} \cdot (- \ q_{2_u}) \end{aligned}$$

that is

$$\begin{aligned} h_{out} &= (ks_{dn} + ks_{up}) \cdot \frac{G_{up} \cdot G_{dn}}{1.0 \ - \ G_{up} \cdot G_{dn}} \cdot (- \ q_{2_u}) + ks_{up} \cdot (- \ q_{2_u}) \\ \rightarrow \frac{h_{out}}{(- \ q_{2_u})} &= (ks_{dn} + ks_{up}) \cdot \frac{G_{up} \cdot G_{dn}}{1.0 \ - \ G_{up} \cdot G_{dn}} + ks_{up} \\ &= (ks_{dn}) \cdot \frac{G_{up} \cdot G_{dn}}{1.0 \ - \ G_{up} \cdot G_{dn}} + (ks_{up}) \cdot \left(\frac{G_{up} \cdot G_{dn}}{1.0 \ - \ G_{up} \cdot G_{dn}} + 1.0 \right) \\ &= (ks_{dn}) \cdot \frac{G_{up} \cdot G_{dn}}{1.0 \ - \ G_{up} \cdot G_{dn}} + (ks_{up}) \cdot \left(\frac{1}{1.0 \ - \ G_{up} \cdot G_{dn}} \right) \\ h_{out} &= \frac{(ks_{dn}) \cdot G_{up} \cdot G_{dn} + (ks_{up})}{1.0 \ - \ G_{up} \cdot G_{dn}} \cdot (- \ q_{2_u}) \end{aligned}$$

Similarly for the TF from the disturbance inflow q_{1_z} to h_{out}:

$$h_{out} = \frac{(ks_{dn} + ks_{up}) \cdot G_{dn}}{1.0 \ - \ G_{dn} \cdot G_{up}} \cdot (+ \ q_{1_z})$$

And finally by superposition:

$$h_{out} = \frac{(ks_{dn}) \cdot G_{up} \cdot G_{dn} + (ks_{up})}{1.0 \ - \ G_{up} \cdot G_{dn}}(- \ q_{2_u}) \quad + \quad \frac{(ks_{dn} + ks_{up}) \cdot G_{dn}}{1.0 \ - \ G_{dn} \cdot G_{up}} \cdot (+ \ q_{1_z})$$

For the situation **case[B]**, the TF's are needed for h_{in} as well. Following the same path as above produces

$$h_{in} = \frac{(ks_{dn}) + (ks_{up}) \cdot G_{dn} \cdot G_{up}}{1.0 \ - \ G_{dn} \cdot G_{up}} \cdot (+ \ q_{1_z}) \quad + \quad \frac{(ks_{dn} + ks_{up}) \cdot G_{up}}{1.0 \ - \ G_{up} \cdot G_{dn}} \cdot (- \ q_{2_u})$$

and the TF for measured output h_{out} is the same as for case [A].

Finally for the situation **case [C]** the inputs must be exchanged while the signs stay the same:

$$q_{1_{z_{[B]}}} \quad \rightarrow \quad q_{1_{u_{[C]}}} \quad \text{and} \quad q_{2_{u_{[B]}}} \quad \rightarrow \quad q_{2_{z_{[C]}}}$$

The result is:

$$\begin{aligned} h_{in} \quad &= \quad \frac{(ks_{dn}) + (ks_{up}) \cdot G_{dn} \cdot G_{up}}{1.0 \ - \ G_{dn} \cdot G_{up}} \cdot (+ \ q_{1_u}) \quad + \quad \frac{(ks_{dn} + ks_{up}) \cdot G_{up}}{1.0 \ - \ G_{up} \cdot G_{dn}} \cdot (- \ q_{2_z}) \\ h_{out} \quad &= \quad \frac{(ks_{dn} + ks_{up}) \cdot G_{dn}}{1.0 \ - \ G_{dn} \cdot G_{up}} \cdot (+ \ q_{1_u}) \quad + \quad \frac{(ks_{dn}) \cdot G_{up} \cdot G_{dn} + (ks_{up})}{1.0 \ - \ G_{up} \cdot G_{dn}}(- \ q_{2_z}) \end{aligned}$$

Discussion

- Inserting the pure delay elements and thus defining G_E and T_E:

  $$G_{dn} \cdot G_{up} = e^{-i \ \omega T_{dn}} \cdot e^{-i \ \omega T_{up}} = e^{-i \ \omega (T_{dn} + T_{up})} \rightarrow G_E = e^{-i \ \omega (T_E)}$$

- The TF's share the common factor G_o

  $$G_o = \frac{1}{1 - G_E} = \frac{1}{1 - e^{-i \ \omega (sT_E)}} = \frac{1}{1 - (cos(\omega T_E) - i \ sin(\omega T_E))}$$

  Points of interest are $\omega T_E = k \cdot \pi \pm \delta(\omega T_E)$ for $k = 0, 1, 2, ...$:

  For $k = 0$

  $$G_o \rightarrow \frac{1}{1 - (cos(\omega T_E) - i \ sin(\omega T_E))} \approx \frac{1}{1 - 1 + i \ \delta(\omega T_E)} = -i \frac{1}{\delta(\omega T_E)}$$

  which is the transfer function of a standard open integrator element.

  For $k = 1, 2, ...$

  $$\begin{aligned} G_o \quad &\rightarrow \quad \frac{(1 - cos(\omega T_E)) + i \ sin(\omega T_E)}{(1 - cos(\omega T_E))^2 + sin^2(\omega T_E)} = \frac{(1 - cos(\omega T_E)) + i \ sin(\omega T_E)}{1 - 2cos(\omega T_E) + cos^2(\omega T_E) + sin^2(\omega T_E)} \\ &= \quad \frac{1}{2} \ \frac{(1 - cos(\omega T_E)) + isin(\omega T_E)}{1 - cos(\omega T_E)} \end{aligned}$$

  For $k = 1 \rightarrow cos(\omega T_E) = -1.0; \quad sin(\omega T_E) = 0..$ Then $G_o = \frac{1+1}{2+2} + i \ 0 = 0.50 + i \ 0.$

  For $k = 2$ set $cos(\omega T_E) \approx +1.0 - 0.5(\delta(\omega T_E))^2; \quad sin(\omega T_E) \approx 0.0 + \delta(\omega T_E)$ then:

  $$G_o\Big|_{k=2} = 0.5 \ (1.0 - i \ \frac{+\omega T_E}{1 - 0.5(\omega T_E)^2}) \quad \rightarrow \quad 0.5 \ (1.0 - i \frac{\delta(\omega T_E)}{1 - 1 + \cdot 0.5 \ \delta(\omega T_E)^2}) = 0.5 \ (1.0 - i \frac{0.5}{\delta(\omega T_E)})$$

At the zero crossover of $\delta(\omega T_E)$ the imaginary part of G_o jumps from $+i\ \infty$ to $-i\ \infty$ and thus connects to the case of $k = 0$.
This generates the lowest resonance. Its frequency is at $\omega_{r_1} T_E = 2\ \pi$. For $F = 0$ with $T_E = 2 \cdot L/U_F = 200\ s$, this is at $\omega_{r_1} = 3.14 \cdot 10^{-2} rad/s$ (to be checked with the results of the following .sce-file...). — Note that the next following resonances will be produced by $k = 4, 6, 8, ...$

For $F = 0.40$, the lowest resonance frequency moves down to $2.640 \cdot 10^{-2}\ rad/s$, due to the increase of T_E from $200s$ to $238s$.

- Consider now the TF from the outflow disturbance input + q_{2_u}:

$$h_{out} = \frac{(ks_{dn}) \cdot G_{up} \cdot G_{dn} + (ks_{up})}{1.0\ -\ G_{up} \cdot G_{dn}} \cdot (-\ q_{2_u})$$

with the additional factor $G_u(s)$:

$$G_u = (ks_{dn}) \cdot G_{up} \cdot G_{dn} + (ks_{up}) = ks_{up} \cdot \left(1 + \frac{ks_{dn}}{ks_{up}} \cdot G_E\right)$$

For $F = 0$, note that $ks_{dn} = ks_{up} = ks = 0.40$ and

$$G_u = ks \cdot (1\ \ +\ \ G_E)$$

For $k = 1, 3, ...$ then $cos(k\ \pi) = -1.0$ and $sin(k\ \pi) = 0.0$

$$G_u = ks \cdot (1\ \ +\ \ cos(\omega T_E) - i\ sin(\omega T_E)) = ks \cdot (1\ \ +\ \ cos(k\ \pi) - i\ sin(k\ \pi)) = 0.0 + i\ 0.0$$

This generates the anti-resonances, at values $k = 1 \rightarrow 1.57 \cdot 10^{-2}\ rad/s$, $k = 3 \rightarrow 4.71 \cdot 10^{-2}\ rad/s$, etc. The gains at the antiresonance frequencies are zero, that is they dip to $-\infty$ $[db]$ in the Bode plot. The phase shifts jump at the antiresonance locations from $-\pi/2$ to $+\pi/2$, with the interleaved phase shifts at the resonance locations jumping back from $+\pi/2$ to $-\pi/2$.

For $F = 0.40$ the lowest anti-resonance moves down to $1.32 \cdot 10^{-2}\ rad/s$. Due to the difference in $ks_{up} > ks_{dn}$ there is now a contribution of the real part > 0 in G_u. In other words the gain spectrum will no longer dip to $-\infty$ at the anti-resonance locations, but stay at a minimum level of $\approx -16\ db$. The detailled calculations are omitted for brevity.

- Consider finally the TF from the inflow disturbance input + q_{1_z}:

$$h_{out} = (ks_{dn} + ks_{up}) \cdot \frac{G_{dn}}{1.0\ -\ G_E} \cdot (+\ q_{1_z})$$

that is

$$G_z = (ks_{dn} + ks_{up}) \cdot G_{dn} = (ks_{dn} + ks_{up}) \cdot (cos(\omega T_{dn}) - i\ sin(\omega T_{dn}))$$

and for $F = 0$

$$G_z = 2 \cdot (ks) \cdot (cos(\omega T_0) - i\ sin(\omega T_0)) \quad \text{with} \quad |G_z| = 2.0 \quad \text{and} \quad arg(G_z) = -\omega \cdot T_0$$

Here the numerator polynomial G_z does not contribute any anti-resonances. Thus the overall gain contains only the resonance spectrum of the basic factor $1/(1-G_E)$. However G_z adds a continuously increasing negative phase shift to the periodic phase plot of $1/(1 - G_E)$.
And the same holds for $F = 0.40$.

Frequency responses for situation case [A], open loop: .sce file

```
// s_c6_01_0_02.sce
// Glf 2017_06_15
// plant with no control,
// output: level at outflow point
// inputs: flow through weir at outflow point
//         additional flow at inflow point

// plant data
//**************
   L = 500;   U_F = 5.0;

// select flow operating point
//    U = 2.00;
      U = 0.02;
      F = U/U_F;   F_d = 0.40;

// Froude wave traveling time
T_dn = L/(U_F)*(1/(1+F)); T_up = L/(U_F)*(1/(1-F));
T_E = T_dn + T_up;   T_0 = 2*L/U_F;

// Froude wave height
ks_dn = F_d/(1 + F); ks_up = F_d/(1 - F);
ks_E = ks_dn + ks_up;

// frequency bounds
vOMmin  = (0.030)/T_0; vOMmax  = (30.0)/T_0;
vOMstep=0.010/T_0; vOM=(vOMmin :vOMstep :vOMmax);

D_dn=%e^(-imult(vOM*T_dn)); D_up=%e^(-imult(vOM*T_up));
D_E  = %e^(-imult(vOM*T_E));

rei = ones(1,length(vOM)); iei = zeros(1,length(vOM));
cei = rei + imult(iei);

// 'zero losses':
kap = 1.0 - 0.0;

// outflow disturbance Frequ Response
//***********************************
oF_u=ks_up.*(cei+(ks_dn/ks_up).*D_E)./(cei-kap.*D_E);
[ophi_u,odb_u] = phasemag(oF_u,"c")

// inflow disturbance Frequ Response
//***********************************
oF_z= + (ks_dn + ks_up).*(D_dn)./(cei - kap.*D_E);
[ophi_z,odb_z] = phasemag(oF_z,"c")

// Bode plots
//**************
f1 = scf(1); clf();
plot2d(vOM,[odb_u',odb_z'],logflag = "ln",style=[2,5],...
       rect=[vOMmin,-40., vOMmax, +40.0]); xgrid(1);
      legend('h_out [db] for input q_2_u',...
             'h_in [db] for input q_2_u',3);

f2 = scf(2); clf();
plot2d(vOM,[ophi_u',ophi_z'],logflag="ln",style=[2,5],...
       rect=[vOMmin,-900., vOMmax, +100.0]); xgrid(1)
      legend('h_out [phi] for input q_2_u', ...
             'h_in [phi] for input q_2_u',3);
```

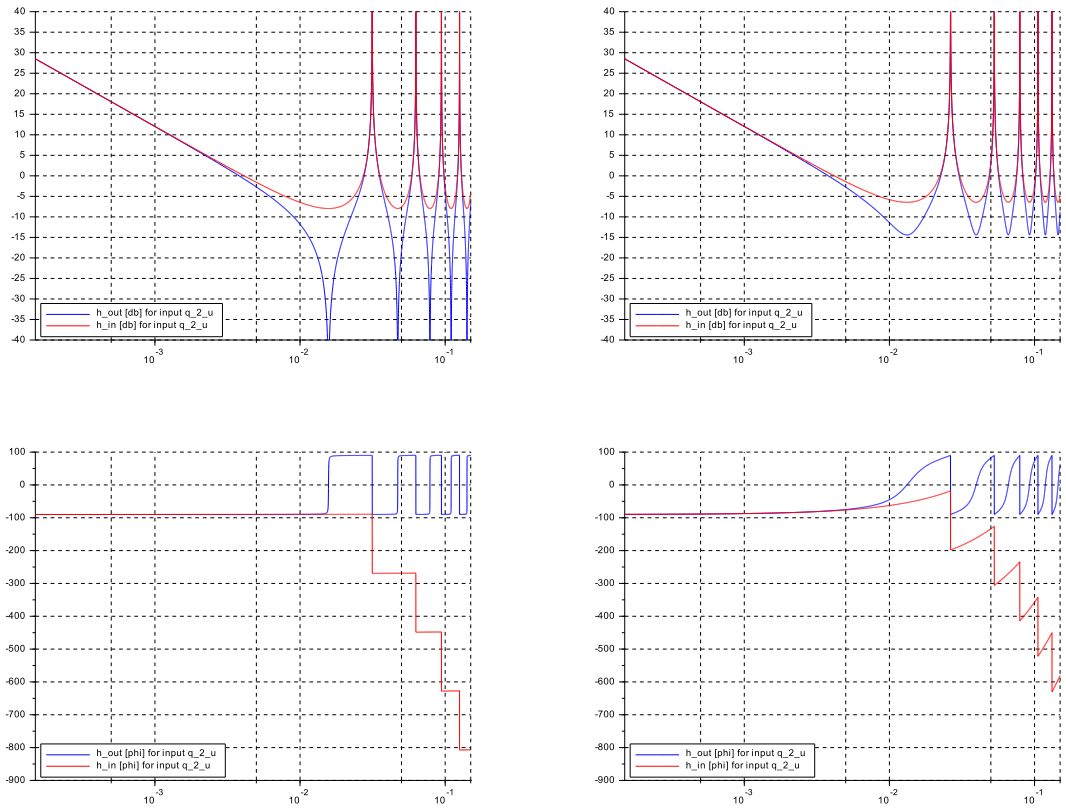


Figure 6.7: Bode plots for h_{out} (blue) and for h_{in} (red) from control outflow, (left) for $F = 0.004$, (right) for $F = F_d = 0.40$

6.3.3 Control Design, Overview

The 'industry standard' situation case [A] is considered first. Initially a P-controller is designed. It is then augmented to a PI-controller. Case [B] is considered next and a PI-controller is designed, and then the same for case [C].

6.3.4 Case [A]: Designing a basic P-controller

Fig.6.8 shows the block diagram of the control system, by extension of Fig.6.3.

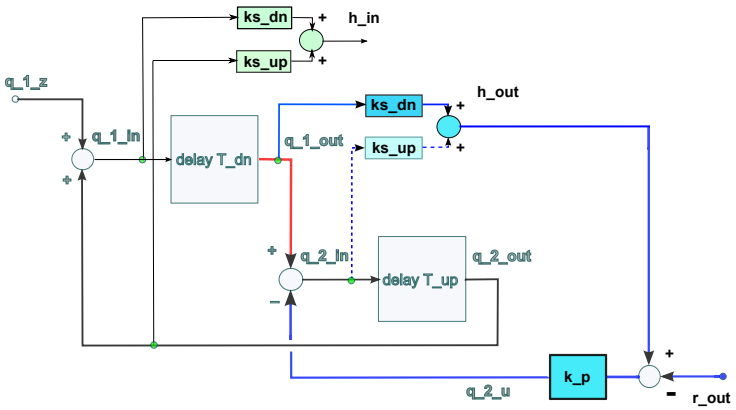


Figure 6.8: Block diagram for P-control path (blue) of h_out by control outflow q_2_u, with controller gain k_p. Notice the internal path (red), see text below

Note that the red path internal to the process with gain +1.0 and the control loop negative feedback with gain $-\ ks\_dn \cdot k\_p$ act in parallel.

They may even *cancel*, for $r\_out = 0.0$ and for

$$+1.0 \ - \ ks\_dn \cdot k\_p \ \stackrel{!}{=} \ 0.0$$

- Then the input path to the T_up-block is zero, and thus the contribution with gain ks_up to the measured value h_out is zero. This is independent of the current value of q_1_out. This is the so-called 'impedance matching' special case, mentioned by Litrico and Fromion in [7].
- The second effect is that the positive feedback loop is cut open. In other words the resonances discussed in the previous subsection are no longer present! And the loop transfer function $1.0/1.0 - G_E$ degenerates into a series connection of the two blocks with T_dn and T_up, that is into G_E.
- and if $r\_out \neq 0$, then the response is $h\_out = G_E \cdot ks\_dn \cdot r\_out$.
- Further note that this effect holds also if the two blocks are no longer pure delays, but may be high order transfer functions (as shall be discussed in the following sections).
- Also note that the relevant process gain ks_dn is a known function of the current operating point $(D,\ Q)$:

  $$ks\_dn = \frac{F_d}{1+F} \quad \rightarrow \quad k\_p := \frac{1+F}{F_d} \quad \text{for the 'impedance matching' case}$$

  which is a neat and compact function for gain scheduling to different operating conditions. And by F_d it covers different design water depths and design flows as well.
- Finally typical values for k_p are well within the span from practical experience at around 4.0, for instance for $F_d = 0.40$, and for actual flow at design flow ($F = F_d$) results in
  $k\_p = \frac{1+F_d}{F_d} = \frac{1.40}{0.40} := 3.5$.

This approach will be investigated in more detail next, starting with the TF and then by simulations.

Transfer function analysis of the P-control loop for case [A]

Control action with P-controller (note the sign convention to obtain negative feedback). The aim is to gain deeper insight into the effect of the individual components.

$$\begin{aligned}
q_2_u &= k_p \cdot (h_out - r_out) \\
\text{that is} \quad h_out &= k_p \cdot \left[-\frac{ks_up + ks_dn\ G_E}{1.0 - G_E} \right] \cdot h_out \\
&+ \quad k_p \cdot \left[+\frac{ks_up + ks_dn\ G_E}{1.0 - G_E} \right] \cdot r_out \\
&+ \quad (ks_dn + ks_up)\ \frac{G_{dn}}{1.0 - G_E} \cdot q_1_z \\
h_out \cdot \left[1.0 + k_p \cdot \frac{ks_up + ks_dn\ G_E}{1.0 - G_E} \right] &= k_p \cdot \left[+\frac{ks_up + ks_dn\ G_E}{1.0 - G_E} \right] \cdot r_out \\
&+ \quad (ks_dn + ks_up)\ \frac{G_{dn}}{1.0 - G_E} \cdot q_1_z \\
h_out \cdot [1.0 - G_E + k_p \cdot ks_up + k_p \cdot ks_dn\ G_E] &= (k_p\ ks_up)_up \left[1.0 + \frac{ks_dn}{ks_dn}\ G_E \right] \cdot r_out \\
&+ [(ks_dn + ks_up)\ G_{dn}] \cdot q_1_z \\
h_out \cdot [(1.0 + k_p \cdot ks_up) - (1.0 - k_p \cdot ks_dn)\ G_E] &= (k_p\ ks_up)_up \left[1.0 + \frac{ks_dn}{ks_dn}\ G_E \right] \cdot r_out \\
&+ [(ks_dn + ks_up)\ G_{dn}] \cdot q_1_z
\end{aligned}$$

finally

$$h\_out = \frac{k\_p\ ks\_up}{1.0 + k\_p\ ks\_up} \cdot \frac{1.0 + \frac{ks\_dn}{ks\_dn}\ G_E}{\left[1 - \frac{1.0 - k\_p\ ks\_dn}{1.0 + k\_p\ ks\_up}\ G_E \right]} \cdot r\_out + \frac{ks\_dn + ks\_up}{1.0 + k\_p\ ks\_up} \cdot \frac{G_{dn}}{\left[1 - \frac{1.0 - k\_p\ ks\_dn}{1.0 + k\_p\ ks\_up}\ G_E \right]} \cdot q\_1\_z$$

Discussion

* the common factor is now

$$G_c = \frac{1.0}{\left[1.0 - \frac{1.0 - k\_p\ ks\_dn}{1.0 + k\_p\ ks\_up}\ G_E \right]}$$

- for $k\_p = 0$ again

$$G_c = \frac{1.0}{\left[1.0 - \frac{1.0 - 0.\ ks\_dn}{1.0 + 0.\ ks\_up}\ G_E \right]} \rightarrow \frac{1.0}{1.0 - G_E}$$

- for $0 < k\_p < 1.0/ks\_dn$ that is $k\_p \cdot ks\_dn = \varepsilon$ with $0 < \varepsilon \ll 1.0$

$$G_c = \frac{1.0}{\left[1.0 - \frac{1.0 - \varepsilon}{1.0 + \varepsilon\ \frac{ks\_up}{ks\_dn}}\ G_E \right]} \approx \frac{1.0}{1.0 - (1.0 - \varepsilon)\ G_E}$$

  This can be interpreted as a small leakage at the lower channel end. It reduces the resonance peaks from $+\infty$ (this can be seen by the approach from above at frequencies $\omega T_E = k \cdot \pi$ with $k = 2, 4, ...$).

- for $k\_p = 1.0/ks\_dn$

$$G_c = \frac{1.0}{\left[1.0 - \frac{1.0 - 1.0}{1.0 + 1.0\ \frac{ks\_up}{ks\_dn}}\ G_E \right]} \rightarrow \frac{1.0}{1.0 - (1.0 - 1.0)\ G_E} = 1.0$$

  that is the 'resonance generator' from the denominator is suppressed indeed.

- for $k\_p > 1.0/ks\_dn$

$$G_c = \frac{1.0}{\left[1.0 + \frac{k\_p \cdot ks\_dn - 1.0}{k\_p \cdot ks\_up + 1.0}\ G_E\right]} = \frac{1.0}{1.0 + \varepsilon\ G_E}$$

Thus the resonance peaks stay suppressed for increasing k_p. Note that for increasing F-values the weight $k\_p \cdot ks\_up$ of G_E decreases, and thus decreases the contribution of G_E in the denominator of G_c, which decreases the resonance peaks further. – Also G_c in its last form may be seen as a the typical TF of a closed loop with εG_E is the TF of the open loop. Thus stability and damping of the closed loop may be investigated on the Nyquist plot (gain and phase margin). Let $G_E := exp(-i\omega T_E)$ then its Nyquist contour is a circle centered at the origin with radius $\varepsilon \cdot 1.0$.

The table collects the numerical results for $G\_open = \varepsilon \cdot 1.0$ at $\omega T_E = \pi$ for three values of F:

$k\_p\|_{F=0}$	2.5	4.0	6.0	8.0	∞
$k\_p \cdot ks\_dn$	1.0	1.60	2.40	3.20	∞
$F = 0$	0.0	−0.2308	−0.4118	−0.5238	−1.0
$F = 0.04$	0.0	−0.2195	−0.3889	−0.4925	−1.0
$F = F_d = 0.40$	0.0	−0.1268	−0.2121	−0.2598	−1.0

Thus the oscillatory stability limit is at $k\_p \to \infty$. Also the value for G_open approx. doubles for $F = 0 \to F = 0.40$. A practical limit to avoid an excessive oscillatory response would be at $|G\_open| \leq 0.25$, that is at very low flow conditions $k\_p \cdot ks\_dn \leq 1.60$ and at design flow $F = F_d = 0.40$ $k\_p \cdot ks\_dn \leq 3.0$.[6]

* The numerator polynomials for both the reference input r_out and the disturbance input q_1_z are the same as in the open loop case. Thus the anti-resonances in the reference response are not modified by the gain k_p, which is to be expected from the general case of control loop with zeros in the numerator. Also the disturbance response produces no anti-resonances, it is the open-loop TF G_E.

* And the gain factors

for the reference response $\dfrac{k\_p\ ks\_up}{1.0 + k\_p\ ks\_up}$ and the disturbance response $\dfrac{ks\_up + ks\_dn}{1.0 + k\_p\ ks\_up}$

are changed by k_p in the typical manner for such P-control loops. Note that here ks_up plays the dominant role.

* Finally note that these conclusions are also valid for the open-loop transfer functions $G_i(s)$ from the high order compartment models (see the following sections).

Frequency responses

This is to confirm and visualise the results of the analysis above, using the pure delay elements in the process model.

.sce file

```
// s_c6_01_1_01
// Glf 2017_06_15
//  case [A]: plant with P-controller
// controlled variable: outflow level
// control variable:  flow at outflow point
// inputs: level reference  & inflow


for kFig = 1:1:2, clf(kFig); end
// plant data
//**************
   L = 500;
   U_F = 5.0;

// select flow operating point
      U = 2.00;
//    U = 0.20;
      U = 0.02;
      F = U/U_F;   F_d = 0.40;

//Froude wave traveling time
T_dn = L/(U_F)*(1/(1+F));
T_up = L/(U_F)*(1/(1-F));
T_E = T_dn + T_up; T_0 = 2*L/U_F;
```

[6] Note that the typical Ziegler-Nichols rule of $|G\_open| = 0.50$ assumes a different form of the plant Nyquist contour !

```
// Froude wave height
ks_dn = F_d/(1 + F);
ks_up = F_d/(1 - F);
ks_E = ks_dn + ks_up;

// select P-controller gain kp
//***************************
   kp = 0.0600*(1/F_d)*(1.0+F);
// kp = 1.000*(1/F_d)*(1.0+F);
   kp = 1.600*(1/F_d)*(1.0+F);

// frequency bounds
vOMmin  = (0.40)/T_0;
vOMmax  = (40.0)/T_0;
vOMstep = 0.005/T_0;
vOM = (vOMmin : vOMstep : vOMmax);

D_dn = %e^(-imult(vOM*T_dn));
D_up = %e^(-imult(vOM*T_up));
D_E  = %e^(-imult(vOM*T_E));

rei = ones(1,length(vOM));
iei = zeros(1,length(vOM));
cei = rei + imult(iei);

// 'zero losses':
kap_E = 1.0 - 0.00;
```

```
// closed loop
gs_dn = (ks_dn*kp); gs_up = (ks_up*kp);
Np = (cei + (gs_dn/gs_up)*kap_E.*D_E);
Dp = (cei-((1.0-gs_dn)/(1.0+gs_up))*kap_E.*D_E);

cF_r = (gs_up/(1+gs_up))*Np./Dp;
cF_z = (ks_E/(1 + gs_up))*D_dn./Dp;

// Bode
[cphi_r,cdb_r] = phasemag(cF_r,"c");
[cphi_z,cdb_z] = phasemag(cF_z,"c");

f1 = scf(1);
plot2d(vOM,[cdb_r',cdb_z'],logflag = "ln",style=[2,5],...
        rect=[vOMmin,-60., vOMmax, +20.0]); xgrid(1);
    legend('h_out [db] from reference input r_2_u',...
         'h_out [db] from inflow q_1_z',3);

f2 = scf(2);
plot2d(vOM,[cphi_r',cphi_z'],logflag="ln",style=[2,5],...
       rect=[vOMmin,-800.,vOMmax, +100.0]); xgrid(1);
    legend('h_out [phi] from reference input r_2_u',...
         'h_out [phi] from inflow q_1_z',3);
```

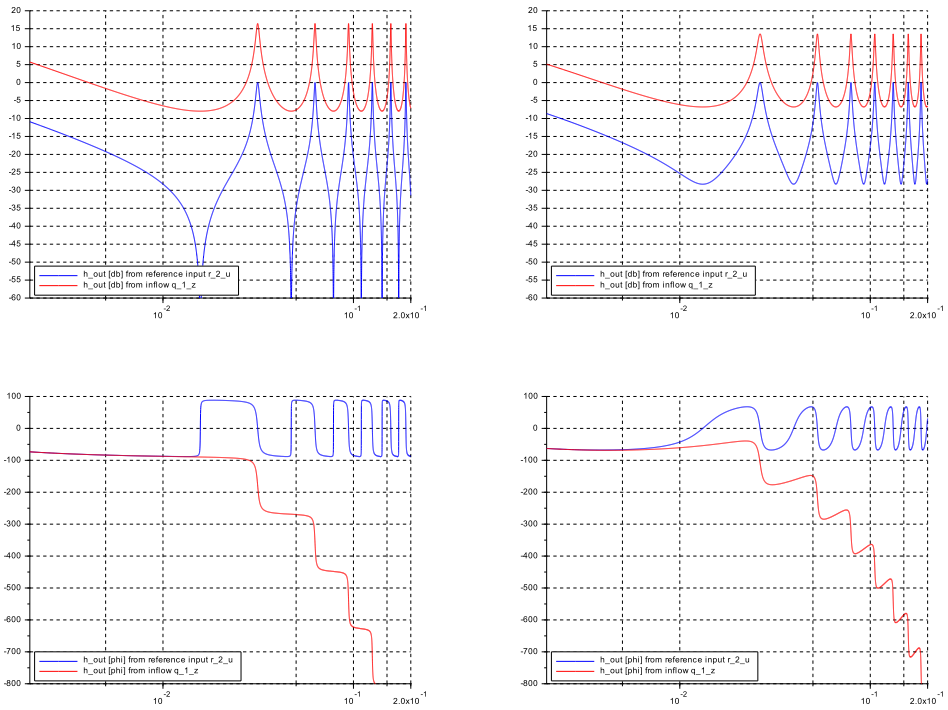


Figure 6.9: Bode plots for P-control of of h_{out} (blue) and for h_{in} (red) from control outflow, (left) for $F = 0.004$, (right) for $F = F_d = 0.40$
small leakage at outflow, simulated by $k\_p \cdot ks\_dn = 0.060$

Discussion
Compared to Fig.6.7, Fig.6.9 shows the reduction of the resonance peaks of the disturbance response the due to the small leakage at the outflow.
And Fig.6.10 documents the further reduction of the resonance peaks in the reference response. The gain of the disturbance response is flat for the 'impedance matching' case at -7.96 *db*.

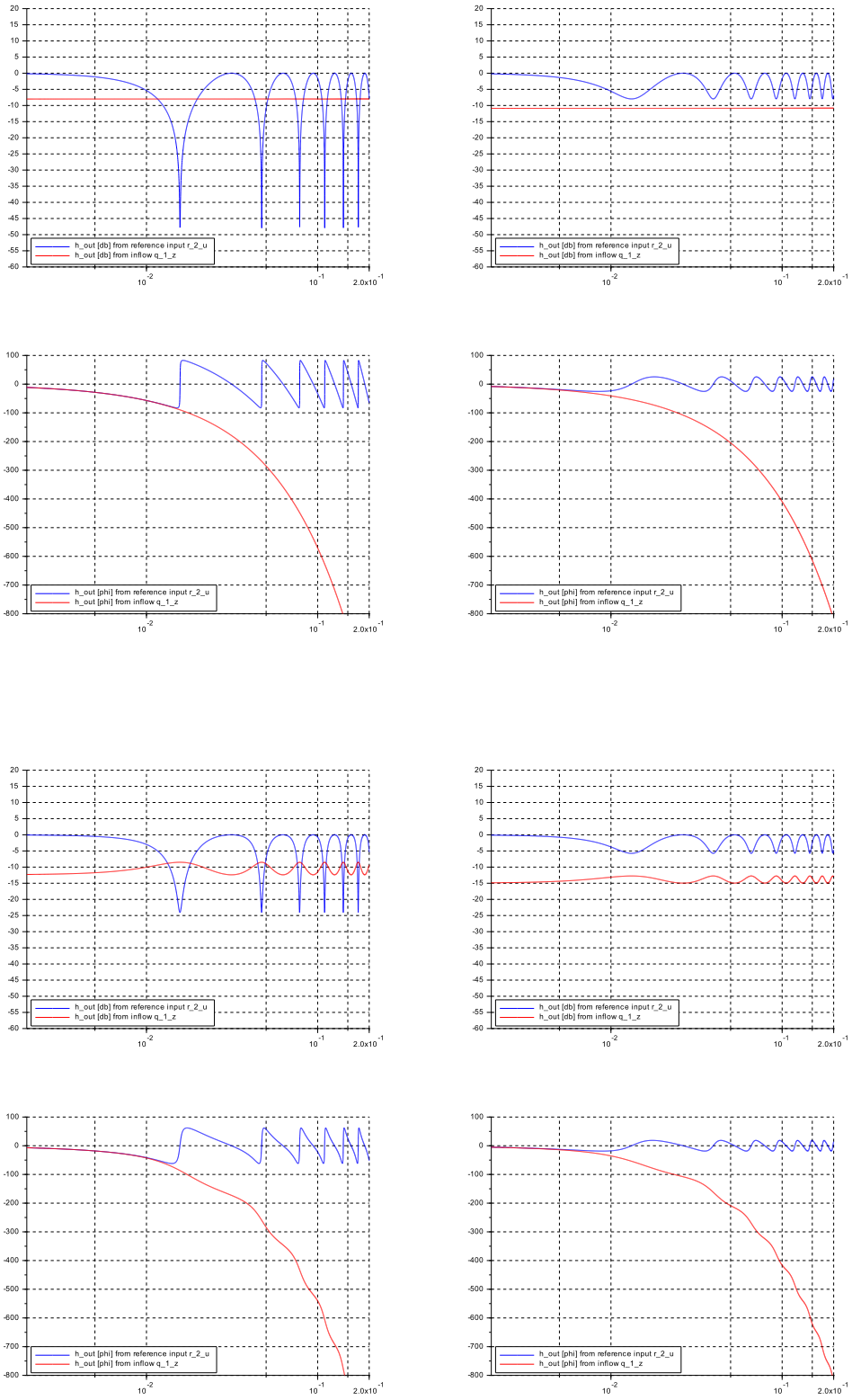


Figure 6.10: Bode plots for P-control of h_{out} (blue) and for h_{in} (red) from control outflow,
(left) for $F = 0.040$, (right) for $F = F_d = 0.40$
(top) $k\_p \cdot ks\_dn = 1.00$ 'impedance matching', (bottom) $k\_p \cdot ks\_dn = 1.60$

Simulation results

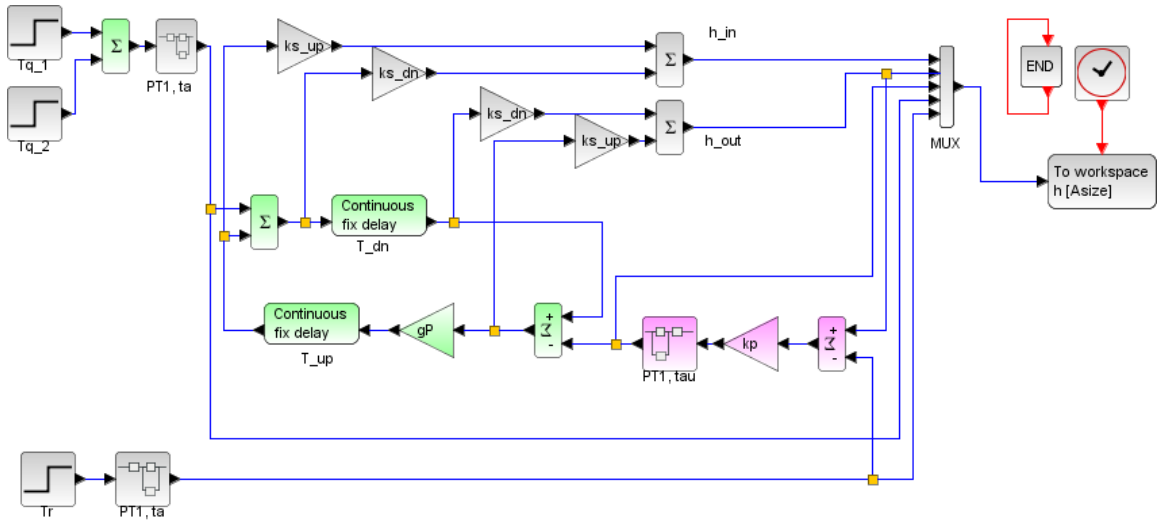


Figure 6.11: Case [A]: Scilab diagram of the P-controlled loop

.sce file for 'context'

```
// s_c6_01_1_02_context.sce
// case [A]
// inflow disturbance step; utflow level reference step
// P-controller for outflow level on outflow actuator
// Glf   2016_10_12

L = 500;
U_F = 5.0;

// select Froude number
//    U = 2.0;
      U = 0.20;

      F_d = 0.40; F = U/U_F;

// delay times
T_up = (L/U_F)/(1-F);
T_dn = (L/U_F)/(1+F);
T_E = T_up + T_dn;
bufsize = 1000*T_E/2;

// no losses
gP = 1.0 - 0.00;

// time constant on inputs; on actuator
ta = 2.0;   tau = 0.1*ta;

// Froude wave heights
ks_dn = F_d/(1+F);
ks_up = F_d/(1-F);
ks_r  = F_d;

// reference step
Tr = 2000.; rh0 = 0.0; rh1 = 1.0*ks_r;

// inflow steps
Tq_1 =   10.0; zq0 = 0.0; zq1 =  1.0;
Tq_2 = 1000.0; zq0 = 0.0; zq2 = -1.0;

// P-controller for out-level on outflow
      kp =   1.00*(1/F_d)*(1+F); // imped.match.
//    kp =   1.60*(1/F_d)*(1+F);

T_fin = 4000;
 CN = 8000;
 delT = T_fin/CN;//readout clock ticks


// Datatransfer to Plots
CC = 6;    // no of channels 5 + 1 for time
Asize = 1.01*CC*CN; // size of data arrays
```

.sce file for 'run' and 'plot'

```
// s_c6_01_1_02_crunplot
//  Glf  2017_06_15
//
stacksize('max');
exec('s_c6_01_1_02_context.sce', -1);
importXcosDiagram('s_c6_01_1_02.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:1,  clf(kfig); end

vcolor = [ 5, 2, 1, 11, 13];

f1 = scf(1);
plot2d(H.time,H.values,vcolor,rect=[0.,-0.80,T_fin,1.40]);
xtitle(['case 1:, P-control, kp =',msprintf('%5.2f',kp),...
'; inflow steps up and down, reference step up,   F =',...
msprintf('%5.3f',F)]); xgrid(1);
  legend("h_in","h_out","q_2_u","q_1_z","r_out",4)
```

Discussion

The impedance matching performs as expected for both low flow and design flow conditions, with a 'dead beat' response. Also the gain scheduling of k_p as function of F performs as expected.
Note the resulting k_p-values given in the plot headings. They are within the usual range.

And on the disturbance response the overshoot on h_out is $\approx 11\%$ for $k\_p \cdot ks\_dn = 1.60$ for both $F = 0.04$ and 0.40. Thus if the value $k\_p \approx 2.5$ from $k\_p \cdot ks\_dn = 1.0$ for $F = 0.040$ is considered too low, then a change to $k\_p = 4.0$ from $k\_p \cdot ks\_dn = 1.60$ may be considered.

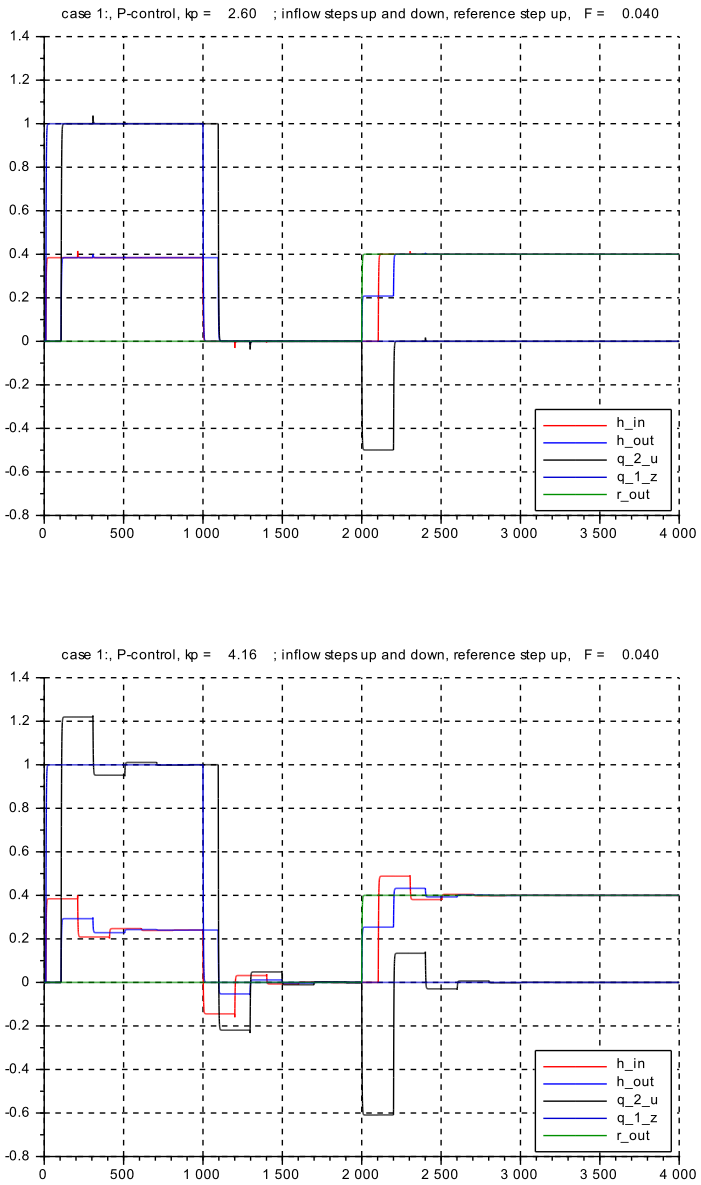


Figure 6.12: outflow level time responses of the P-controlled process, for $\mathbf{F = 0.04}$ (top) with $k\_p \cdot ks\_dn = 1.00$, (bottom) with $k\_p \cdot ks\_dn = 1.60$

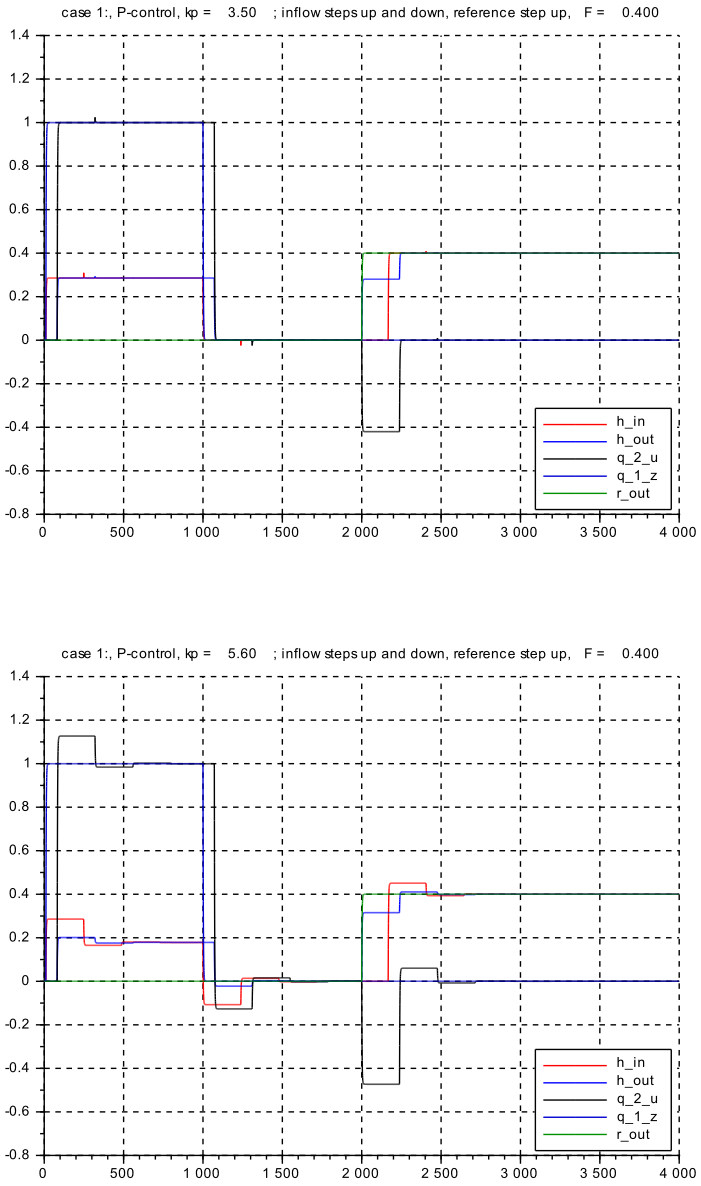


Figure 6.13: outflow level time responses of the P-controlled process, for $\mathbf{F = 0.40}$ (top) with $k\_p \cdot ks\_dn = 1.00$, (bottom) with $k\_p \cdot ks\_dn = 1.60$

6.3.5 Case [A]: Augmenting the P-control to PI-control

The control structure

The integral action is implemented here in a cascaded structure, see Fig.6.14, by extension of Fig.6.8.

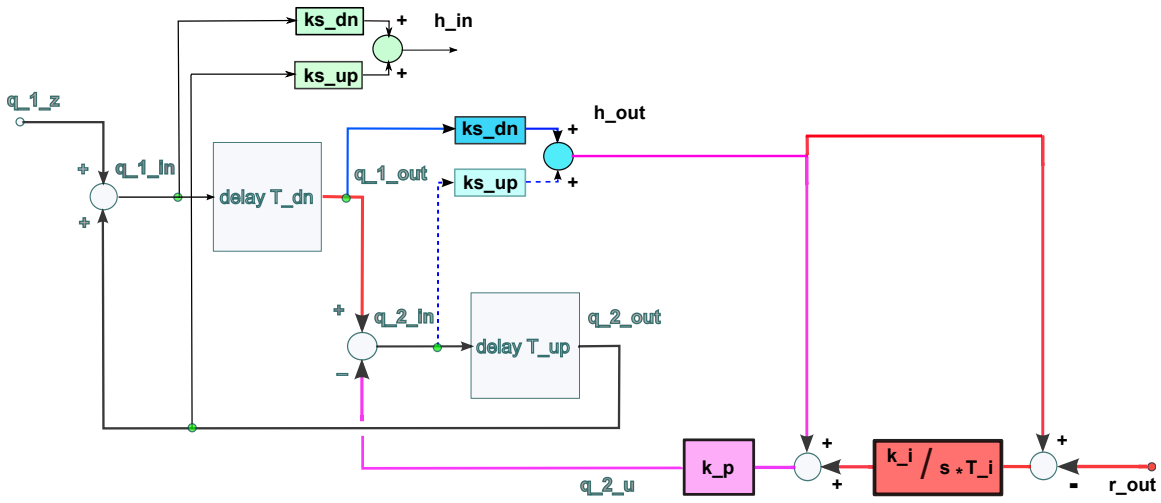


Figure 6.14: Block diagram for the cascaded PI-control, designated as ***PcI***
'master' controller with I-action (red), 'slave' controller with P-action (pink)
'master' controller gain k_i/T_i, 'slave' controller gain k_p.

The reasons for this roundabout approach are

- This structure can be used directly for cases [B] and [C], where the measured inputs for master and slave controllers are not the same ones.

- The property of 'impedance matching' is conserved, and also the basic gain-scheduling rules from there.

- The Transfer Function (TF) can be built in steps, using as the first step the previously derived ones for the closed slave loop, and then multiplying by the master controller $S_r = k\_i/(s \cdot T\_i)$ to create the open loop TF for the overall system, and from there to the TF of the closed loop by using the basic formulas for feedback loops.

- Further this approach produces directly the 2dof-structure to the reference inputs[7].

- Note that the extra parameter T_i allows the integral action to be time-scaled to the dominant time parameter of the plant. In our case this would be the parameter T_E from the dominant dynamics $G\_E(s)$, that is setting $T\_i := T\_E$.
  And then the dosage of the integral action effect is produced by its gain k_i.

- Finally the gains kp_p, ki_p for the parallel PI-controller $S_r = kp_p + ki_p/(sTi_p)$ structure can be directly evaluated (set $Ti_p := T\_i$), and from there the gains kp_s, Tn_s of the standard 1dof-form $S_r = kp_s(1.0 + 1/(sTn_s))$.

[7] highly recommended in this application area to avoid excessive flow surges

Transfer functions

From the previous subsection

$$\begin{aligned} y_{out} &= \lambda_r \frac{1+\mu G_E}{1+\varkappa G_E} \cdot r_{2_s} \quad + \lambda_z \frac{G_{dn}}{1+\varkappa G_E} \\ \text{where} \quad \lambda_r &= \frac{k_p k s_{up}}{1+k_p k s_{up}} \\ \mu &= \frac{k s_{dn}}{k s_{up}} \\ \varkappa &= \frac{k_p k s_{dn}-1}{k_p k s_{up}+1} \\ \lambda_z &= \frac{k s_{dn}}{1+k_p k s_{up}} \end{aligned}$$

Inserting the master controller $u_{2_m} := (k_i/(sT_i)) \cdot (r_{2_m} - y_{out})$, where $r_{2_s} \to u_{2_m}$:

$$y_{out} = \lambda_r \frac{1+\mu G_E}{1+\varkappa G_E} \cdot \frac{k_i}{sT_i} \cdot r_{2_m} \quad - \quad \lambda_r \frac{1+\mu G_E}{1+\varkappa G_E} \cdot \frac{k_i}{sT_i} \cdot y_{out} \quad + \quad \lambda_z \frac{G_{dn}}{1+\varkappa G_E} \cdot z_1$$

after some algebra

$$\begin{aligned} y_{out} &= \frac{1}{s\Theta_m+1} \cdot r_{2_m} \; + \; \frac{\lambda_z G_{dn}}{\lambda_r(1+\mu G_E)} \cdot \frac{1}{s\Theta_m+1} \cdot s\frac{T_i}{k_i} \cdot z_1 \\ \text{with 'time constant' } \Theta_m\text{:} \quad \Theta_m &= \frac{T_i}{k_i} \frac{(1+\varkappa G_E)}{\lambda_r(1+\mu G_E)} \end{aligned}$$

Design procedure for the controller parameters

Consider first

1. the special case of 'impedance matching' $k_p k s_{dn} := 1.0$
2. at very low flow $F \to 0$
3. and also at very low frequencies $\omega \to 0$

Then

$$\begin{aligned} \text{from 3.} \quad G_E &\to 1.0 \quad \text{and} \quad G_{dn} \to 1.0 \\ \text{from 2.} \quad \mu &= \frac{k s_{dn}}{k s_{up}} \to 1.0 \\ \text{from 1.} \quad \varkappa &= k_p k s_{dn} - 1 \to 0.0 \\ \text{Also} \quad \lambda_r &= \frac{k_p k s_{up}}{1+k_p k s_{up}} = \frac{(k_p k s_{dn})\frac{k s_{up}}{k s_{dn}}}{1+(k_p k s_{dn})\frac{k s_{up}}{k s_{dn}}} \to \frac{1.0}{1+1.0} = \frac{1}{2} \\ \text{and} \quad \Theta_m &\to \frac{T_i}{k_i}\frac{1.0}{\lambda_r(1+1)} \to \frac{T_i}{k_i} \\ \text{and also} \quad \lambda_z &= \frac{k s_{dn}}{1+(k_p k s_{dn})\frac{k s_{up}}{k s_{dn}}} \to \frac{k s_{dn}}{2} \end{aligned}$$

$$\begin{aligned} \text{Thus} \quad y_{out} &= \frac{1}{s\frac{T_i}{k_i}+1} \cdot r_{2_m} \; + \; \frac{0.5 k s_{dn} \cdot 1.0}{0.5 \cdot 2.0} \cdot \frac{1}{s\frac{T_i}{k_i}+1} \cdot s\frac{T_i}{k_i} \cdot z_1 \\ \text{finally} \quad y_{out} &= \frac{1}{s\frac{T_i}{k_i}+1} \cdot r_{2_m} \; + \; 0.5 k s_{dn} \cdot \frac{1}{s\frac{T_i}{k_i}+1} \cdot s\frac{T_i}{k_i} \cdot z_1 \end{aligned}$$

Consider next relaxing item 2, $F \rightarrow \geq 0$. Then from the modelling section

$$ks_{dn} = \frac{F_d}{1+F} \quad \text{and} \quad ks_{up} = \frac{F_d}{1-F}$$

$$\lambda_r = \frac{(k_p k s_{dn})\frac{1+F}{1-F}}{1+(k_p k s_{dn})\frac{1+F}{1-F}} \quad \rightarrow \quad \frac{(1.0)\frac{1+F}{1-F}}{1+(1.0)\frac{1+F}{1-F}} = \frac{1+F}{(1-F)+(1+F)} = \frac{1+F}{2}$$

All other elements do not change. Thus for the denominator of the transfer functions

$$s\Theta + 1 = s\frac{T_i}{k_i}\frac{1}{\lambda_r(1+G_E)} \quad \rightarrow \quad s\frac{T_i}{k_i \cdot (1+F)} \cdot \frac{1}{0.5(1+1)} + 1 = s\frac{T_i}{k_i \cdot (1+F)}$$

In other words the gain of the integral action master controller is to be gain-scheduled by $k_i \cdot (1+F)$. – Note the relation to the gain-scheduling of the proportional gain in the 'slave' loop.

Consider further relaxing only item 3, that is $0 \leq \omega \cdot T_E \ll \pi$ at $F \rightarrow 0$ and for $k_p k s_{dn} = 1$. Then the characteristic equation of the denominator is

$$0 = s\frac{T_i}{k_i} + 0.5 \cdot (1+G_E)$$

The Nyquist plot of the second term is a circle starting at $+1.0 + j0$ with center at $0.50 + j0$ and running through the origin $0 + j0$ at $\omega T_E = \pi$. It can be approximated by the rational TF G_f

$$G_f = \frac{k_f}{1+sT_f} \quad \text{with} \quad k_f = 1.0 \quad \text{and} \quad T_f \quad \text{to be determined next}$$

The phase shift for the original is $-45°$ at $\omega T_E = \pi/2$, whereas the approximation has the identical phase shift at $\omega T_f = 1.0$. Therefore select

$$T_f := \frac{2}{\pi} T_E$$

Then back to the characteristic equation with the approximation:

$$0 = s\frac{T_i}{k_i} + \frac{1}{1+T_f} \quad = \quad sT_i(1+sT_f) + k_i \quad = \quad s^2 + s\frac{1}{T_f} + \frac{k_i}{T_i T_f} \quad = \quad s^2 + s\frac{\pi}{2T_E} + \frac{\pi k_i}{2T_i T_E}$$

As mentioned before, a reasonable choice for the time scaling of the integral action is the dominant time parameter of the plant, that is

$$T_i := T_E$$

Thus

$$0 = s^2 + s\frac{\pi}{2T_E} + \frac{\pi}{2T_E^2} \cdot k_i$$

to be compared to the standard form of a damped second order element with parameters $2D$ and Ω:

$$0 = s^2 + 2Ds\Omega + \Omega^2$$

Equating coefficients for s^0 yields:

$$(\Omega T_E)^2 = \frac{\pi}{2} k_i \quad \rightarrow \quad \Omega T_E = \sqrt{\frac{\pi}{2} k_i}$$

and for s^1:

$$2D = \frac{\pi}{2}\frac{1}{\Omega T_E} \quad \rightarrow \quad 2D = \frac{\pi}{2}\frac{1}{\sqrt{\frac{\pi}{2} k_i}} = \sqrt{\frac{\pi}{2}\frac{1}{k_i}}$$

Set $2D := \sqrt{2}$ to obtain a maximally flat Bode gain plot (or a $\approx 5\%$ overshoot of the step response) produces

$$k_i = \frac{\pi}{4} \approx 0.785 \quad \text{and} \quad \Omega T_E = \frac{\pi}{\sqrt{8}} \approx 1.111$$

Consider also relaxing items 3 and 2 by letting both $0 < \omega T_E < \pi$ and $F \rightarrow F_d$, while keeping item 1 at $k_p k s_{dn} = 1.0$.
The characteristic equation then modifies to:

$$\begin{aligned}
0 &= sT_i \frac{1}{\frac{ks_{up}+ks_{dn}G_E}{ks_{up}+ks_{dn}}} + k_i \quad \text{with} \quad ks_{up}, ks_{dn} \quad \text{as above} \\
&= sT_i \frac{1}{F_d + \frac{1-F_d}{1+sT_f}} + k_i \\
&= sT_i \frac{1+sT_f}{F_d(1+sT_f)+(1-F_d)} + k_i \\
&= s^2 T_i T_f + sT_i + sT_f F_d k_i + k_i \\
&= s^2 + s\left[\frac{1}{T_f} + F_d k_i \frac{1}{T_i}\right] + k_i \frac{1}{T_i T_f}
\end{aligned}$$

Then using $T_f = T_E(2/\pi)$ and $k_i = \pi/4$ from above

$$\begin{aligned}
\Omega T_E &= \frac{\pi}{\sqrt{8}} \\
\text{and} \quad 2D &= \frac{1}{\Omega T_f} + F_d k_i \frac{1}{\Omega T_i} = \frac{\pi}{2}\frac{1}{\Omega T_E} + F_d \frac{\pi}{4}\frac{1}{\Omega T_E} = \frac{\pi}{2}\frac{1}{\Omega T_E}\left[1 + \frac{F_d}{2}\right] \\
2D &= 0.6 \cdot \sqrt{8} \approx 1.70 \quad \text{for} \quad F_d = 0.40
\end{aligned}$$

Consider finally relaxing item 1 to $k_p k s_{dn} = 1.60 > 1.0$ while setting item 2 to $F \rightarrow 0$, but allowing $0 < \omega T_E \ll \pi$. The characteristic equation modifies to

$$\begin{aligned}
0 &= s\frac{T_i}{k_i}\frac{1+\varkappa G_E}{\lambda_r(1+\mu G_E)} + 1 \quad \text{with} \quad \varkappa = \frac{k_p k s_{dn} - 1}{k_p k s_{dn} + 1} = \frac{0.60}{2.60} = 0.231 \\
\text{with} \quad \lambda_r &= 0.50 \quad \text{and} \quad \mu = 1.0 \\
\text{approximating} \quad 1 + \varkappa G_E &\rightarrow (1-\varkappa) + 2\varkappa \frac{1}{1+sT_f} \\
\text{inserting} \quad 0 &= sT_i \frac{\frac{(1-\varkappa)(sT_f+1)+2\varkappa}{1+sT_f}}{\frac{1}{1+sT_f}} + k_i \\
&= sT_i\left[(1-\varkappa)(1+sT_f) + 2\varkappa\right] + k_i \\
&= sT_i\left[(1+\varkappa) + sT_f(1-\varkappa)\right] + k_i
\end{aligned}$$

$$\begin{aligned}
0 &= (1-\varkappa)s^2 T_i T_f + sT_i(1+\varkappa) + k_i \\
&= s^2 + s\frac{1}{T_f}\frac{1+\varkappa}{1-\varkappa} + \frac{k_i}{1-\varkappa}\frac{1}{T_i T_f} \\
\text{to compare to} \quad 0 &= s^2 + 2D_c s\Omega_c + \Omega_c^2 \quad \text{with} \quad {}_c \quad \text{for closed loop} \\
\text{yielding with} \quad T_f &= \frac{2}{\pi}T_E \rightarrow (\Omega_c T_E)^2 = k_i \frac{\pi}{2}\frac{1}{1-\varkappa}; \quad 2D_c = \frac{1+\varkappa}{1-\varkappa}\frac{1}{(\Omega_c T_E)}\frac{\pi}{2}
\end{aligned}$$

Let $k_i = \pi/4$ (fixed) then $\Omega_c T_E = 1.27$; $2D_c = 1.99$
or set $k_i = (\pi/4)\cdot(1+F)$ (gain-scheduled) then $\Omega_c T_E = 1.50$; $2D_c = 1.67$

Note that the first anti-resonance is at $\Omega_{a_1} T_E = \pi$. Thus the frequency margin Ω_{a_1}/Ω_c is 2...3.

So far both k_i and T_i are gain-scheduled. From an application point of view it is more convenient to replace the gain-scheduled T_E by the echo travelling time at zero flow $T_0 := 2L/U_F$ (which is constant) valid for $0 \le F \ll 1.0$:

$$\frac{k_i}{T_i} = \frac{(\pi/4)\cdot(1+F)}{2(L_t ot/U_F)}\cdot(1-F^2) = \frac{(\pi/4)}{2(L_t ot/U_F)}\cdot(1+F-F^2-F^3) \rightarrow\approx \frac{(\pi/4)}{2(L_t ot/U_F)}\cdot(1+F) = (\pi/4)\cdot(1+F)\cdot\frac{1}{T_0}$$

Summary:
Thus the master controller parameters are set to: $k_i := (0.80) \cdot (1 + F)$ and $T_i := T_0$.
Note that for k_i this amounts to a compensation of the variable gain $ks_{dn} \sim 1/(1 + F)$ of the measured variable h_{out}, that is to a constant overall gain of the integral action. And together with its constant time parameter $T_i = T_0$ this will keep the dynamic response due to the I-action of the master controller *invariant over F* !
This particular property will be aimed at also in cases [B] and [C].

Frequency responses

This is to confirm and visualise the results of the analysis above.

.sce file

```
// s_c6_01_1_03.sce
// Glf 2017_06_15
// case [A]: plant with cascaded PI-2dof-controller
// controlled master variable: outflow level
// controlled slave variable: outflow level
// control variable: outflow at outflow point
// inputs: level reference  and inflow


// plant data
//**************
   L = 500;   U_F = 5.0;

// select flow operating point
//    U = 2.00;
      U = 0.20;
      F = U/U_F;       F_d = 0.40;

//Froude wave traveling time
T_dn=L/(U_F)*(1/(1+F)); T_up=L/(U_F)*(1/(1-F));
T_E = T_dn + T_up; T_0 = 2*L/U_F;

// Froude wave height
ks_dn = F_d/(1 + F);
ks_up = F_d/(1 - F);
ks_E = ks_dn + ks_up;

// P-action, cascaded 'slave' controller
//*************************************
    kp = 1.00*(1/F_d)*(1.0+F);
//  kp = 1.60*(1/F_d)*(1.0+F);

// I-action, cascaded 'master' controller
// **************************************
    ki = 0.8*(1 + F); Ti = T_0;

// select frequency bounds
// ***********************
vOMmin  = (0.20)/T_0; vOMmax =(40.0)/T_0;
vOMstep = 0.01/T_0; vOM=(vOMmin :vOMstep :vOMmax);

D_dn = %e^(-imult(vOM*T_dn));
D_up = %e^(-imult(vOM*T_up));
D_E  = %e^(-imult(vOM*T_E));

rei=ones(1,length(vOM)); iei=zeros(1,length(vOM));
cei = rei + imult(iei);

// 'zero losses':
kap_E = 1.0 - 0.00;

// P-action, closed 'slave' loop
gs_dn = (ks_dn*kp); gs_up = (ks_up*kp);
Np = (cei+(gs_dn/gs_up)*kap_E.*D_E);
Dp = (cei-((1.0-gs_dn)/(1.0+gs_up))*kap_E.*D_E);

E_2_v = (gs_up/(1+gs_up))*Np./Dp;
E_2_z = (ks_E/(1 + gs_up))*D_dn./Dp;

// I-action 'master' controller
S_2 = - imult(ki./(vOM*Ti));

// open 'master' loop
  oF_m = E_2_v.*S_2;

// closed master loop, reference input
  cF_m_r = (oF_m)./(cei + oF_m);
// closed master loop, disturbance input,
  cF_m_z = E_2_z./(cei + oF_m);

// Bode
  [cphi_m_r,cdb_m_r] = phasemag(cF_m_r,"c");
  [cphi_m_z,cdb_m_z] = phasemag(cF_m_z,"c");

  for kFig = 1:1:2, clf(kFig); end
f1 = scf(1);
plot2d(vOM,[cdb_m_r',cdb_m_z'],logflag="ln",...
style=[2,5], rect=[vOMmin,-40., vOMmax,+5.0]);
xtitle(["case A: PcI-control, closed loop; ...
    F =",msprintf('%5.4f',F)]);  xgrid(1);
legend('h_out [db] from reference input r_2_u',...
       'h_out [db] from inflow q_1_z',3);

f2 = scf(2);
plot2d(vOM,[cphi_m_r',cphi_m_z'],logflag="ln",...
style=[2,5], rect=[vOMmin,-800.,vOMmax,+100.0]);
xtitle(["case A: PcI-control, closed loop; ...
    F =",msprintf('%5.4f',F)]);  xgrid(1);
legend('h_out [phi] from reference input r_2_u',...
       'h_out [phi] from inflow q_1_z',3);
```

Discussion

- The discussion above has focused on the closed loop eigenvalues only, but not on the residua. This aspect is added here.

- The reference gain response decays with $-20db/dec$ and the mean phase to $-90°$. And the inflow disturbance gain response at low frequency moves up with $+20db/dec$ with phase starting at $+90°$, and then levels out at λ_z. This corresponds to the TF's of first order from above with the 'time constant' Θ.

- As expected the anti-resonances locations are not modified by the closed loop, and the gain there is given by the value of F. The resonance peaks are levelled by the P-control action for the reference response.

- On the inflow disturbance response the gain shows small resonances and anti-resonances. This effect increases with the slave controller gain $k_p ks_{dn} = 1.0 \rightarrow 1.6$. Also the first peak moves to higher frequencies and to lower gain, as to be expected from the TF.

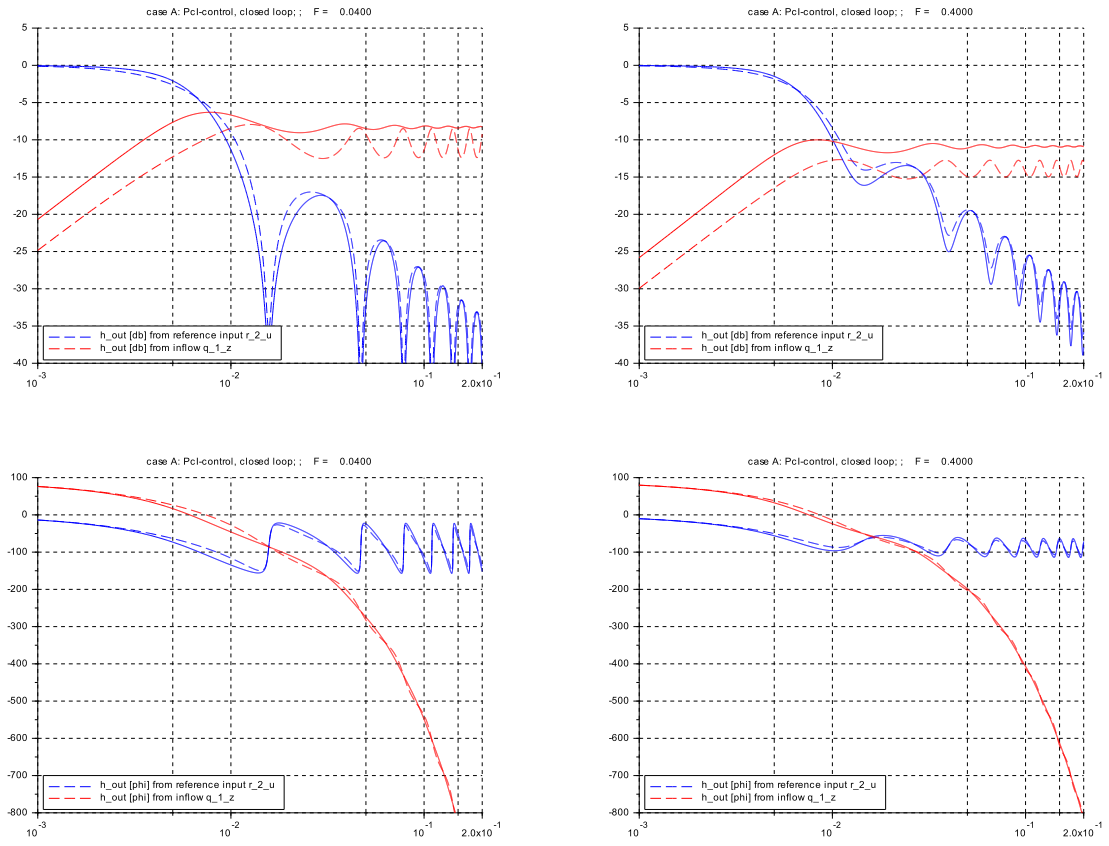


Figure 6.15: Case [A]: Bode plots for PcI-control of h_{out} for inputs 'reference'(blue) and 'inflow'(red),
(left) for $F = 0.040$, (right) for $F = F_d = 0.40$
(solid) for $k\_p \cdot ks\_dn = 1.00$, (dashed) for $k\_p \cdot ks\_dn = 1.60$

Simulation results

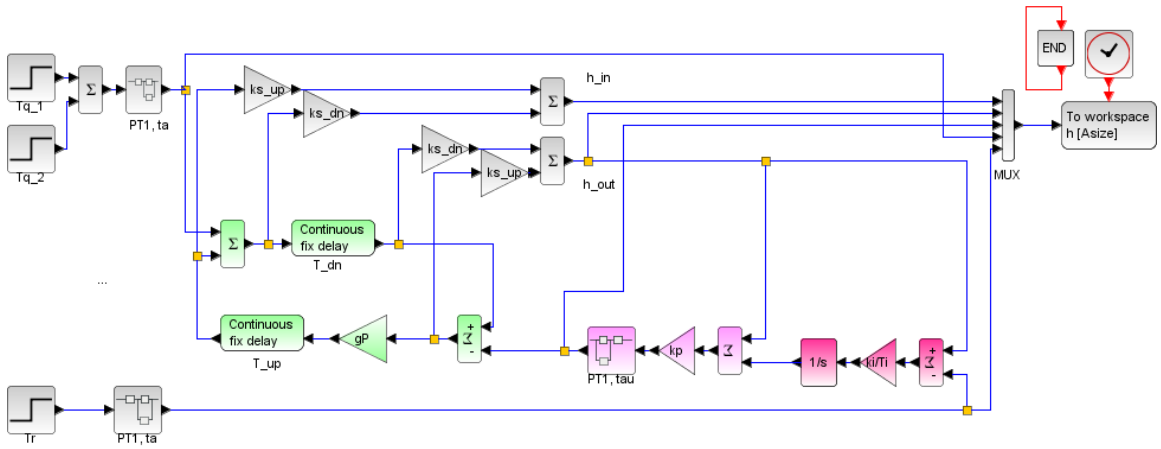


Figure 6.16: Case [A]: top level diagram of the PcI-controlled loop

.sce file for 'context'

```
// s_c6_01_1_04_context.sce
// inflow and outflowlevel reference steps
// case A: cascaded-PI-2dof-controller
// co-located at outflow
// Glf   2017_06_15

L = 500;  U_F = 5.0;
// select Froude number  and 'best' gain
      U = 2.0;
//    U = 0.20;
      F = U/U_F; F_d = 0.40;

// delay times
T_up = (L/U_F)/(1-F); T_dn = (L/U_F)/(1+F);
T_E = T_up + T_dn;    T_0 = 2*(L/U_F);
bufsize = 1000*T_E/2;

// no losses
gP = 1.0 - 0.00;
ta = 2.5; tau = 0.1*ta;

// size of Froude wave
ks_dn = F_d/(1+F); ks_up = F_d/(1-F);

// reference step
Tr = 2600.; rh0 = 0.0; rh1 = 1.0*F_d;

// inflow steps
Tq_1 =   10.0; zq0 = 0.0; zq1 =  1.0;
Tq_2 = 1200.0; zq0 = 0.0; zq2 = -1.0;

// P-action, cascaded 'slave' controller
      kp = 1.00*(1/F_d)*(1+F);
//    kp = 1.60*(1/F_d)*(1+F);

// I-action, cascade 'master' controller
ki = 0.80*(1 + F); Ti = T_0;

// simulation parameters
T_fin = 4000;
   CN = 4000;
 delT = T_fin/CN;   // readout clock ticks
CC = 6;   // no of channels 5 + 1 for time
Asize = 1.01*CC*CN; // size of data arrays
```

.sce file for 'run' and 'plot'

```
// s_c6_01_1_04_crunplot
// case 1: cascaded PI-2dof-control,
// co-located at outflow
//  Glf  2017_06_15

stacksize('max');
exec('s_c6_01_1_04_context.sce', -1);
importXcosDiagram('s_c6_01_1_04.zcos');
typeof(scs_m); scs_m.props.context;
Info = list();
Info = scicos_simulate(scs_m,Info);
//*********************************

vcolor = [ 5, 2, 1, 11, 13];
f1 = scf(1); clf();
plot2d(H.time,H.values,vcolor,...
       rect=[0.,-1.0,T_fin,1.6]);
xtitle(['case A: PcI-control;...
    F =',msprintf('%5.4f',F),...
'; kp =',msprintf('%5.2f',kp),...
', ki = ',msprintf('%5.2f',ki),...
', Ti = ',msprintf('%5.2f',Ti),]); xgrid(1);
legend('h_in','h_out','u_out','q_z_1','r_out',4)
```

Discussion

- The design based on the targets for the eigenvalues:
  bandwidth $\pi/3.0 < \Omega T_E < \pi/2.5$, and damping ratio $1.4 < 2D < 1.7$
  for the approximate system works well on the original system: For all cases the overshoot on $h_{out}(t)$ is negligible, with no sluggish responses.

- Also the gain-scheduling derived on the approximate system
  $k_i = 0.80(1 + F)$ and $T_i = T_E$
  works well on the original system: The decay transient stays within the acceptable range.

- The decay time to near zero for the case $k_p ks_{dn} = 1.0$ is approx 600 s, that is 3 T_E, which corresponds to the prediction from the first order TF with 'time constant' Θ.
  Going to $k_p ks_{dn} = 1.60$ reduces this to approx 500 s.

- The reference step responses are well behaved. There is no perceptible difference between the responses for both P-gain values, which corresponds with the small deviations of the reference frequency responses, Fig.6.15

- In contrast the inflow disturbance responses differ markedly. For $k_p ks_{dn} = 1.0$ the response is smooth (with the exception of the first step in q_u_2 on the arrival of the Froude wave at the outflow end). But for $k_p ks_{dn} = 1.60$ there is a stronger flow peaking and a second smaller jump after the echo travel time. This may not be admissible for the downstream channel operation.

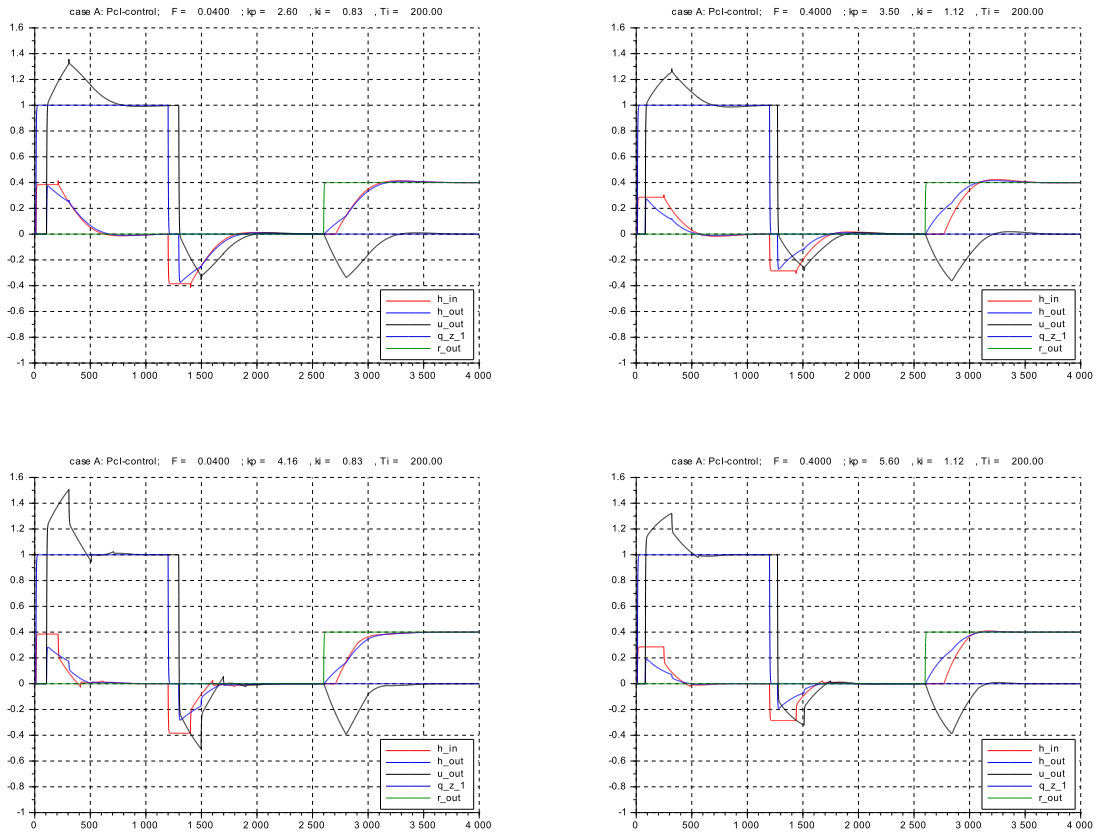


Figure 6.17: Case [A]: step responses for PcI-control of h_out for inputs 'inflow' and 'reference',
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top) for $k\_p \cdot ks\_dn = 1.00$, (bottom) for $k\_p \cdot ks\_dn = 1.60$

6.3.6 Case [B]: Designing the PcI-controller

Fig.6.18 shows the block diagram of the control system, by modification of Fig.6.14.

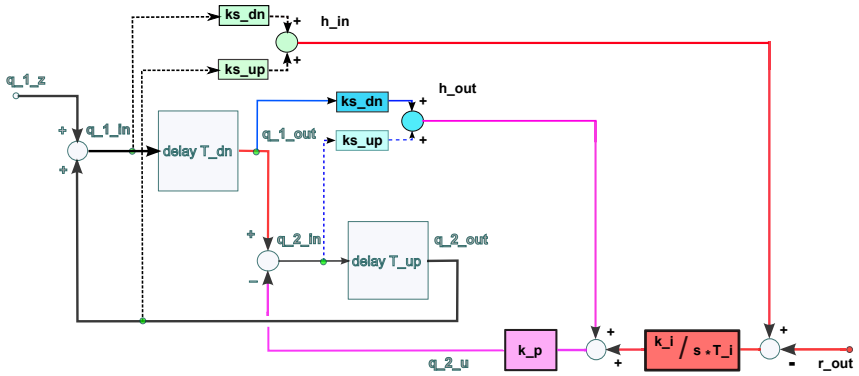


Figure 6.18: Case [B]: Block diagram for PcI-control of h_out by 'slave P-controller' (pink) and h_in by 'master I-controller' (red), acting on control outflow q_2_u

The P-'slave'-control loop is the same as for case [A]. In order to keep its closed loop response invariant of F, the gain k_p must be set to $k_p \cdot ks_{dn} := const$. And for the impedance matching design, $const := 1.0$.

Transfer function analysis

First the TF of the system without any control from control input q_u_2 to y_in is by using the same technique as previously:

$$y_{in} = \frac{1}{1 - G_E} \left[(ks_{dn} + ks_{up}) \cdot G_{up} \cdot (-q_{u_2}) \quad + \quad (ks_{dn} + ks_{up} \cdot G_E) \right]$$

Closing the P-'slave'-loop as for case [A] produces

$$y_{out} = \frac{1}{(1 + k_p ks_{up}) + (k_p ks_{dn} - 1)\, G_E} \left[k_p \left(ks_{up} + ks_{dn} G_E \right) \cdot r_{out} \quad + \quad (ks_{dn} + ks_{up}) \cdot G_E \cdot q_{z_1} \right]$$

and for the additional output y_in needed for the 'master'-control:

$$y_{in} = \frac{1}{(1 + k_p ks_{up}) + (k_p ks_{dn} - 1)\, G_E} \left[k_p \left(ks_{up} + ks_{dn} \right) \cdot G_{up} \cdot r_{out} \quad + \quad (ks_{dn} + ks_{up} \cdot G_E) \cdot q_{z_1} \right]$$

Closing the I-'master'-loop with using the abbreviations $\varkappa,\ \mu,\ \lambda_r,\ \lambda_z$ from above

$$\begin{aligned}
r_{out} \quad &\rightarrow \quad u_{i_2} \\
S_i \quad &= \quad \frac{k_i}{T_i} \cdot (r_{in} - y_{in}) \\
y_{in} \quad &= \quad \frac{\left[\lambda_r (1 + \mu)\ G_{up} \cdot r_{in} \quad + \quad \lambda_z (\mu + 1.0\ G_E) \cdot s \frac{T_i}{k_i} \cdot q_{z_1} \right]}{\left[s \frac{T_i}{k_i} \cdot (1 + \varkappa G_E) + \lambda_r\, (1 + \mu)\, G_{up} \right]} \\
\text{with} \quad \lambda_r (1 + \mu) \quad &= \quad \frac{k_p ks_{dn} \cdot (ks_{up}/ks_{dn})}{1 + k_p ks_{dn} \cdot (ks_{up}/ks_{dn})} \cdot (1 + (ks_{dn}/ks_{up})) \quad \text{and set} \quad k_p ks_{dn} := 1.0 \\
&= \quad \frac{(ks_{up}/ks_{dn}) + 1}{1 + (ks_{up}/ks_{dn})} \rightarrow 1.0 \quad \text{independent of} \quad F!
\end{aligned}$$

Consider next the common denominator in order to determine the poles of the TF with assuming $k_p k s_{dn} := 1.0$ that is $\varkappa = 0$:

$$0 = s\frac{T_i}{k_i} \cdot (1+0) + 1.0 \cdot G_{up} \quad \text{with} \quad G_{up} = e^{-sT_{up}}$$

and replace G_{up} by the first order lag $G_f = 1/(1+sT_f)$
with T_f adjusted such that G_f produces the same lag at the same frequency $\omega = 1.0/T_{up}$ as the original G_{up}:

$$\begin{aligned}
T_f &:= \frac{\pi}{4} \cdot T_{up} \\
\text{where from above} \quad T_{up} &= \frac{1}{2} T_0 \frac{1}{1-F} \\
\text{and as above} \quad T_i &:= T_0 \\
\text{Further define} \quad k_i &:= k_{i_0} \cdot (1-F) \\
\text{that is} \quad \frac{T_i}{k_i} &= \frac{T_i}{k_{i_0}(1-F)} = 2T_{up}\frac{1}{k_{i_0}}
\end{aligned}$$

Thus for the characteristic equation with the approximation

$$\begin{aligned}
0 &= sT_i + k_i \frac{1}{1+sT_f} \\
&= s^2 2T_{up}\frac{\pi}{4} \cdot T_{up} + s2T_{up} + k_{i_0} \\
&= s^2 + s\frac{4}{\pi}\frac{1}{T_{up}} + k_{i_0}\frac{2}{\pi T_{up}^2} \\
\text{equate to} \quad 0 &= s^2 + s2D\Omega + \Omega^2 \\
\text{that is} \quad (\Omega T_{up}) &= \sqrt{\frac{2k_{i_0}}{\pi}} \\
\text{and} \quad 2D &= 2\sqrt{2}\frac{1}{\sqrt{\pi}}\frac{1}{\sqrt{k_{i_0}}}
\end{aligned}$$

Set $k_{i_0} := 1.0$, then $2D \approx 1.60$ and $\Omega T_0(1-F) \approx 0.400(1-F)$.
Thus the loop bandwidth will decrease with increasing F while the damping stays constant.

Summary controller settings for case [B]: $k_i := 1.0 \cdot (1-F)$ and $T_i := T_0$.

Simulation results

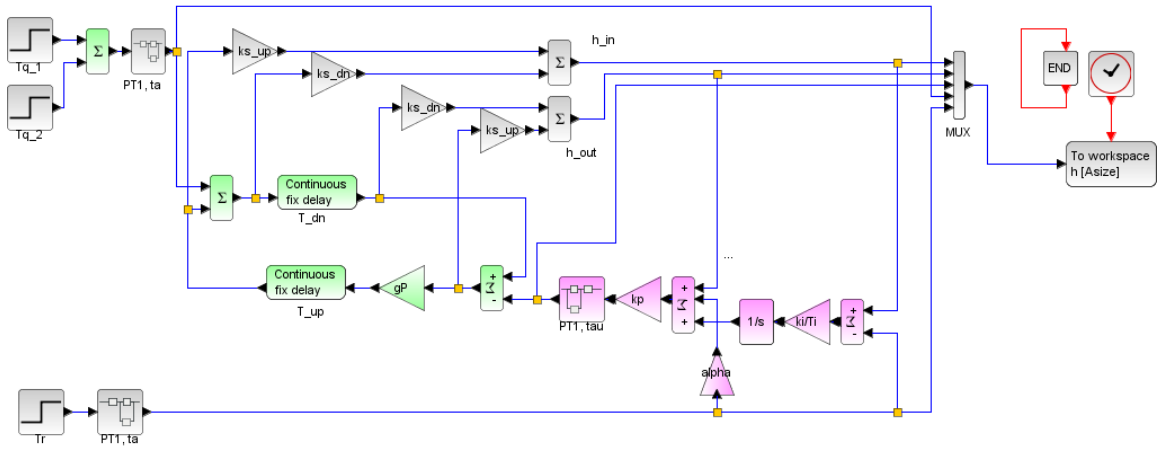


Figure 6.19: Case [B]: Top level diagram of the PcI-controlled loop

.sce file for 'context'

```
// s_c6_01_2_01_context.sce
// inflow and outflowlevel reference steps
// case B: cascaded PI-2dof-controller
// for inflow level, manipulated at outflow
// Glf   2017_06_15

L = 500;   U_F = 5.0;    F_d = 0.40;

// select Froude number  and 'best' gain
      U = 2.0;
//    U = 0.2;

  F = U/U_F;
T_dn = (L/U_F)/(1+F); T_up = (L/U_F)/(1-F);
T_E = T_up + T_dn; T_0 = 2*(L/U_F);

ks_dn=F_d/(1+F); ks_up=F_d/(1-F); ks_r=F_d;
gP = 1.0 - 0.00; bufsize = 1500*T_0/2;

ta = 2.5;   tau = 0.1*ta;

// reference step
Tr = 2600.; rh0 = 0.0; rh1 = 1.0*ks_r;

// inflow steps
Tq_1 =   10.0; zq0 = 0.0; zq1 =  1.0;
Tq_2 = 1200.0; zq0 = 0.0; zq2 = -1.0;

// P-slave-controller for out-level on outflow
      kp = 1.00*(1/F_d)*(1+F);   alpha = 0.0;
//    kp = 1.26*(1/F_d)*(1+F);   alpha = 0.0;

// I-master-controller for in-level on outflow
      ki = 1.00*(1-F);  Ti = T_0;

T_fin = 4000;
   CN = 4000;  delT = T_fin/CN;
   CC = 6;    Asize = 1.01*CC*CN;
```

.sce file for 'run' and 'plot'

```
// s_c6_01_2_01_crunplot
// case B: cascaded PI-2dof-control,
//  inflow level, manipulated at outflow
//  Glf  2017_06_15

stacksize('max');
exec('s_c6_01_2_01_context.sce', -1);
importXcosDiagram('s_c6_01_2_01.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:1,  clf(kfig); end

vcolor = [ 5, 2, 1, 11, 13];

f1 = scf(1);
plot2d(H.time,H.values,vcolor,rect=[0.,-1.0,T_fin,1.6]);
xtitle(['case B: casc PI-2dof-control, F =',...
msprintf('%5.4f',F),'; kp =',msprintf('%5.2f',kp),...
', ki = ",msprintf('%5.2f',ki),', Ti = ',...
msprintf('%5.2f',Ti)]); xgrid(1);
legend('h_in', 'h_out', 'u_out','zq_in', 'rd_out',4)
```

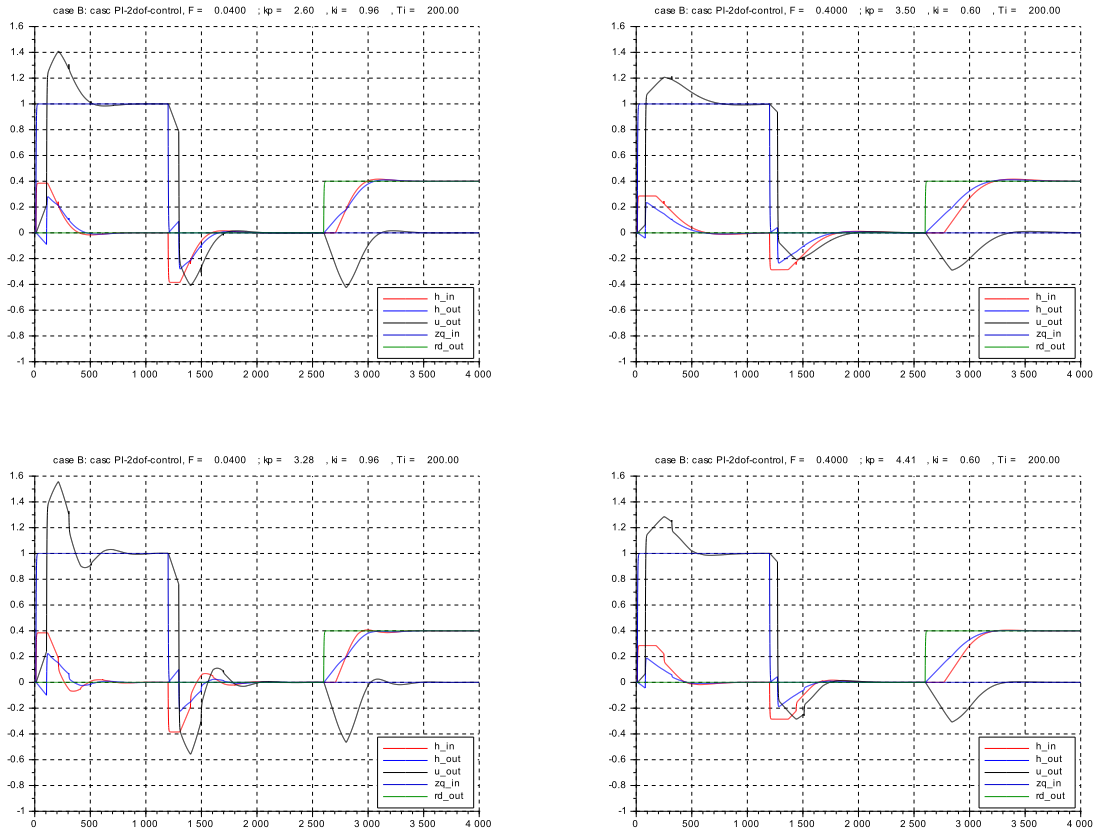


Figure 6.20: Case [B]: step responses for PcI-control of h_{in} for inputs 'inflow' and 'reference' h_{in},
(top) for $k_p\ ks_{dn} = 1.0$ (bottom) for $k_p\ ks_{dn} = 1.26$
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$

Discussion

- Again the design based on the eigenvalues for the approximate system works well on the original system: For all cases the overshoot on $h_{in}(t)$ is negligible, with no sluggish responses.
- Also the gain-scheduling derived on the approximate system
  ($k_p = 1.0/ks_{dn}$, $k_i = 1.0(1 - F)$, and $T_i = T_0$)
  works well on the original system: The decay transient stays within the acceptable range.
- The decay time to near zero for the case $k_p ks_{dn} = 1.0$ is approx 600 s, that is 3 T_E, which corresponds to the prediction from the first order TF with 'time constant' Θ.
- In contrast to case[A], here the transient response is very sensitive to an increased gain k_p. The responses get very weakly damped for $k_p ks_{dn} = 1.60$, especially at the low F-value. To illustrate this effect simulations are shown for $k_p ks_{dn} = \sqrt{1.60} \approx 1.26$.
- Thus for case [B] only the impedance matching design seems to work well enough.

In other words the level in case [B] is more difficult to control than in case [A]. And in case [C] this will get even more difficult...

6.3.7 Case [C]: Designing the PcI-controller

Fig.6.21 shows the block diagram of the control system, modified from Fig.6.18.

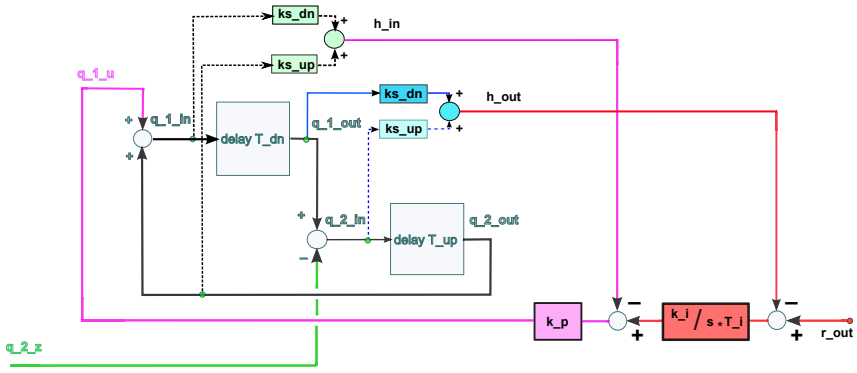


Figure 6.21: Case [C]: Block diagram for PcI-control of *h_in* by 'slave P-controller' (pink) and *h_out* by 'master I-controller' (red), acting on control inflow *q_1_u*

Transfer function analysis

Using the same procedure as for cases [A] and [B] the characteristic equation is

$$
\begin{aligned}
0 &= sT_i(1+\varkappa G_E) + k_i \cdot \lambda_r \cdot (1+\mu) \cdot G_{dn} \\
\text{where}\quad \lambda_r \cdot (1+\mu) &\rightarrow 1.0 \quad \text{as above and} \quad \varkappa := 0 \quad \text{from} \quad k_p ks_{dn} := 1.0 \\
\text{with}\quad G_{dn} &= e^{-sT_{dn}} \\
\text{and}\quad T_{dn} &= \frac{1}{2} \cdot T_0 \frac{1}{1+F} \\
\text{replacing}\quad G_{dn} &\rightarrow G_f = \frac{1}{1+sT_f} \quad \text{with} \quad T_f = \frac{\pi}{4} T_{dn} \\
\rightarrow\quad 0 &= s^2 + s\frac{1}{T_f} + k_i \frac{1}{T_i T_f} \\
\text{defining}\quad k_i &:= k_{i_0} \cdot (1+F) \quad \text{and inserting} \\
&= s^2 + s\frac{4}{\pi}\frac{1}{T_{dn}} + \frac{2}{\pi} k_{i_0} \left[\frac{1}{T_{dn}}\right]^2 \\
&= s^2 + s2D\Omega + \Omega^2 \\
\text{equating yields with}\quad k_{i_0} &:= 1.0 \\
(\Omega T_{dn}) &= \sqrt{\frac{2k_{i_0}}{\pi}} \approx 0.80 \quad \text{and} \quad 2D = 2\sqrt{2}\frac{1}{\sqrt{\pi}} \approx 1.60 \\
\text{and}\quad \Omega T_0(1+F) &\approx 0.400(1+F)
\end{aligned}
$$

Thus the loop bandwidth will increase with increasing F while the damping stays constant.

Summary controller settings for case [C]: $k_i := 1.0 \cdot (1+F)$ and $T_i := T_0$.

Simulation results

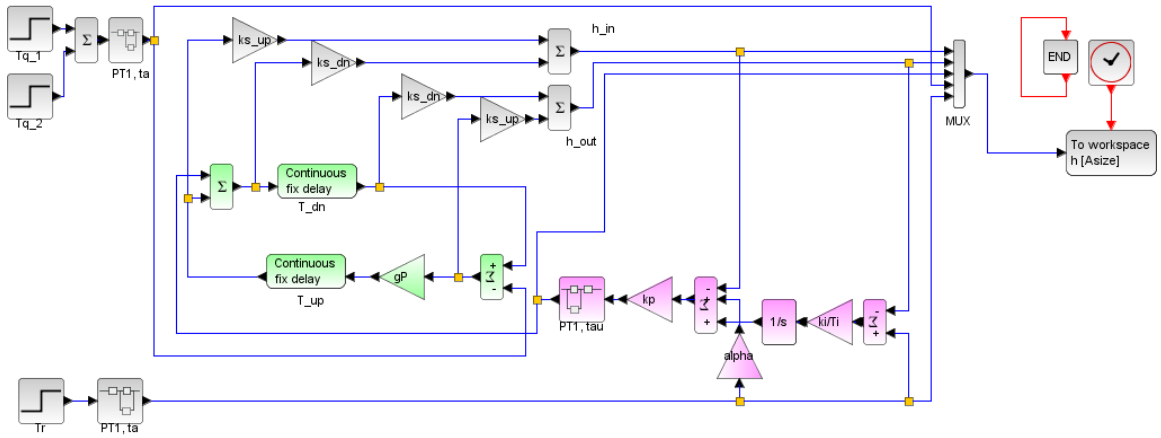


Figure 6.22: Case [C]: Top level diagram of the PcI-controlled loop

.sce file for 'context'

```
// s_c6_01_3_01_context.sce
// inflow anf outflowlevel reference steps
// case C: cascaded PI-2dof-controller for outflow at inflow
// Glf   2017_06_15

L = 500;  U_F = 5.0; F_d = 0.40;

// select Froude number  and 'best' gain
//    U = 2.0;
      U = 0.2;

  F = U/U_F;
T_dn = (L/U_F)/(1+F); T_up = (L/U_F)/(1-F);
T_E = T_up + T_dn;  T_0 = 2*(L/U_F);

ks_dn=F_d/(1+F); ks_up=F_d/(1-F); ks_r=F_d;
gP = 1.0 - 0.00; bufsize = 2000*T_E/2;

ta = 2.5;   tau = 0.1*ta;

// reference step
Tr = 2600.; rh0 = 0.0; rh1 = 1.0*ks_r;

// inflow steps
Tq_1 =   10.0; zq0 = 0.0; zq1 =  1.0;
Tq_2 = 1200.0; zq0 = 0.0; zq2 = -1.0;

// P-slave-controller for in-level on inflow
      kp = 1.00*(1/F_d)*(1-F);   alpha = 0.;
//    kp = 1.26*(1/F_d)*(1-F);   alpha = 0.;
// I-master-controller for out-level on inflow
      ki = 1.00*(1-F);  Ti = T_0;

T_fin = 4000;
   CN = 4000;  delT = T_fin/CN;
   CC = 6;    Asize = 1.01*CC*CN;
```

.sce file for 'run' and 'plot'

```
// s_c6_01_3_01_crunplot
// case C: cascaded PI-2dof-control,
// for outflow level, manipulated at inflow
//  Glf  2017_06_15
//
stacksize('max');
exec('s_c6_01_3_01_context.sce', -1);
importXcosDiagram('s_c6_01_3_01.zcos');
typeof(scs_m); scs_m.props.context;
Info = list(); Info = scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:1,  clf(kfig); end

vcolor = [ 5, 2, 1, 11, 13];

f1 = scf(1);
plot2d(H.time,H.values,vcolor,rect=[0.,-1.0,T_fin,2.0]);
xtitle(['case C: casc PI-2dof-control, F =',...
msprintf('%5.4f',F),'; kp=',msprintf('%5.2f',kp),...
', ki=',msprintf('%5.2f',ki),', Ti=',...
msprintf('%5.2f',Ti)]); xgrid(1);
legend('h_in', 'h_out', 'u_in','zq_out', 'r_h_out',4)
```

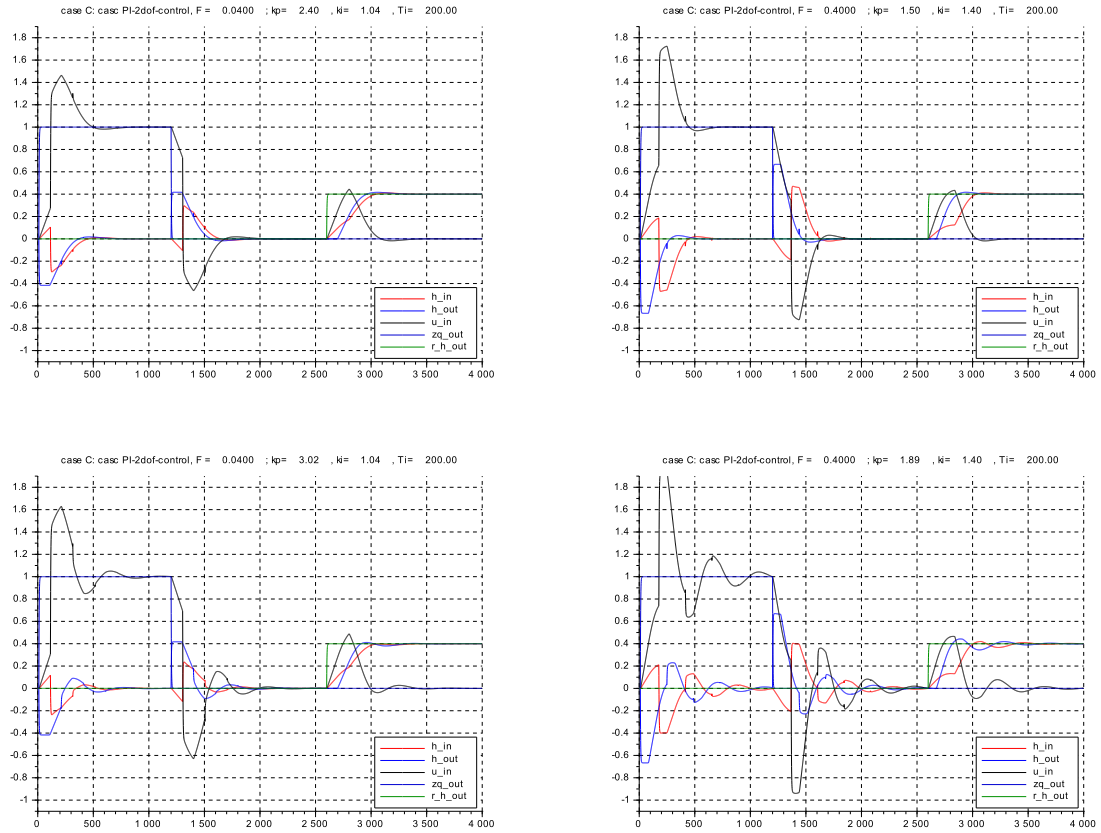


Figure 6.23: Case [C]: step responses for PcI-control of h_{out} for inputs 'inflow' and 'reference' r_{out},
(top) for $k_p\ ks_{dn} = 1.0$ (bottom) for $k_p\ ks_{dn} = 1.26$
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$

Discussion

- Again the design based on the eigenvalues for the approximate system works well on the original system: For all cases the overshoot on $h_{in}(t)$ is negligible, with no sluggish responses.
- Also the gain-scheduling derived on the approximate system
$$k_p = 1.0/ks_{dn},\ k_i = 1.0(1+F), \quad \text{and} \quad T_i = T_0$$
works well on the original system: The decay transient stays within the acceptable range.
- The decay time to near zero for the case $k_p ks_{dn} = 1.0$ is approx 600 s, that is 3 T_E.
- In contrast to case[B], here the transient response is even more sensitive to an increased gain k_p. The responses get very weakly damped for $k_p ks_{dn} > 1.0$, especially at $F = 0.40$. To illustrate this effect simulations are shown for $k_p ks_{dn} = \sqrt{1.60} \approx 1.26$. Therefore also for case [C] only the impedance matching design seems to work well enough for practical applications.
- Further simulations (not shown here) indicate that this effect is mainly due to k_p and less to k_i, T_i.

The reason for this is the difference in the shape of the open 'master'-loop Nyquist contour. For case [A] the delay G_E in $1 + G_E$ introduces a circle which is offset on the positive real axis such that it never crosses the imaginary axis into the left half plane. However in cases [B] with G_{up} and [C] with G_{dn} this circle is centered at the origin and thus crosses well into the left half plane. Therefore the phase shift from G_{up} in case [B] or G_{dn} in case [C] will get much larger than from $1 + G_E$ in case[A] for frequencies $(\omega\ T_{up} \geq 1)$ or $(\omega\ T_{dn} \geq 1)$.
And this will make the closed loop dynamics much more sensitive to increasing gain k_p.

6.4 Control Design for the linearized state space model, 'sys_ss'

6.4.1 Motivation

This is meant as an intermediate step to the final goal to design model based controllers for the nonlinear wide range time domain model with non-uniform geometry and friction from chapter 3. It will allow checking the performance of the control design from the previous section (developed with the PDE-model) both in the frequency and the time domain in a 'laboratory setting'. Further it allows to demonstrate the effect of spacial discretisation in a more compact way than with the full time domain model.
In order to stay as close as possible to the PDE model the same assumptions will be used:
- simple geometry: constant width B and constant bottom depth S, long channel $L_{tot} \gg B$,
- zero friction,
- small deviations from a quasi-stationary operating point $\overline{Q}, \overline{D}$, showing the effect of F,
- starting at a high number of compartments, then down to the chapter 3 number,
- and the same simplified boundary conditions on the inflow and outflow ends as in the PDE-model.

6.4.2 Modelling

The basic element for the plant model is taken from chapter 2 in its state-space form, which is linearized and with 'floating' coefficients around the local operating point (local depth $\overline{D}$ and flow $\overline{Q}$).

$$Q_i(t) \;\rightarrow\; \overline{Q}\left(1+\frac{\delta Q_i(t)}{\overline{Q}}\right); \qquad D_{i-1}^2 \rightarrow \overline{D}^2\left(1+2\frac{\delta D_{i-1}(t)}{\overline{D}}+0\right);$$

$$\frac{Q_i^2}{D_i'} \;\rightarrow\; \frac{\overline{Q}^2}{\overline{D}}\left(1+2\frac{\delta Q_i(t)}{\overline{Q}}-\frac{\delta D_i'(t)}{\overline{D}}+0\right) \;=\; \frac{\overline{Q}^2}{\overline{D}}\left(1+2\frac{\delta Q_i(t)}{\overline{Q}}-\frac{\delta D_{i+1}(t)}{\overline{D}}+0\right)$$

while using for discretization of D_i: $\varkappa = 0$ for $0 \le F \le 1.0$, ('downwind') as the control design shall focus on the subcritical flow regime.

and further using the abbreviations

$$\frac{\delta D_{i-1}(t)}{\overline{D}} \;\rightarrow\; x_{i-1}; \qquad \frac{\delta Q_i(t)}{\overline{Q}} \rightarrow x_i; \qquad \frac{\delta D_{i+1}(t)}{\overline{D}} \rightarrow x_{i+1};$$

$$\frac{\delta Q_{i-2}(t)}{\overline{Q}} \;\rightarrow\; x_{i-2} \;\; (:= u_{i-2}); \qquad \frac{\delta D_{i-2}'(t)}{\overline{D}} \rightarrow x_{i-1}; \qquad \frac{\delta Q_{i+2}(t)}{\overline{Q}} \rightarrow x_{i+2} \;\; (:= u_{i+2});$$

And for the coefficients for the local subsystem around location i with compartment length $L := L_{tot}/(N+1)$, where N is the number of momentum balance compartments in the overall channel length L_{tot}. Note that this generates a state space system of order $2 \cdot N + 1$.

$$\text{time constant for filling compartment } i-1 \qquad \frac{LB\overline{D}}{\overline{Q}} \;=\; T_{i-1}$$

$$\text{time constant for momentum compartment } i \qquad \frac{L\overline{Q}}{g\overline{D}^2\overline{B}} \;=\; T_i$$

$$\text{time constant for filling compartment } i+1 \qquad \frac{LB\overline{D}}{\overline{Q}} \;=\; T_{i+1}$$

$$\text{coefficient for momentum flows in compartment } i \qquad \frac{\overline{Q}^2}{g\overline{B}^2\overline{D}^3} \;=\; \frac{\overline{B}^2\overline{D}^2\overline{U}^2}{g\overline{B}^2\overline{D}^3} = \frac{\overline{U}^2}{g\overline{D}} = \overline{F}^2 := \phi$$

Equations for x_{i-1} and x_{i+1} are straightforward, but equation for x_i needs some further steps

$$T_i\frac{d}{dt}x_i \;=\; +1x_{i-1} \;-\; 1x_{i+1} + \phi\left[2x_{i-2}-x_{i-1}\right] - \phi\left[2x_i - x_{i-1} - x_{i+1}\right]$$

$$\;=\; (1-\phi)x_{i-1} \;-\; 2\phi x_i \;-\; (1-\phi)x_{i+1} \;+\; 2\phi x_{i-2}$$

The equation for x_2 shall be rewritten in condensed form by using the abbreviations, (note again $\varkappa := 0$)

$$\begin{aligned} a &= (1-\phi) \\ b &= (1-\phi) \\ c &= (\varkappa\phi) = 0. \\ p &= (2\phi) = (2\overline{F}^2) \end{aligned}$$

Then

$$\begin{array}{rcccccc} T_{i-1}\frac{d}{dt}x_{i-1} & = & & -1\cdot x_i & & +1\cdot x_{i-2} & \\ T_i\frac{d}{dt}x_i & = & +a\cdot x_{i-1} & -p\cdot x_i & -b\cdot x_{i+1} & +p\cdot x_{i-2} & \\ T_{i+1}\frac{d}{dt}x_{i+1} & = & & +1\cdot x_i & & & -1\cdot x_{i+2} \end{array}$$

The next step is to replace the local reference pair $\overline{Q},\ \overline{D}$ by the design reference pair $Q_d,\ D_d$. Then the state variables are re-defined as

$$\begin{array}{lll} \frac{\delta D_{i-1}(t)}{D_d} \to x_{i-1}; & \frac{\delta Q_i(t)}{Q_d} \to x_i; & \frac{\delta D_{i+1}(t)}{D_d} \to x_{i+1}; \\ \frac{\delta Q_{i-2}(t)}{Q_d} \to x_{i-2} \ (:= u_{i-2}); & \frac{\delta D'_{i-2}(t)}{D_d} \to x_{i-1}; & \frac{\delta Q_{i+2}(t)}{Q_d} \to x_{i+2} \ (:= u_{i+2}); \end{array}$$

The coefficients turn into:

$$\begin{array}{rrcl} \text{time constant for filling compartment } i-1 & \frac{LBD_d}{Q_d} & = & T_{i-1} = \frac{L}{U_d} = \frac{L}{U_F}\cdot\frac{1}{F_d} \\ \text{time constant for momentum compartment } i & \frac{LQ_d}{gD_d^2B_d} & = & T_i = \frac{LU_dB_dD_d}{U_F^2B_dD_d} = \frac{L}{U_F}\frac{U_d}{U_F} = \frac{L}{U_F}\cdot F_d \\ \text{time constant for filling compartment } i+1 & \frac{LBD_d}{Q_d} & = & T_{i+1} = \frac{L}{U_F}\cdot\frac{1}{F_d} \end{array}$$

$$\text{and using} \qquad \overline{D} := D_d \quad \text{but generally} \quad \overline{Q} \neq Q_d$$

$$\begin{array}{rrcl} \text{momentum flows into compartment } i \text{ from } Q_k & \frac{\overline{Q}\cdot Q_d}{gB_d^2\overline{D}^3} & = & \frac{B_d^2\overline{D}^2\overline{U}\cdot U_d}{gB_d^2\overline{D}^3} = \frac{\overline{U}\cdot U_d}{g\overline{D}} = \overline{F}\cdot F_d \\ \text{and from } D_k & \frac{\overline{Q}^2}{gB_d^2\overline{D}^3}\cdot\frac{D_d}{\overline{D}} & = & \frac{\overline{Q}^2B_d^2D_d^2}{U_F^2B_d^2D_d^2} = \overline{F}^2 \end{array}$$

Dropping the $\overline{var}$ at the local operating point finally (where $\varkappa := 0$):

$$\begin{array}{rcccccc} T_{i-1}\frac{d}{dt}x_{i-1} & = & +1\cdot x_{i-2} & & -1\cdot x_i & & \\ T_i\frac{d}{dt}x_i & = & +2F_dF\cdot x_{i-2} & +(1-F^2)\cdot x_{i-1} & -2F_dF\cdot x_i & -(1-F^2)\cdot x_{i+1} & \\ T_{i+1}\frac{d}{dt}x_{i+1} & = & & & +1\cdot x_i & & -1\cdot x_{i+2} \end{array}$$

6.4.3 Implementation in Scilab/SciNotes

The standard state space representation `syslin` of `'scilab'` with matrices A, B, C, D with state vector length $Mx = 2N + 1$ is used. Note that all coefficients in line k need to be divided by T_k. The frequency responses are obtained by `scilab` functions `repfreq` and the step responses by `csim`.
Two values of the spacial discretization parameter N are taken, the first ($N = 100$) to be close to the `sysinf`-case from above, and the second ($N = 20$) to correspond to the basic assumption in chapter 3. No smaller values of N are shown here to conserve space, although the following script is suitable down to $N = 1$. – And the results for the system input at z_in are omitted for the same reason.

```
// s_c6_02_0.sce
// Glf 2017_06_15
// Open loop Bode plot, step response
// from outflow, manipulated var. u_out
// and from inflow  z_in
// to outflow level x_out and to inflow level x_in

stacksize('max');
   for kFig= 1:1:6 clf(kFig); end
// reference values
  g = 10.; U_d = 2.0; D = 2.5; L_tot = 500.;

// select flow speed / Froude number
//    U = 2.0;
      U = 0.20;

// selecting SIZE  N / building the system:
//   N = 100;
     N = 20;

  Mx = 2*N+1;   L = L_tot/(N+1);
// defining matrix elements
  U_F = sqrt(g*D);  F = U/U_F;  F_d = U_d/U_F;
  PHI = F^2;  PHI_d = F_d^2;  T = (L/U_F);
  T_1 = T*(1/F_d);  T_2 = T*(F_d);
  p = 2*F*F_d;  a = (1 - PHI);  b = a;  c = 0;

// time constants vector buildup
  vT = ones(1, Mx);
  for n= 1:2:Mx-1,
   vT(n)  = T_1; vT(n+1) = T_2; end;
   vT(Mx) = T_1;

// system matrix AM buildup
   AM = zeros(Mx,Mx);

// Volume elements
// first line
  m = 1; AM(m,m+1) = -1.0/vT(m);
// intermediate non-even numbered lines
  for m = 3:2:Mx-2,
   AM(m,m-1)= +1.0/vT(m); AM(m,m+1)= -1.0/vT(m); end
//final line
  m = Mx; AM(m,m-1) = +1.0/vT(m);

// Flow/Momentum elements
// second line
  m = 2;
    AM(m,m-1) = +a/vT(m); AM(m,m) = -p/vT(m);
    AM(m,m+1) = -b/vT(m);
//all other even numbered lines
  for m = 4:2:Mx-1,
    AM(m,m-3) = -c/vT(m); AM(m,m-2) = +p/vT(m);
    AM(m,m-1) = +a/vT(m); AM(m,m)   = -p/vT(m);
    AM(m,m+1) = -b/vT(m); end;

//Matrix BM for  input u_out
//**************************
 BM_u = zeros(Mx,1); BM_u(Mx)= (1/vT(Mx));

// Matrix CM
  CM_u=zeros(2,Mx);
  CM_u(1,Mx) = 1.0;  // for x(out)
  CM_u(2,1)  = 1.0;  // and for x(in)

    H_u=syslin('c',AM,BM_u,CM_u);

// plant frequency response
f_min=0.00010; f_max=0.10; fstep=0.000010;
  f_r=f_min:fstep:f_max;

// Bode plot plant
  repf_u = repfreq(H_u,f_r);
  [dbB_1,phiB_1]=dbphi(repf_u(1,:));
  [dbB_2,phiB_2]=dbphi(repf_u(2,:));

f1 = scf(1);
plot2d(f_r,[dbB_1',dbB_2'],logflag = 'ln',...
    style=[2,5],rect=[f_min,-50.,f_max,+50.0]);
  xgrid(1); legend('x_out','x_in',2);
xtitle(['plant: Bode plots DB to outflow u_out,...
  no. compartments =',msprintf('%3.0f',Mx),...
 ' ,  Froude F = ',msprintf('%5.2f',F)]);

f2 = scf(2);
plot2d(f_r,[phiB_1',phiB_2'],logflag = 'ln',...
    style=[2,5],rect=[f_min,-500.,f_max,+100.0]);
  xgrid(1); legend('x_out','x_in',2);
xtitle(['plant: Bode plots ARG to outflow u_out,...
  no. compartments =',msprintf('%3.0f',Mx),...
     ' ,  Froude F = ',msprintf('%5.2f',F)]);

// step response
T_f = 1200.;  vvt = [0.: 1.0 : T_f];
 [vy]  = csim('step', vvt, H_u);
f3 = scf(3);
plot2d(vvt',[vy]',style=[2,5]);
legend('x_out','x_in',2); xgrid(1);
xtitle(['plant: step response to outflow  u_out,...
  no. compartments   =',msprintf('%3.0f',Mx),...
   ' ,  Froude F = ',msprintf('%5.2f',F)]);
```

Discussion for open loop frequency responses:

- By comparing with the Bode diagrams for the `sys_inf`-case, both the resonance peaks and the anti-resonance minima are no longer constant. The peaks are highest at the lowest resonance and for increasing frequency decrease with $-40dB/dec$. This must be due to the low pass filter effect of the basic second order oscillation element.

- Further the first resonance peak height increases for lower F-values, approx. with the ratio F_d/F.

- And it also rises with the number N, approx. with N/N_d, where $N\_d := 20$.

These findings agree with intuitive expectations. However they are only correlations from the simulations. A formal derivation is not attempted here.

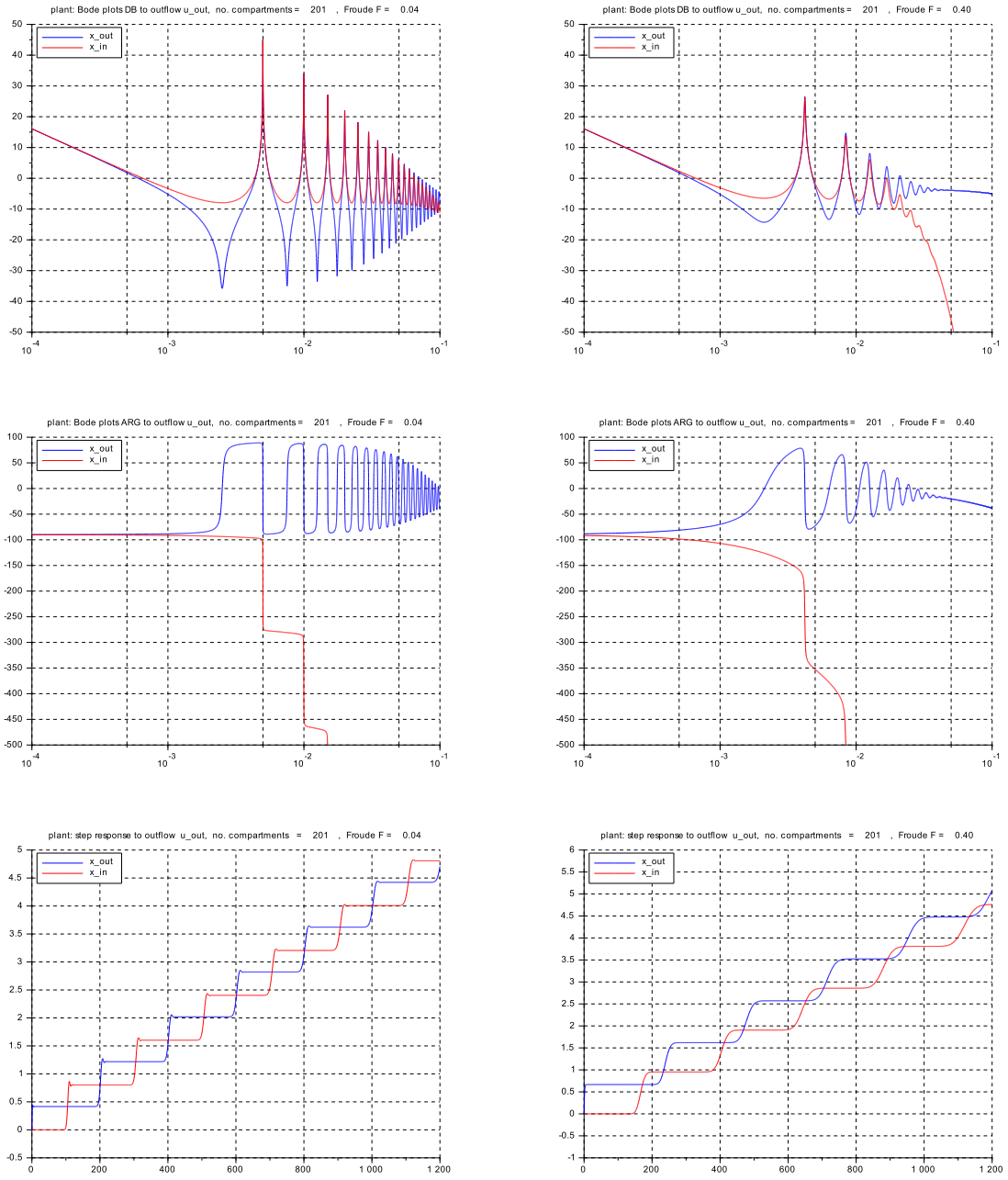


Figure 6.24: Responses of open loop system of h_{out} and h_{in} to input 'outflow' u_{out},
for spacial discretisation parameter $\mathbf{N = 100}$: $\rightarrow$ 201 compartments with length $L \approx 5m$
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses

Discussion for step responses:

- For $N = 100$; $F = 0.040$, and comparing with the `sysinf`-case, the responses have essentially the same step shape. For $F = 0.400$, a filter effect of second order appears with a period of $T \approx 220s$. This correlates with the lowest resonance peak.
- For $N = 20$; $F = 0.040$ a damped oscillation appears on the steps with $T \approx 20s$. Its period increases with time. For $F = 0.400$, the response reduces to the dominant oscillation with $T \approx 220s$.

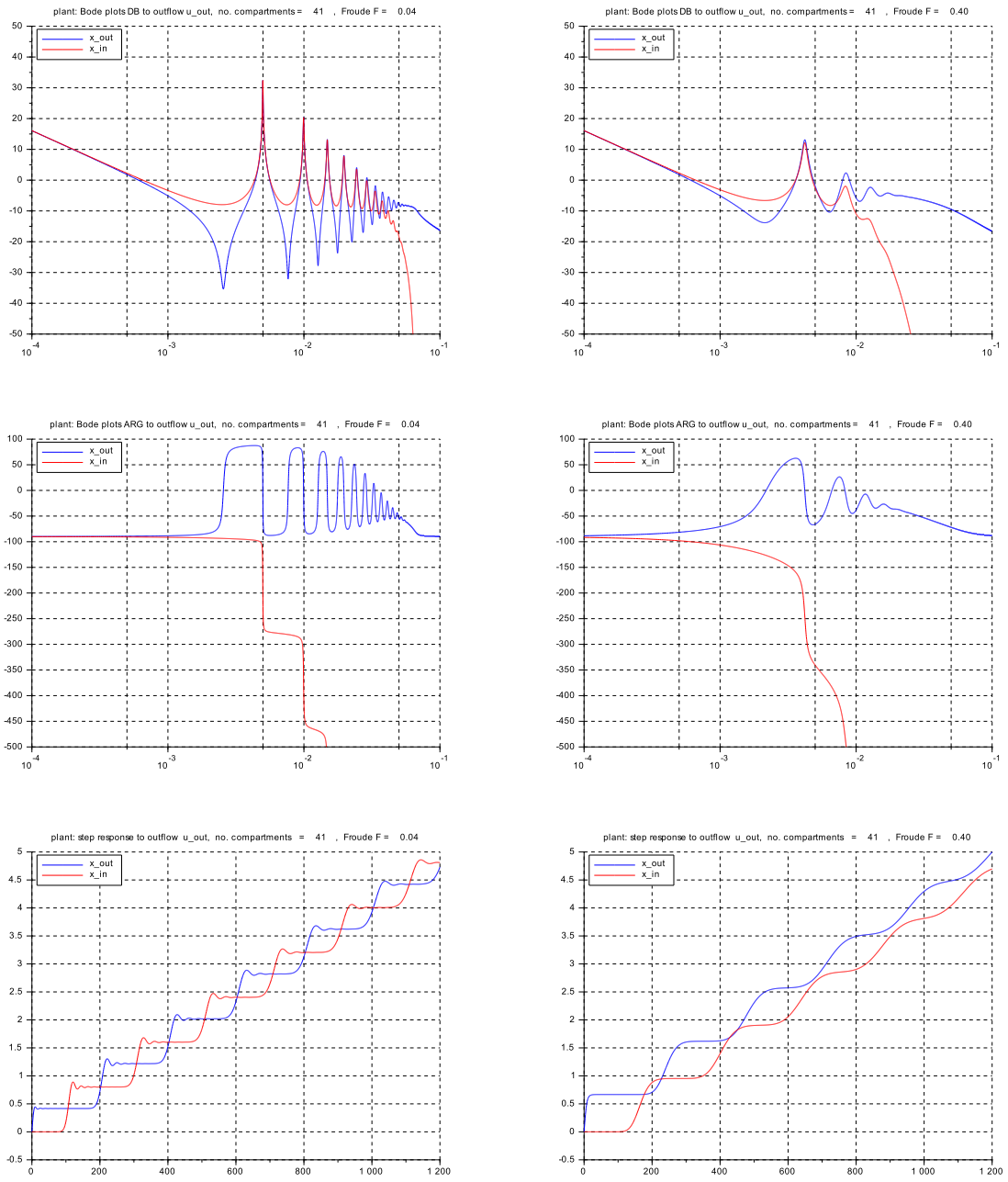


Figure 6.25: Responses of open loop system of h_{out} and h_{in} to input 'outflow' u_{out},
for spacial discretisation parameter $\mathbf{N = 20}$: $\rightarrow 41$ compartments with length $L \approx 25m$
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses

6.4.4 Control layout Case [A]

The system is extended directly to PcI-control. The integral action adds one element at position $Mx = 2N + 2$ of the state vector.

```
// ss_c6_02_1.sce
// Glf 2017_06_15
// control case [A] "industry standard"
// Closed loop  Bode plots, step responses
// PI_2dof-controller, gain Kp, Ki, T_i,
// I from x_out, cascaded to P from x_out
// manipulated var. u_out
// inputs: reference r_out and inflow z_in steps
// output traces: responses on x_out, x_in, u_out

stacksize('max');
// reference values
  g = 10.; U_d = 2.0; D = 2.5; L_tot = 500.;

// select flow speed / Froude number
//  U = 2.00;
    U = 0.20;

// no. of compartements
    N =  20; // nominal

 // building the system: SIZE  N, Mx
 //*********************************
      Mx = 2*N+2;   L = L_tot/(N+1);

// defining matrix elements
  U_F = sqrt(g*D);  F = U/U_F; F_d = U_d/U_F;
  PHI = F^2;  PHI_d = F_d^2;  T = (L/U_F);
  T_1 = T/F_d;   T_2 = T*(F_d);
  p = 2*F*F_d;  a = (1 - PHI); b = a;  c = 0.0;
  T_0 = 2*L_tot/U_F;

// Kp, Ki, T_i from sys_inf, with gain scheduling
    Kp = 1.0*(1/F_d)*(1+F); // impedance matching
//  Kp = 1.6*(1/F_d)*(1+F);
//  Kp = 2.4*(1/F_d)*(1+F);

    Ki =  0.80*(1+F);    T_i = T_0;

// time constants vector buildup
  vT = ones(1, Mx);
  for n= 1:2:Mx-1,
    vT(n) = T_1; vT(n+1) = T_2; end;
  vT(Mx-1) = T_1; vT(Mx) = T_i;

// system matrix AM buildup
   AM = zeros(Mx,Mx);
// Volume elements
  // first line
  m = 1; AM(m,m+1) = -1.0/vT(m);
  // intermediate non-even numbered lines
  for m = 3:2:Mx-3,
  AM(m,m-1)= +1.0/vT(m); AM(m,m+1)= -1.0/vT(m); end
  //final line for plant
  AM(Mx-1,Mx-2) = +1.0/vT(Mx-1);

//  line and column element for P-action
AM(Mx-1,Mx-1)= - Kp/vT(Mx-1);

// line and column elements for I-action
AM(Mx,Mx-1)= -Ki/vT(Mx); AM(Mx-1,Mx)= +1*Kp/vT(Mx-1);

// Flow/Momentum elements
// second line
  m = 2;
  AM(m,m-1) = +a/vT(m); AM(m,m) = -p/vT(m);
  AM(m,m+1) = -b/vT(m);
//all other even numbered lines
for m = 4:2:Mx-1,
  AM(m,m-3) = -c/vT(m); AM(m,m-2) = +p/vT(m);
  AM(m,m-1) = +a/vT(m); AM(m,m)   = -p/vT(m);
  AM(m,m+1) = -b/vT(m); end;

//Matrix BM for reference r_out
 BM_r = zeros(Mx,1); BM_r(Mx) = +1.0*(Ki/vT(Mx));

// Matrix CM
CM = zeros(3,Mx);
CM(1,Mx-1)=1.0; CM(2,1)=1.0; //for x_out and x_in
   CM(3,Mx-1) =  + Kp ;   // for trace u_out
   CM(3,Mx)   =  - Kp;    //     "
H_r=syslin('c',AM,BM_r,CM);

//  responses to outlevel reference r_out
//**************************************
f_min=0.00005; f_max=0.10; fstep=0.00001;
f_r=f_min:fstep:f_max;
FR_r = repfreq(H_r,f_r);

// Bode plot to r_out
  repf_r = repfreq(H_r,f_r);
  [dbB_1,phiB_1]=dbphi(repf_r(1,:));
  [dbB_2,phiB_2]=dbphi(repf_r(2,:));

f1 = scf(1);
plot2d(f_r,[dbB_1',dbB_2'],logflag = 'ln',...
style=[2,5],rect=[f_min,-30.,f_max,+10.0]); xgrid(1);
legend('x_out','x_in',2);
xtitle(['case[A]: Bode plots DB from r_out,...
 no. compartments =',msprintf('%3.0f',Mx-1),...
 ' ,  Froude F = ',msprintf('%5.2f',F)]);

f2 = scf(2);
plot2d(f_r,[phiB_1',phiB_2'],logflag = 'ln',...
style=[2,5],rect=[f_min,-200.,f_max,+100.0]);xgrid(1);
legend('x_out','x_in',2);
xtitle(['case[A] Bode plots PHI from r_out,...
   no. compartments =',msprintf('%3.0f',Mx-1),...
     ' ,  Froude F = ',msprintf('%5.2f',F)]);

// step response to r_out with u(t) trace
T_f=1200.; vvt=[0.:1.0:T_f]; [vy]=csim('step',vvt,H_r);

f3 = scf(3); plot2d(vvt',[vy]',style=[2,5,1],...
rect=[0.,-1.8,T_f,+1.2]);legend('x_out','x_in','u_out',4);
xtitle(['case[A]: step response to ref. r_out,...
 no. compartments   =',msprintf('%3.0f',Mx-1),...
' ,  Froude F = ',msprintf('%5.2f',F)]);xgrid(1);

// responses to inflow disturbance z_in
//*************************************
 BM_z=zeros(Mx,1);
 BM_z(1)= +(1.0/vT(1)); BM_z(2)= +p*(1.0/vT(2));
 H_z = syslin('c',AM,BM_z,CM);

 // Bode plot
  repf_z = repfreq(H_z,f_r);
  [dbBd_1,phiBd_1]=dbphi(repf_z(1,:));
  [dbBd_2,phiBd_2]=dbphi(repf_z(2,:));

f4 = scf(4);
plot2d(f_r,[dbBd_1',dbBd_2'],logflag = 'ln',...
style=[2,5],rect=[f_min,-30.,f_max,+10.0]);
  xgrid(1); legend('x_out','x_in',2);
  xtitle(['case[A]: Bode plots  DB,  from z_in,...
  no. compartments   =',msprintf('%3.0f',Mx-1),...
  ' , Froude F = ',msprintf('%5.2f',F)]);

f5 = scf(5);
plot2d(f_r,[phiBd_1',phiBd_2'],logflag = 'ln',...
  style=[2,5],rect=[f_min,-200.,f_max,+100.0]);
  xgrid(1); legend('x_out','x_in',2);
  xtitle(['case[A]: Bode plots PHI, from z_in ,...
  no. compartments   =',msprintf('%3.0f',Mx-1),...
  ' , Froude F = ',msprintf('%5.2f',F)]);

// step response
T_f = 1200.; vvt=[0.:1.0:T_f]; [vy]=csim('step',vvt,H_z);

f6 = scf(6);
plot2d(vvt',[vy]',style=[2,5,1],rect=[0.,-1.0,T_f,+2.0]);
legend('x_out','x_in','u_out',1);
xtitle(['case[A]: step response to inflow z_in,...
  no. compartments   =',msprintf('%3.0f',Mx-1),...
' ,  Froude F = ',msprintf('%5.2f',F)]); xgrid(1);
```

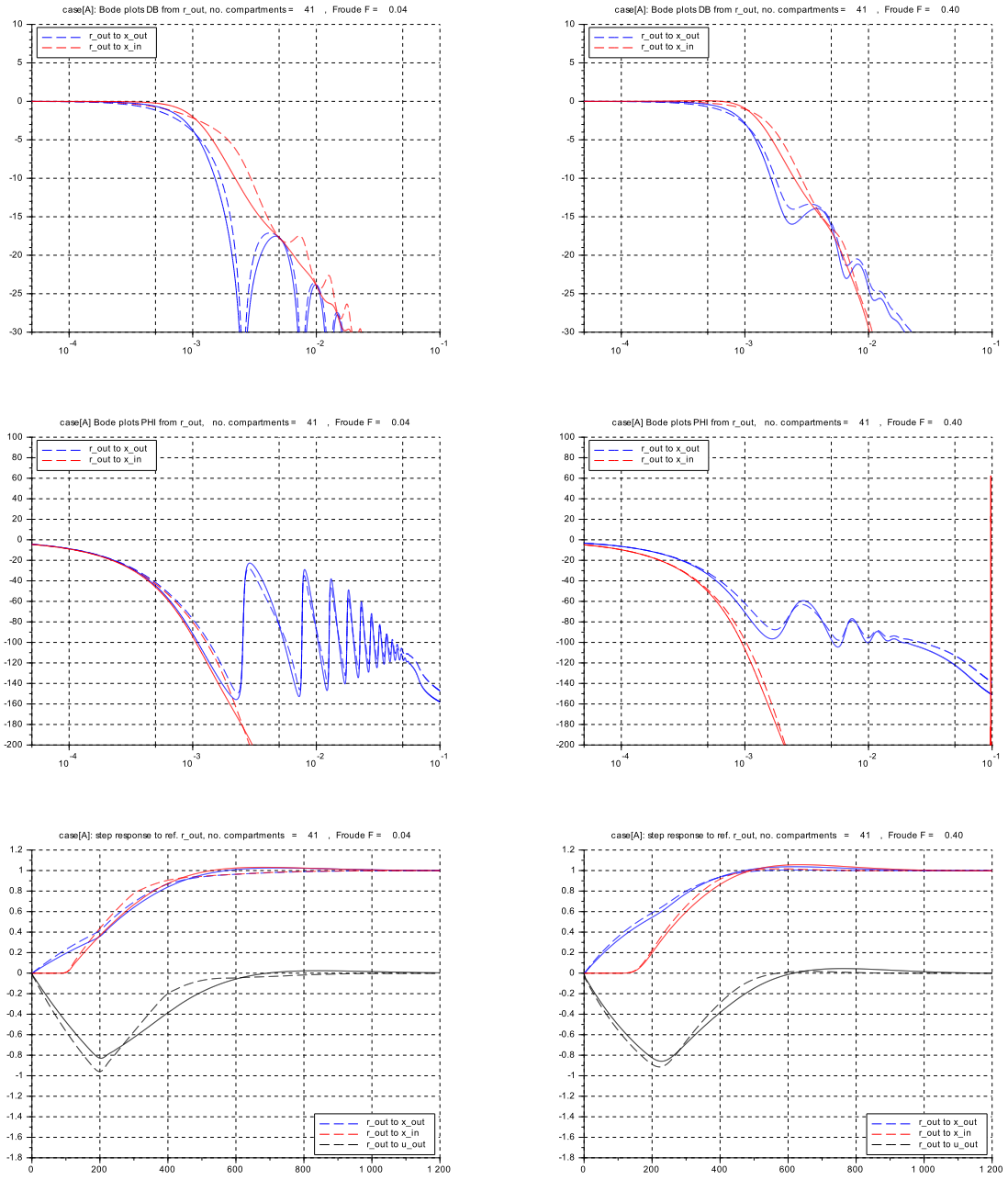


Figure 6.26: Case [A]: Responses of closed loop system (PcascI) to reference r_out,
for spacial discretisation parameter $N = 20$: $\rightarrow 41$ compartments
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses
(solid) for $Kp = 1.00 * (1/F_d) * (1 + F)$ (dashed) for $Kp = 1.60 * (1/F_d) * (1 + F)$

Discussion:
The frequency response shows the same bandwidth as for the `sysinf`-case Fig.6.15, but then decays faster. And the step response is very close to Fig.6.17 with the same settling time $\approx 600s (\approx 3T_E)$ and the same near optimal damping. Note that at low flow $F = 0.040$ and at higher Kp-values (see .sce-file), the damping decreases sharply, but not so at design flow $F = 0.400$ (simulation results omitted here).

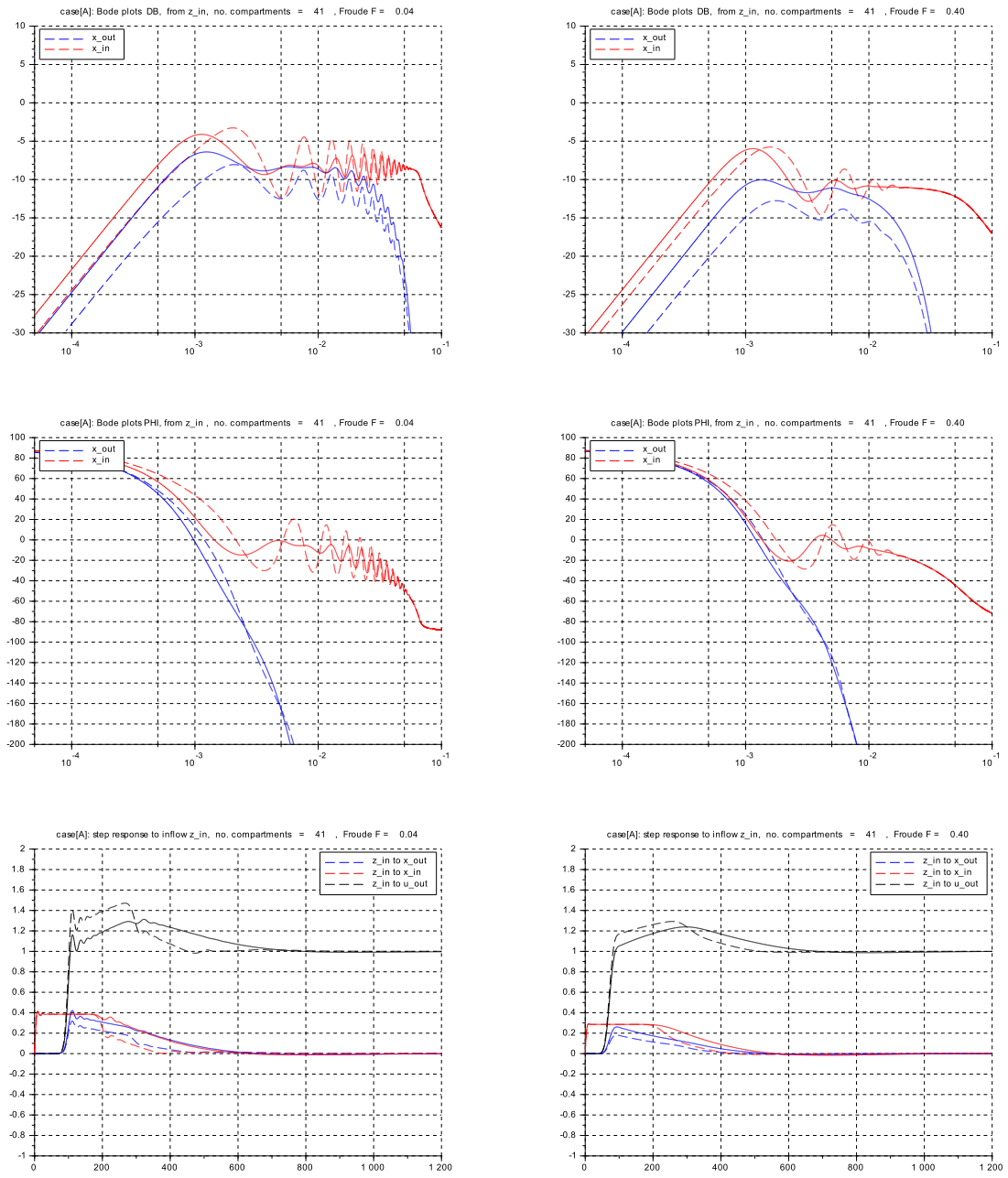


Figure 6.27: Case [A]: Responses of closed loop system (PcascI) to inflow z_in,
for spacial discretisation parameter $N = 20$: $\rightarrow 41$ compartments
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses
(solid) for $Kp = 1.00 * (1/F_d) * (1 + F)$ (dashed) for $Kp = 1.60 * (1/F_d) * (1 + F)$

Discussion:
The same findings apply. - For practical applications the maximum deviation of the controlled variable (outflow level) is often specified. For $F = 0.040$ and $Kp = 1.00 * (1/F_d) * (1 + F)$ it is ≈ 0.39 (equal to the height of the downstream Froude-wave), and reduces to 0.28 (that is approx 72%) for $Kp = 1.60 * (1/F_d) * (1 + F)$. And the outflow peak overshoot rises from approx 0.30 to 0.50[8].

[8] In a cascade of run-of-river plants with individual level control this will lead to stronger downstream 'surge flow pileup'

6.4.5 Control layout Case [B]

```
// s_c6_02_2.sce
// Glf 2017_06_15
// control case [B] "upstream level control by outflow"
// Closed loop: Bode plots, step responses
// PI-2dof-controller, gain Kp, Ki, T_i
// split-action: I from x_in, cascaded to P from x_out
// manipulated variable: u_out
// inputs: reference r_in and inflow z_in steps
// output traces: responses on x_out and x_in, u_out

// reference values
  g = 10.; U_d = 2.0; D = 2.5; L_tot = 500.;

// operating point: select flow speed / Froude number
//    U = 2.0;
      U = 0.20;

// no. of compartements
     N = 20;

 // building the system: SIZE  N, Mx
 //*********************************
      Mx = 2*N+2;   L = L_tot/(N+1);

// defining matrix elements
  U_F = sqrt(g*D);  F = U/U_F; F_d = U_d/U_F;
  PHI = F^2;  PHI_d = F_d^2;
  T_1 = (L/U_F)*(1/F_d);   T_2 = (L/U_F)*(F_d);
  p = 2*F*F_d;  a = (1 - PHI); b = a;  c = 0.0;
  T_t_up = L_tot/(U_F - U); //delay upstream direct.
  T_0 = 2*(L_tot/U_F);

// PI-controller from sys_inf, with gain scheduling
     Kp = 1.00*(1/F_d)*(1+F);  // impedance matching
//   Kp = 1.26*(1/F_d)*(1+F);
//   Kp = 1.60*(1/F_d)*(1+F);

     Ki = 1.0*(1 - F);   Ti = T_0;

// time constants vector buildup
  vT = ones(1, Mx);
  for n= 1:2:Mx-1, vT(n)=T_1; vT(n+1)=T_2; end;
  vT(Mx-1) = T_1; vT(Mx) = Ti;

// system matrix AM buildup
   AM = zeros(Mx,Mx);
// Volume elements, first line
m = 1; AM(m,m+1) = -1.0/vT(m);
  // intermediate non-even numbered lines
  for m = 3:2:Mx-3,
AM(m,m-1)=+1.0/vT(m); AM(m,m+1)=-1.0/vT(m); end
  //final line for plant
AM(Mx-1,Mx-2) = +1.0/vT(Mx-1);

// line and column element for P-action
AM(Mx-1,Mx-1)= - Kp/vT(Mx-1);

// line and column elements for I-action
AM(Mx,1)=-Ki/vT(Mx); AM(Mx-1,Mx)=Kp/vT(Mx-1);

// Flow/Momentum elements, second line
m = 2; AM(m,m-1)= +a/vT(m); AM(m,m)= -p/vT(m);
       AM(m,m+1) = -b/vT(m);
//all other even numbered lines
for m = 4:2:Mx-1,
  AM(m,m-3) = -c/vT(m); AM(m,m-2) = +p/vT(m);
  AM(m,m-1) = +a/vT(m); AM(m,m)   = -p/vT(m);
  AM(m,m+1) = -b/vT(m); end;

//Matrix BM_r
 BM_r = zeros(Mx,1); BM_r(Mx) = +Ki/vT(Mx);

// Matrix CM
CM = zeros(3,Mx);
CM(1,Mx-1)=1.0; CM(2,1)=1.0;//for x(out),x(in)
CM(3,Mx-1) = Kp; CM(3,Mx) = -Kp; //for u_out
     H_r=syslin('c',AM,BM_r,CM);

//  responses to reference r_in
//************************************
f_min=0.00005; f_max=0.08; fstep=0.000010;
f_r=(f_min:fstep:f_max); f_rp = f_r;
FR_r = repfreq(H_r,f_r);

// Bode plot to r_in
  repf_r = repfreq(H_r,f_r);
  [dbB_1,phiB_1]=dbphi(repf_r(1,:));
  [dbB_2,phiB_2]=dbphi(repf_r(2,:));

f1 = scf(1);
plot2d(f_rp,[dbB_1',dbB_2'],logflag = 'ln',...
  style=[2,5],rect=[f_min,-30.,0.10,+10.0]);
  xgrid(1); legend('x_out','x_in',2);
xtitle(['case[B]: PcI-2dof, Bode DB from r_in,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

f2 = scf(2);
plot2d(f_rp,[phiB_1',phiB_2'],logflag = 'ln',...
  style=[2,5],rect=[f_min,-400.,0.10,+0.0]);
  xgrid(1); legend('x_41','x_1',1);
xtitle(['case[B]: PcI-2dof, Bode PHI from r_in,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

// step response with u(t) trace
 T_f = 1200; delT = 1.0;
vvt=[0.:delT:T_f]; [vy]=csim('step',vvt,H_r);

f3 = scf(3);
plot2d(vvt',[vy]',style=[2,5,1],rect=[0.,-1.8,T_f,+1.40]);
legend('x_out','x_in','u_out',1); xgrid(1);
xtitle(['case[B]: PcI-2dof,step to ref. r_in,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

// responses to inflow disturbance z_in
//************************************
 BM_z=zeros(Mx,1);
 BM_z(1)= +(1.0/vT(1)); BM_z(2)= +p*(1.0/vT(2));
 H_z = syslin('c',AM,BM_z,CM);

 // Bode plot
  repf_z = repfreq(H_z,f_r);
  [dbBd_1,phiBd_1]=dbphi(repf_z(1,:));
  [dbBd_2,phiBd_2]=dbphi(repf_z(2,:));

f4 = scf(4);
plot2d(f_rp,[dbBd_1',dbBd_2'],logflag = 'ln',...
    style=[2,5],rect=[f_min,-30.,0.10,+10.0]);
  xgrid(1); legend('x_out','x_in',2);
  xtitle(['case [B]: PcI-2dof, Bode  DB from z_in,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

f5 = scf(5);
plot2d(f_rp,[phiBd_1',phiBd_2'],logflag = 'ln',...
    style=[2,5],rect=[f_min,-200.,0.10,+100.0]);
  xgrid(1); legend('x_out','x_in',1);
  xtitle(['case [B]: PcI-2dof, Bode PHI from z_in,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

// step response
  [vy]=csim('step',vvt,H_z);

f6 = scf(6);
plot2d(vvt',[vy]',style=[2,5,1],rect=[0.,-0.40,T_f,+2.0]);
legend('x_out','x_in','u_out',1);  xgrid(1);
xtitle(['case [B]: PcI-2dof, step to inflow z_in,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);
```

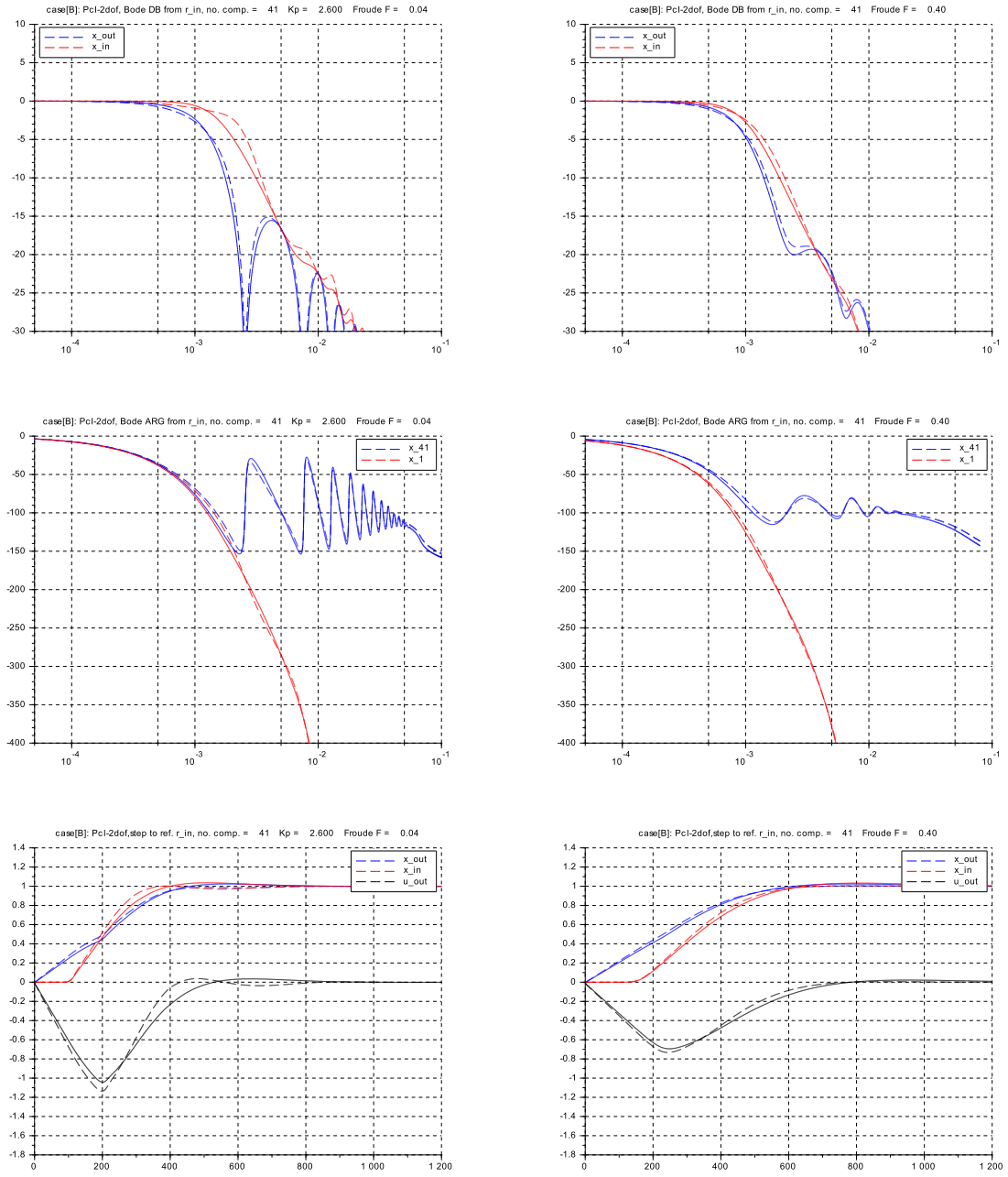


Figure 6.28: Case [B]: Responses of closed loop system (PcascI) to reference r_in,
for spacial discretisation parameter $N = 20$: $\rightarrow 41$ compartments
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses
(solid) for $Kp = 1.00 * (1/F_d) * (1 + F)$ (dashed) for $Kp = 1.26 * (1/F_d) * (1 + F)$

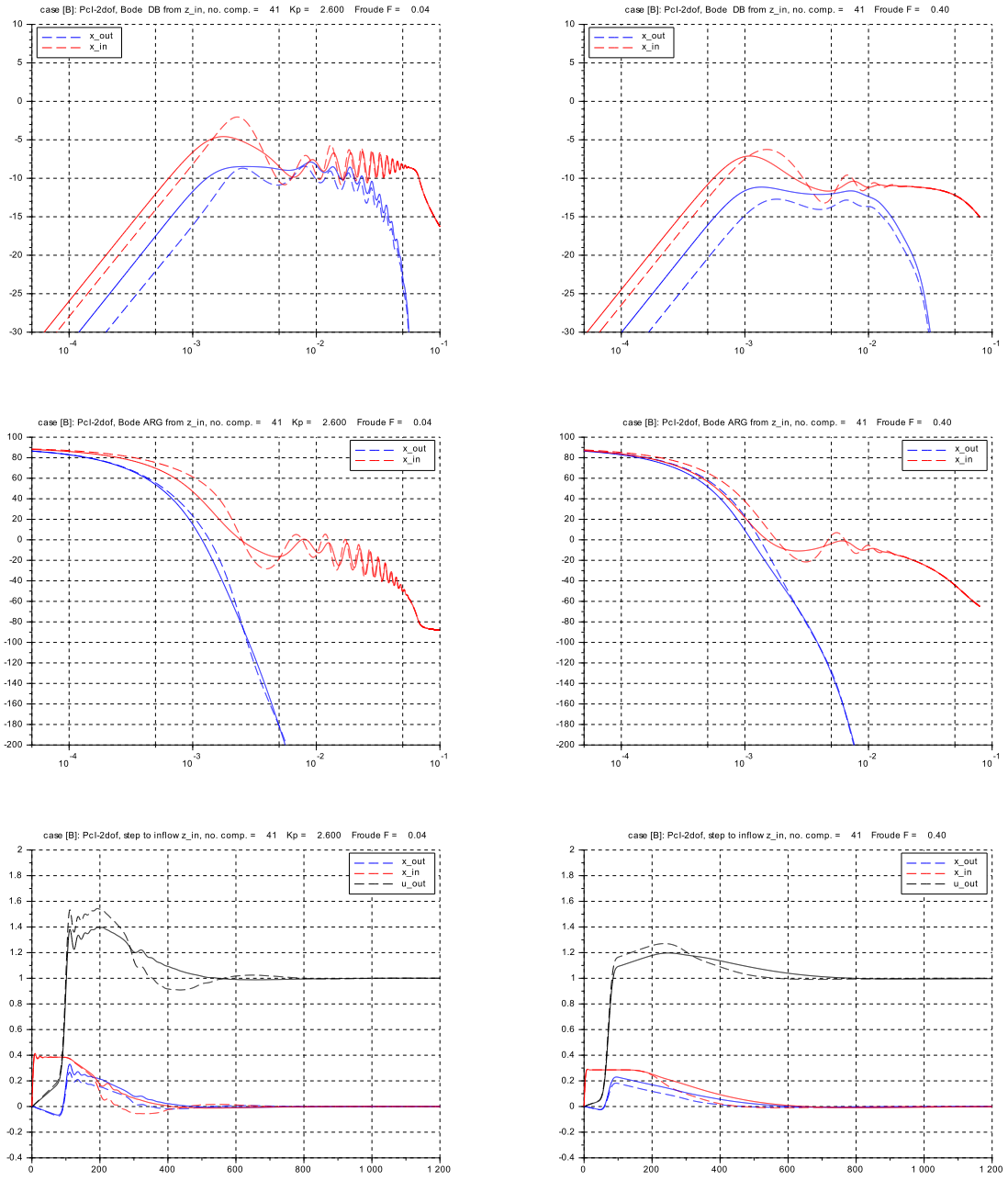


Figure 6.29: Case [B]: Responses of closed loop system (PcI) to inflow z_in,
for spacial discretisation parameter $N = 20$: $\rightarrow 41$ compartments
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses
(solid) for $Kp = 1.00 * (1/F_d) * (1 + F)$ (dashed) for $Kp = 1.26 * (1/F_d) * (1 + F)$

Discussion:
From an application point of view, by inspection the transient responses are sufficiently close to those for the `sysinf`-case, Fig.6.20. That is the findings from there are valid here as well.

6.4.6 Control layout Case [C]

```
// s_c6_02_3.sce
// Glf 2017_06_15
// control case [C]: "irrigation channel loop layout"
// Closed loop: Bode plots, step responses
// PI-2dof-controller, gain Kp, Ki, T_i
// split-structure: I from outflow level, P from inflow level
// manipulated variable: inflow
// inputs: reference r_out and outflow z_out steps
// output traces: responses on x_out and x_in, u_in

// reference values
  g = 10.; U_d = 2.0; D = 2.5; L_tot = 500.;

// operating point: select flow speed / Froude number
//    U = 2.0;
      U = 0.20;

//  no. of compartements
     N = 20;

 // building the system: SIZE  N, Mx
 //*********************************
      Mx = 2*N+2;   L = L_tot/(N+1);

// defining matrix elements
  U_F = sqrt(g*D);  F = U/U_F; F_d = U_d/U_F;
  PHI = F^2;  PHI_d = F_d^2;
  T_1 = (L/U_F)*(1/F_d);   T_2 = (L/U_F)*(F_d);
  p = 2*F*F_d;  a = (1 - PHI); b = a;  c = 0.0;
  T_t_dn = L_tot/(U_F + U); //delay downstream direct.
  T_0 = 2*(L_tot/U_F);

// PI-controller from sys_inf, with gain scheduling
     Kp = 1.0*(1/F_d)*(1-F);  // impedance matching
//   Kp = 1.26*(1/F_d)*(1-F);

     Ki = 1.0*(1+F);  Ti = T_0;

// time constants vector buildup
  vT = ones(1, Mx);
  for n= 1:2:Mx-2, vT(n)=T_1; vT(n+1)=T_2; end
    vT(Mx-1) = T_1;    vT(Mx)   = Ti;

// system matrix AM buildup
   AM = zeros(Mx,Mx);
// volume elements
// first line
m = 1; AM(m,m+1) = -1.0/vT(m);
// intermediate non-even numbered lines
  for m = 3:2:Mx-3,
AM(m,m-1)=+1.0/vT(m);  AM(m,m+1)=-1.0/vT(m); end
//final line for plant
m = Mx-1; AM(m,m-1) = +1.0/vT(m);

// line and column element for P-action
  AM(1,1) = - Kp/vT(1);

// line and column elements for I-action
AM(Mx,Mx-1)=-Ki/vT(Mx); AM(1,Mx)=Kp/vT(1);

// Flow/Momentum elements
// second line
m = 2;AM(m,m-1)=(+a - Kp*p)/vT(m); AM(m,m)=-p/vT(m);
AM(m,m+1) = -b/vT(m); AM(m,Mx) = + Kp*p/vT(m);
//all other even numbered lines
for m = 4:2:Mx-1,
  AM(m,m-3) = -c/vT(m); AM(m,m-2) = +p/vT(m);
  AM(m,m-1) = +a/vT(m); AM(m,m) = -p/vT(m);
  AM(m,m+1) = -b/vT(m);  end

 //Matrix BM_r
  BM_r = zeros(Mx,1); BM_r(Mx)=+(Ki/vT(Mx));

// Matrix CM
CM = zeros(3,Mx);
CM(1,Mx-1)= 1.0; CM(2,1)= 1.0;//for x(out),x(in)
CM(3,1)= - Kp; CM(3,Mx)= + Kp; // for trace u_in

     H_r=syslin('c',AM,BM_r,CM);


//  responses to reference r_out
//*********************************
f_min=0.00005;f_maxf=0.080;f_max=0.10; fstep=0.00001;
f_r=f_min:fstep:f_maxf; FR_r = repfreq(H_r,f_r);

// Bode plot to r_out
  repf_r = repfreq(H_r,f_r);
  [dbB_1,phiB_1]=dbphi(repf_r(1,:));
  [dbB_2,phiB_2]=dbphi(repf_r(2,:));

f1 = scf(1);
plot2d(f_r,[dbB_1',dbB_2'],logflag = 'ln',...
  style=[2,5],rect=[f_min,-30.,f_max,+10.0]);
xgrid(1); legend('x_out','x_in',2);
xtitle(['case[C]: PcI-2dof, Bode DB from r_out,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

f2 = scf(2);
plot2d(f_r,[phiB_1',phiB_2'],logflag = 'ln',...
  style=[2,5],rect=[f_min,-400.,f_max,+0.0]);
  xgrid(1); legend('x_out','x_in',1);
xtitle(['case[C]: PcI-2dof, Bode PHI from r_out,...
 no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F = ',msprintf('%5.2f',F)]);

 T_f = 1200; delT = 1.0;
vvt=[0.:delT:T_f]; [vy]=csim('step',vvt,H_r);

f3 = scf(3);
plot2d(vvt',[vy]',style=[2,5,1],rect=[0.,-0.8,T_f,+2.0]);
legend('x_out','x_in1','u_in',1);
xgrid(1); xtitle(['case[C]: PcI-2dof,step to ref. r_out,...
no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

// responses to outflow disturbance z_out
//*********************************
 BM_z=zeros(Mx,1);
 BM_z(Mx-1)= -(1.0/vT(Mx-1));
 H_z = syslin('c',AM,BM_z,CM);

 // Bode plot
  repf_z = repfreq(H_z,f_r);
  [dbBd_1,phiBd_1]=dbphi(repf_z(1,:));
  [dbBd_2,phiBd_2]=dbphi(repf_z(2,:));

f4 = scf(4);
plot2d(f_r,[dbBd_1',dbBd_2'],logflag = 'ln',...
    style=[2,5],rect=[f_min,-30.,f_max,+10.0]);
  xgrid(1); legend('x_out','x_in',2);
  xtitle(['case[C]: PcI-2dof, Bode DB from z_out...
, no. comp. =',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

f5 = scf(5);
plot2d(f_r,[phiBd_1',phiBd_2'],logflag = 'ln',...
    style=[2,5],rect=[f_min,-400.,f_max,+0.0]);
  xgrid(1); legend('x_out','x_in',2);
xtitle(['case[C]: PcI-2dof, Bode ARG from z_out...
, no. comp.=',msprintf('%3.0f',Mx-1),...
'Froude F =',msprintf('%5.2f',F)]);

// step response
  [vy]=csim('step',vvt,H_z);

f6 = scf(6);
plot2d(vvt',[vy]',style=[2,5,1],rect=[0.,-1.4,T_f,+2.0]);
legend('x_out','x_in','u_1',1);  xgrid(1);
xtitle(['case[C]: PcI-2dof, step to outflow z_out,...
, no. comp. =',msprintf('%3.0f',Mx-1),...
',  Froude F = ',msprintf('%5.2f',F)]);
```

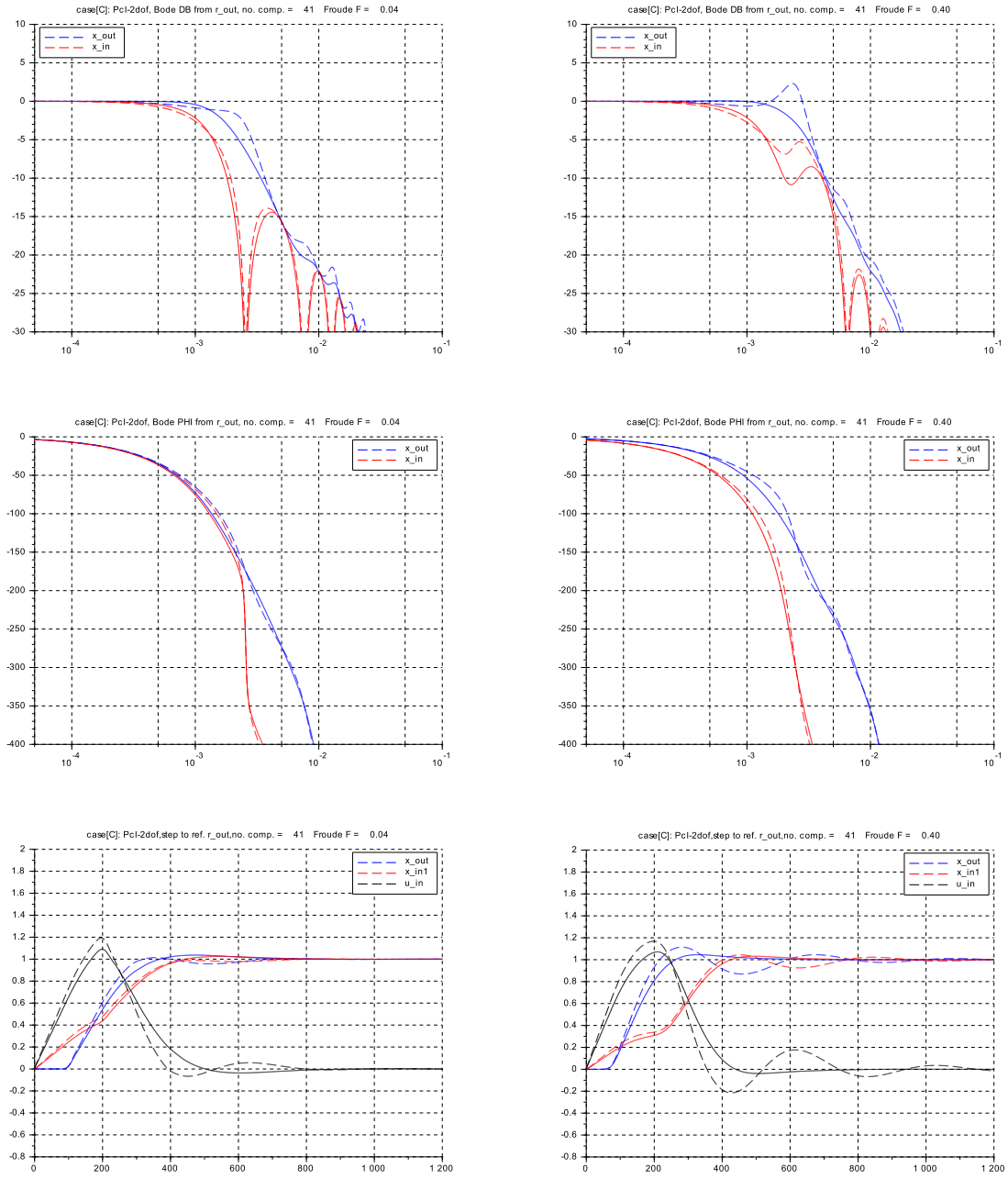


Figure 6.30: Case [C]: Responses of closed loop system (PcascI) to reference r_out,
for spacial discretisation parameter $N = 20$: $\rightarrow 41$ compartments
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses
(solid) for $Kp = 1.00 * (1/F_d) * (1 + F)$ (dashed) for $Kp = 1.26 * (1/F_d) * (1 + F)$

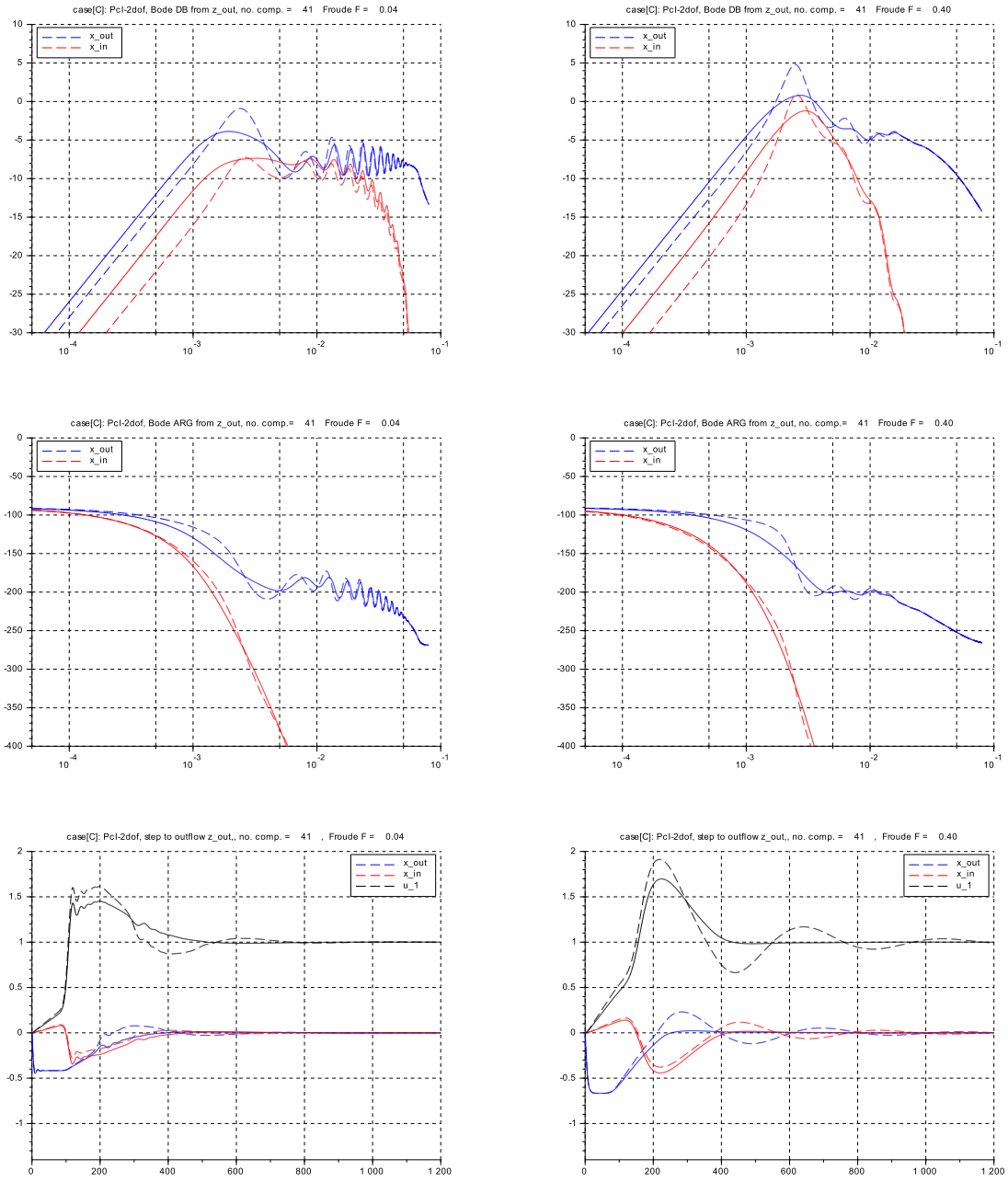


Figure 6.31: Case [C]: Responses of closed loop system (PcI) to inflow *z_out*,
for spacial discretisation parameter $N = 20$: $\rightarrow$ 41 compartments
(left) for $F = 0.040$, (right) for $F = F_d = 0.400$
(top, center) Bode plots (bottom) step responses
(solid) for $Kp = 1.00 * (1/F_d) * (1 + F)$ (dashed) for $Kp = 1.60 * (1/F_d) * (1 + F)$

Discussion:
Again the transient responses are sufficiently close to those for the `sysinf`-case, Fig.6.23. Thus the same findings apply.

6.5 Control Design for the nonlinear time domain model

6.5.1 Motivation

This is to be the last verification step before applying the control design procedure to real plant situations. The focus is on the effect of GMS-friction, with the bottom slope at reference inclination (for design flow and depth) and for different flows but at reference depth at the controlled location.

A typical real world application case would be (among many others) the Channel Hydro Power Station 'Wildegg-Brugg' on the lower Aare river (WGS 84 location: 47.468993 8.170052).

6.5.2 Modelling

The process model is carried over from sect.3.3, the case `s_c3_41_00` with the basic geometry 'constant cross section' (constant width $B(k)$ and nominally horizontal bottom $S(k) = const.$. The bottom inclination is then calculated from the GMS formula to obtain constant (reference) depth D_r for the reference flow Q_r.

Now the model variables are given in SI-units (and not in p.u. as in the previous section). In order to apply the control design procedure directly the reference values of flow Q_ref and depth D_ref are used for sensor scaling and actuator scaling

$$sensor\_gain = 1.00/D\_r \quad \text{and} \quad actuator\_gain = Q\_r/1.00$$

Note that this assumes that the design location depth is used directly for the sensor input (as in the previous sections) and not the measured level there and also with no sensor zero offset [9].
And the integral action time scaling is set here to the (artificial) echo travelling time at zero flow and reference depth $Ti0 := 2 \cdot (L\_tot/\sqrt{g \cdot D\_r})$. Note that this assumes that the bottom geodesic height difference from the inflow location to the outflow location is small compared to D_ref.

The flow now may vary continuously over the whole operating range. Here it shall cover the range from $Q = Q\_r = 50\ m^3/s$ down to $Q = 0.10 \cdot Q\_r$, while D is kept at D_r. This results in F from $F = F\_r = 0.400$ down to $F = 0.040$. The speed of the these large scale flow variations usually is very low. A typical rate for a medium size run-of-river plant would be $\approx \pm 5\%$ of Q_r per one hour. — Here this is set to 'per $180s$ or $3min$' that is a factor of 20 faster (to keep the simulation time within bounds). Note that this will produce tracking errors in the control loop simulation which are greater than what is to be expected in real installations.
Superposed to this large scale 'drift' of the flow, there will be small local changes of flow due to turbine runup/shutdown not fully compensated by the parallel weir flow, filling/emptying of locks, etc. They are assumed to be 'steps' of size $\pm 4\%$ of $Q\_r = 2.0\ m^3/s$. They shall occur at $Q = Q\_r$ and at $Q = 0.10 \cdot Q\_r$ during 'steady state' conditions (again keeping in mind that the underlying large scale flow drift is much slower in reality).

Next the open loop step responses of the plant are simulated and compared to the results from the two previous models. Then the closed loop responses for the control cases [A], [B] and [C] are investigated.

[9] as in most real plant applications

6.5.3 Open loop responses

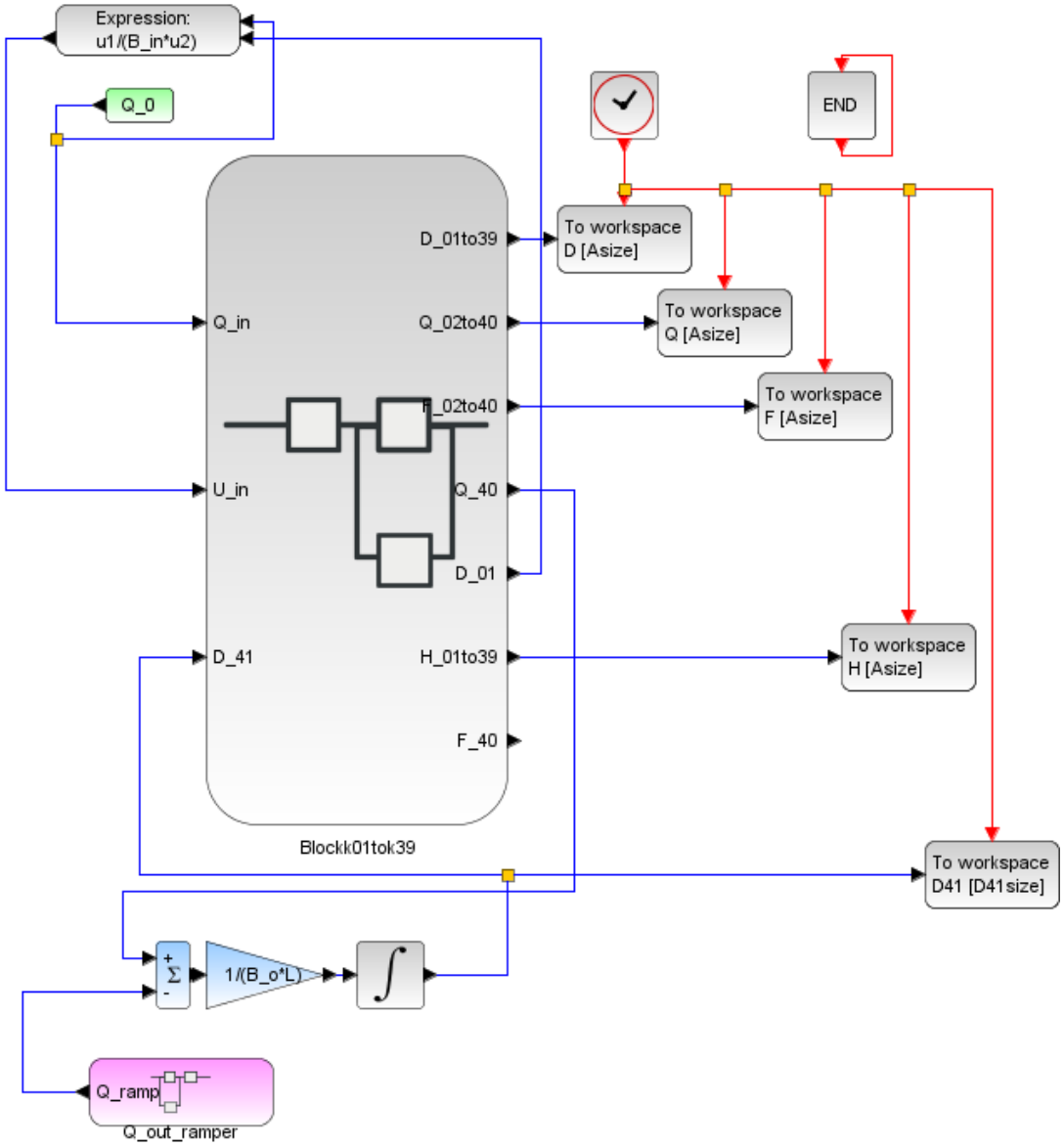


Figure 6.32: Top level diagram for open loop step responses to outflow *z_out*

The single outflow 'step' is set to -2% that is $-1\ m^3/s$ at $50\ s$. The simulation runs for $1200\ s$ that is for ≈ 6 times the echo travel time at zero flow.

```
// s_c6_03_0_context
// open loop response to outflow step
// Glf  2017_06_15

N= 20;      // no Volume and Momentum-segments
g = 10.; L_tot = 500.; L = L_tot/(N+1);
kap = 0.0;  // "downwind"

// operating point
   Q_d = 50.0; D_d = 2.50;  B_d = 10.0; S_d = 1*(-1.0);
   U_d = 2.0;  U_F_d = 5.0; F_d = 0.40;

// GMS-friction coefficient
// k_s =  50.0;
   k_s = 100.0;

// reference bottom slope
  U_d =  2.0;  R_d = (B_d*D_d)/(B_d + 2*D_d);
  I_d = (U_d/(k_s*(R_d)^(2/3)))^2;

//working spans for integrators
 D_min = +0.001; D_max = 4*D_d;
 Q_min = +0.001; Q_max = 4*Q_d;

// channel geometry
//*****************
// Basic layout: constant width
vb = 1.0*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
          1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0];
vB = B_d*vb;

//  basic layout: bottom 'horizontal'
vs  = -1*[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
            1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = (-S_d)*vs;

// inflow fixed
// Q_0 = Q_d;
   Q_0 = 0.10*Q_d;
```

```
// outflow generation
dQ = 1.0; // -> 1.0 m^3/s, or 2 % of Q_d
t_st_1 = 50.0;   r_1_0 = Q_0; r_1_1 = Q_0 - dQ;
t_st_2 = 1200.0; r_2_0 = 0.;  r_2_1 = +2*dQ;
t_st_3 = 1200.0; r_3_0 = 0.;  r_3_1 = -1.0*dQ;
T_fin = 1200.;
// outflow slew rate
g_st = 10.0; u_up_st = +1.0; u_dn_st = -1.0;
tau_st = 10.0; Q_st_0 = Q_0;

vd0 = ones(1,(2*N+1)); vD0 = D_d*vd0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// friction slope of the bottom vS
vi0    = ones(1,(2*N+1));
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vI0(kk)    = I_d*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4); end
// shift bottom zero to outflow point
    Sf_o = vSf(kk+1);
for k5= 1:1:(2*N+1),
    vS(k5)  = vS(k5)  - Sf_o*vi0(k5); end

// initial depths
if Q_0 <= 0.15*Q_d then
   vH0 = (D_d + S_d)*ones(1,(2*N+1));
for k6 = 1:1:(2*N+1),
      vD0(k6)  = vH0(k6) - vS(k6); end
else  vD0 = D_d*vd0;  end

// Inflow and outflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);
B_o=vB(kk+1);S_o=vS(kk+1);D_o=vD0(kk+1);Q_o=vQ0(kk+1);

// Data transfer to Plots
CC=21; CN=1200; delT=T_fin/CN; Asize=1.01*CC*CN;
C41 = 2; D41size = 1.01*C41*CN;
```

```
// s_c6_03_0_crunplot
//  Glf  2017_06_15

stacksize('max'); exec('s_c6_03_0_context.sce', -1);
importXcosDiagram('s_c6_03_0.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:1,  clf(kfig); end

vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

if Q_0 <= 0.10*Q_d then

  if k_s <= 60 then
```

```
    f1 = scf(1);
    plot2d(D.time,D.values,vcolor,rect=[0.,2.1,T_fin,2.7]);
    plot2d(D41.time,D41.values,rect=[0.,2.1,T_fin,2.7]);
    xtitle("case [o]: D_1 to D_41"); xgrid(1);
  else
    f1 = scf(1);
    plot2d(D.time,D.values,vcolor,rect=[0.,2.3,T_fin,2.7]);
    plot2d(D41.time,D41.values,rect=[0.,2.3,T_fin,2.7]);
    xtitle("case [o]: D_1 to D_41"); xgrid(1);
  end

else
f1 = scf(1);
plot2d(D.time,D.values,vcolor,rect=[0.,2.4,T_fin,2.8]);
plot2d(D41.time,D41.values,rect=[0.,2.4,T_fin,2.8]);
xtitle("case [o]: D_1 to D_41"); xgrid(1);
end
```

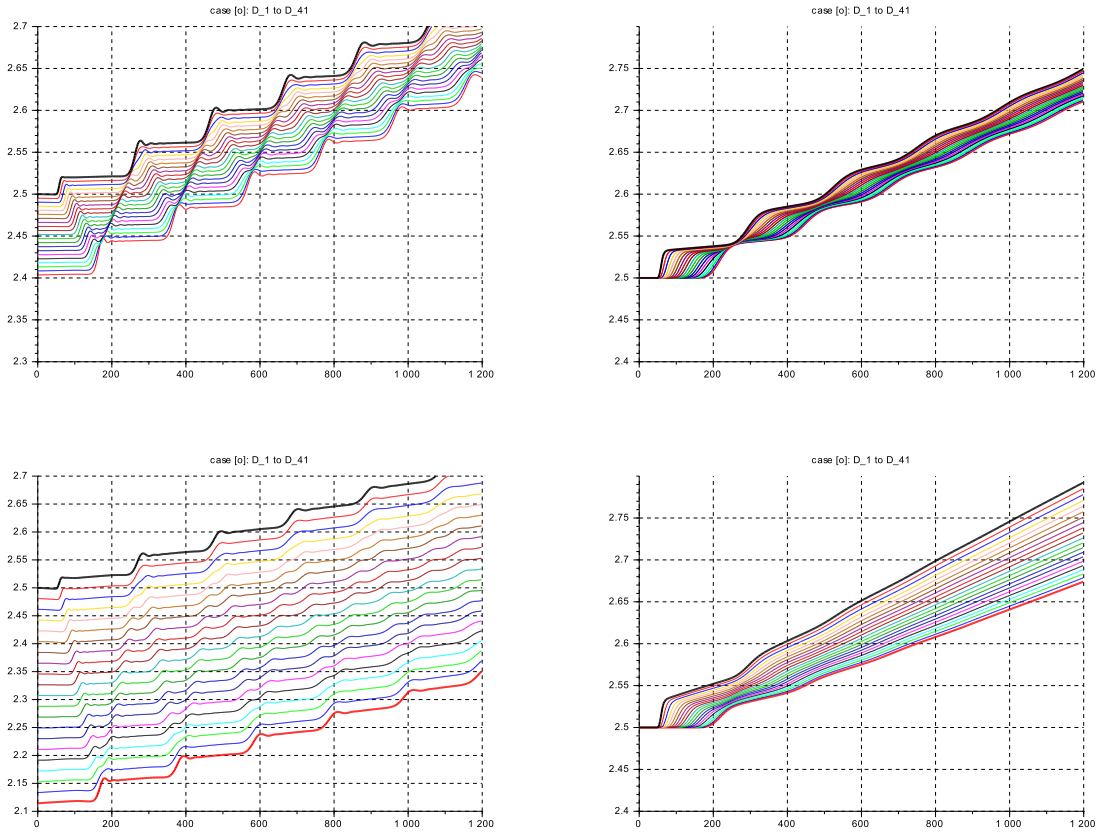


Figure 6.33: open loop step responses of depth D to outflow z_out,
(left column) for $F = 0.040$ (right column) for $F = 0.400$,
(top row) for $k\_s = 100$ (bottom row) for $k\_s = 50$

Discussion:

- top left: very similar to observed open loop responses of the sysinf-model.
The Froude waves are very distinct, dissipation is low.
There is now a wave slope, highest on the first step, decreasing for later reflections and there is a small damped oscillation after the slope ends.
The height for the first outflow 'step' is 0.20 and increases to 0.40 for the first inflow reflection.
The 'plateau' between steps is approx. horizontal.

- bottom left: The depth spread is 4 times higher than for top left, as expected.
The wave slope and height is approx. the same as above.
But the 'plateau' phase is no longer horizontal but with increasing inclination

- top right: $F = 0.04 \rightarrow F = 0.40$: The Froude waves are dissipated after approx 4 peaks.
The wave height is reduced, as with the sysinf-model,
and the 'plateau' phase inclination increases more quickly than for $F = 0.040$.

- bottom right: The dissipation is even stronger, the Froude waves disappear after the second peak.
This will help to reduce the overshoot of the closed loop response.

6.5.4 Case [A]

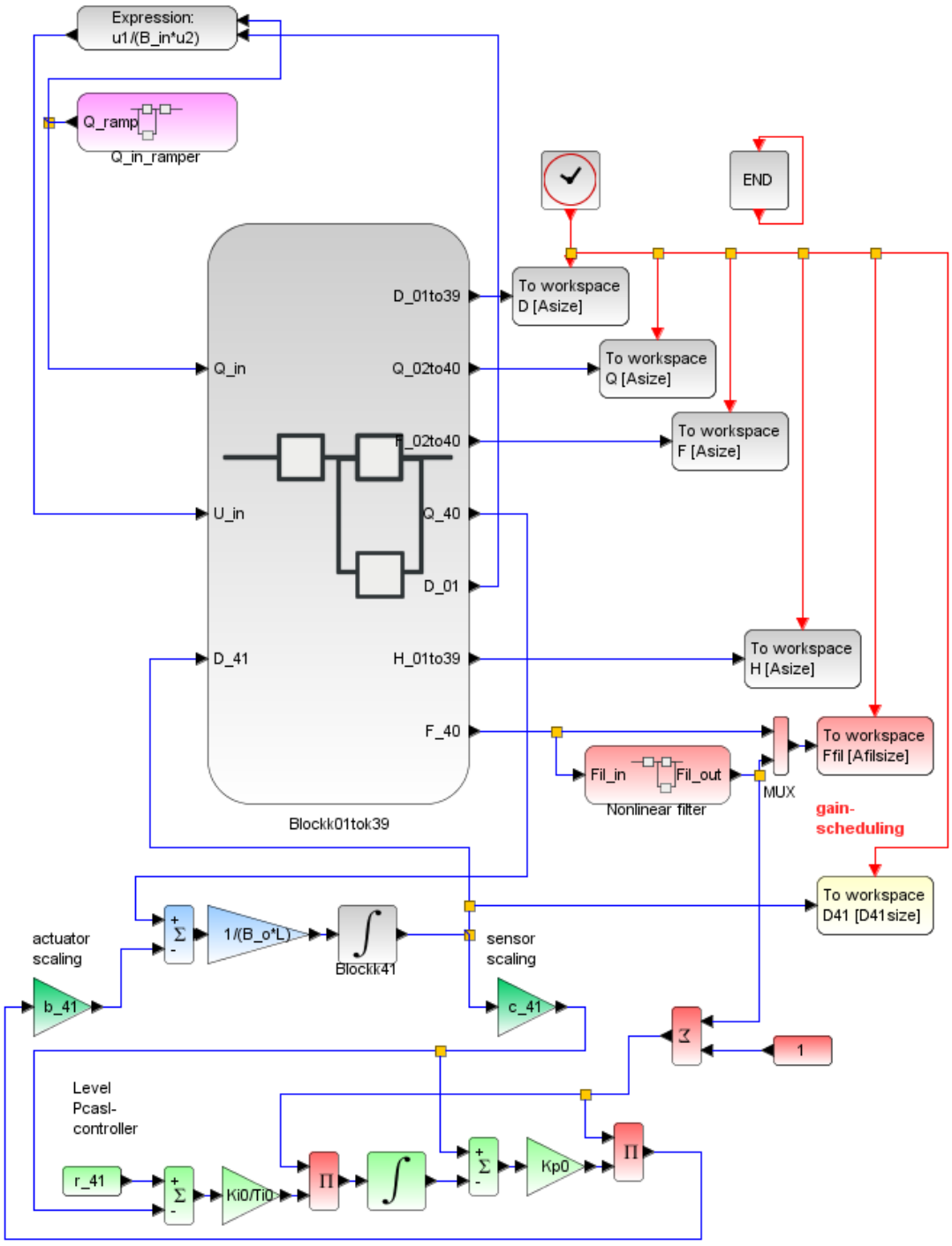


Figure 6.34: Case [A]: top level diagram for closed loop step responses to inflow z_in

```
// s_c6_03_1_context
// Glf  2017_06_15

N= 20;      // no Volume and Momentum-segments
g = 10.; L_tot = 500.; L = L_tot/(N+1);
kap = 0.50;  // "centered"

// operating point
   Q_d = 50.0; D_d = 2.50;
   B_d = 10.0; S_d = 1*(-1.0);
   U_d = 2.0;  U_F_d = 5.0; F_d = 0.40;

// GMS-friction coefficient
   //  k_s = 30.0;
       k_s = 60.0;
   //  k_s = 100.0;

// reference bottom slope
  R_d = (B_d*D_d)/(B_d + 2*D_d);
  I_d = (U_d/(k_s*(R_d)^(2/3)))^2;

//working spans for integrators
 D_min = +0.001; D_max = 4*D_d;
 Q_min = +0.001; Q_max = 4*Q_d;

// channel geometry
// Basic layout: constant width
vB = B_d*ones(1,(2*N+1));

//  basic layout: bottom 'horizontal'
vS0 = (S_d)*ones(1,(2*N+1));

vd0 = ones(1,(2*N+1)); vD0 = D_d*vd0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_d*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// friction slope of the bottom vS
vi0    = ones(1,(2*N+1));
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));

for kk=2:2:(2*N),
  vI0(kk)    = I_d*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end
// shift bottom zero to outflow point
    Sf_o = vSf(kk+1);
for k5= 1:1:(2*N+1),
    vS(k5)  = vS(k5)  - Sf_o*vi0(k5);
end

//  outflow data
B_o=vB(kk+1);S_o=vS(kk+1);D_o=vD0(kk+1);Q_o=vQ0(kk+1);

// inflow generation
  dQ_sm = 0.04; dQ_la = 0.90;
t_st_1 =   100.0; r_1_0 =Q_d; r_1_1=(1.0+dQ_sm)*Q_d;
t_st_2 =  1400.0; r_2_0 = 0.; r_2_1=-(dQ_la+dQ_sm)*Q_d;
t_st_3 =  7000.0; r_3_0 = 0.; r_3_1=    + dQ_sm*Q_d;
t_st_4 =  9000.0; r_4_0 = 0.; r_4_1=+(dQ_la-dQ_sm)*Q_d;
t_st_5 = 13000.0; r_5_0 = 0.; r_5_1=    + dQ_sm*Q_d;
T_fin =  16000.0;
// inflow slew rate
g_st   = 10.0; u_up_st = +1.00; u_dn_st = -1.00;
tau_st = 72.0; Q_st_0  = Q_d;

// sensor & actuator scaling factors
c_41 = 1.00 / D_d;  b_41 = Q_d / 1.00;

// nonlinear filtering of F_40
b = 0.040; bhi = +0.380; blo = 0.0; bhys = 0.005;
T_fil = 300.0; F_f_0 = F_d ;
g_paral = 0.5;

// controller gain settings
T_E_d = 2.0*(L_tot / U_F_d);
r_41 = 1.0; // <- D_41=2.50m, H_41=1.50m
  Kp0 = 1.0*(1/F_d); // impedance matching

  Ki0 = 1.0;     Ti0 = T_E_d;
// initial cond. xi0 for I-action
Kp1 = Kp0*(1+F_d); xi0 = (Kp1 - 1.0)/Kp1;

// Data transfer to Plots
CC = 21;   CN = 3200;   delT = T_fin/CN;
Asize = 1.01*CC*CN;   Afilsize=1.01*3*CN;
C41 = 2;  D41size = 1.01*C41*CN;
```

```
// s_c6_03_1_crunplot
//  Glf  2017_06_15

stacksize('max'); exec('s_c6_03_1_context.sce', -1);
importXcosDiagram('s_c6_03_1.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:10,  clf(kfig); end
vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

t_0 = t_st_1 - 100.0; t_e0 = t_st_1+1200.0;
f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[t_0,48.,t_e0,57.]);
xtitle(['case [A]: Q_2 to Q_40; F = 0.400; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[t_0,2.46,t_e0,2.66]);
plot2d(D41.time,D41.values,style=1,rect=[t_0,2.40,t_e0,2.60]);
xtitle(['case [A]: D_1 to D_41; F = 0.400; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[t_0,1.4,t_e0,2.8]);
xtitle(['case [A]: H_1 to H_41; F = 0.400; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[t_0,0.38,t_e0,0.45]);
xtitle(['case [A]: F_2 to F_40; F = 0.400; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

t_6=t_st_3-100.0; t_e6=t_st_3+1200.0;
f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[t_6,4.0,t_e6,12.0]);
xtitle(['case [A]: Q_2 to Q_40;  F = 0.040; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[t_6,1.3,t_e6,2.7]);
plot2d(D41.time,D41.values,style=1,rect=[t_6,2.0,t_e6,2.6]);
xtitle(['case [A]: D_1 to D_41;  F = 0.040; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[t_6,1.3,t_e6,1.7]);
xtitle(['case [A]: H_1 to H_41;  F = 0.040; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[t_6,0.02,t_e6,0.22]);
xtitle(['case [A]: F_2 to F_40,  F = 0.040; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f9 = scf(9);
plot2d(Q.time,Q.values,vcolor,rect=[0,0,T_fin,60]);
xtitle(['case [A]: Q_2 to Q_40; ; k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);

f10 = scf(10);
plot2d(Ffil.time,Ffil.values,style=[5,2],...
       rect=[0,0.,T_fin,0.50]);
xtitle(['case [A]: F_40 (red), Ffil (blue); k_s  = ',...
 msprintf('%4.0f', k_s)]); xgrid(1);
```

The nonlinear filter in Fig.6.34 is to decouple the dynamics of the gain scheduling loop from the dynamics of the level control loop.
In most practical cases large shifts of the operating point (by Q) are very slow compared to the small 'steps' in Q for the level control loop. Here a much shorter time span shall be used: The operating point on Q is set to move from its design value Q_d down to (extrapolated) zero in $3'600s$, to be compared to the echo travelling time at zero flow of $200s$.
The decoupling of the scheduling parameter $Ffil(t)$ from the actual $F(t)$ is designed as follows: The basis is a first order tracking loop, Fig.6.35. A dead band of span $\pm b$ is inserted around its equilibrium by the two relay blocks to suppress changes of the 'scheduling F' around its equilibrium. And a linear path is added with gain g_paral to drift the filter output slowly to the dead span center. The filter time constant $T\_fil = 300\ s$ is adjusted such that the output Fil_out will stay near the previous equilibrium during the small transient dynamic but then will follow fast enough to the next equilibrium.
This nonlinear filter has been necessary for case [C], where the closed loop response would be not acceptable without it. It has been also helpful for case [B], but is really not needed for case[A].

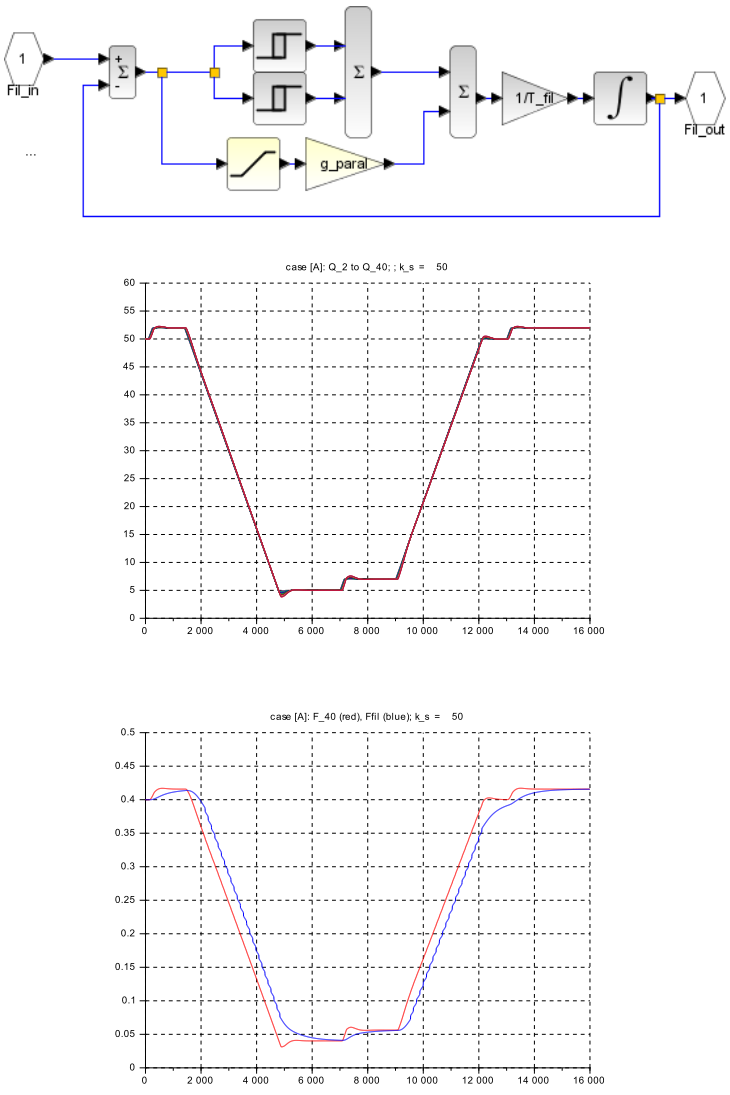


Figure 6.35: Case [A]: closed loop responses of Q and F to large inflow shifts and small 'steps' z_in,
(top) diagram for nonlinear filtering of F_40
(center) Q for full time range
(bottom) for nonlinearly filtered F

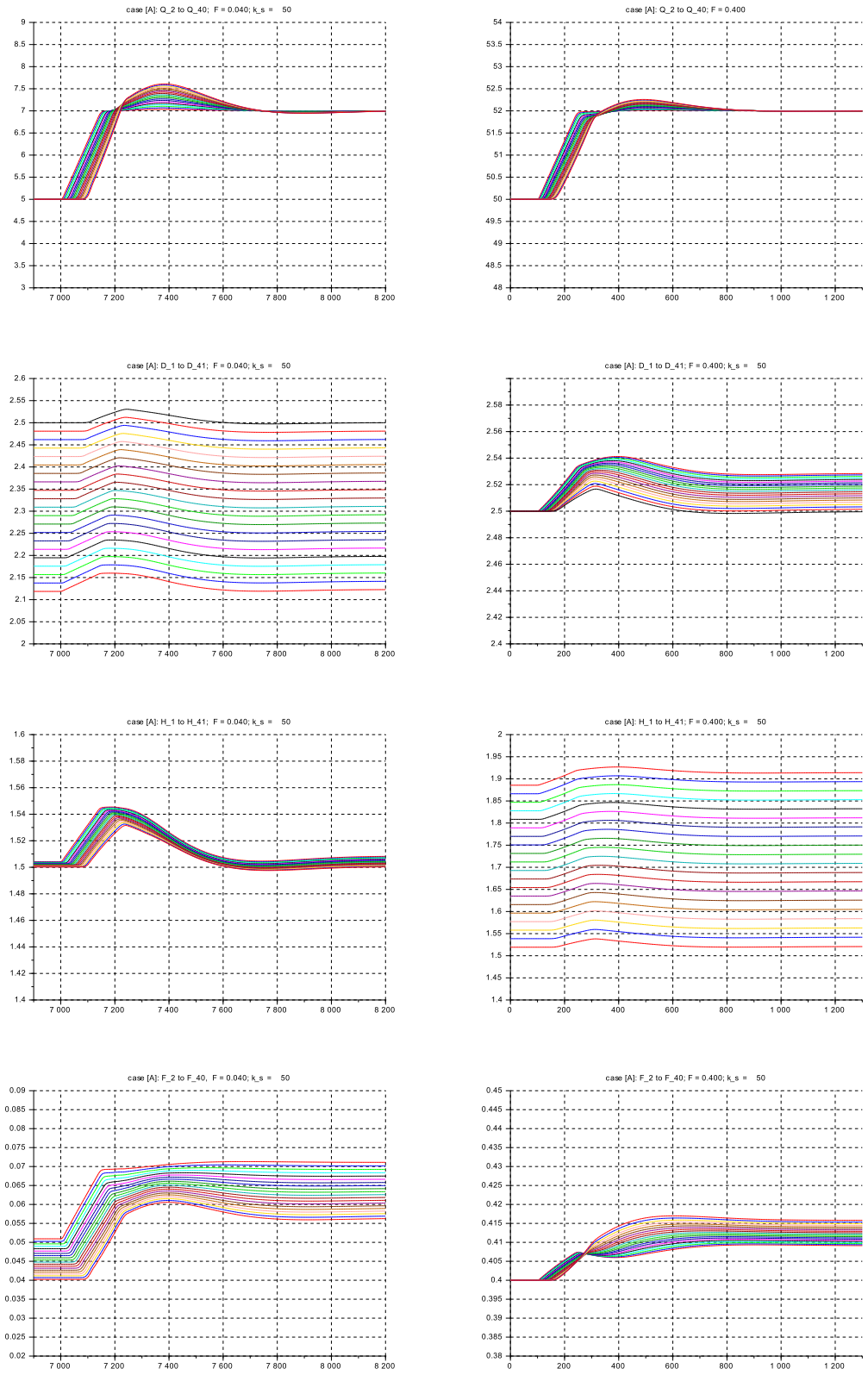


Figure 6.36: Case [A]: zoomed-in responses of Q, D, H, F to inflow 'steps' z_in, (left) $F = 0.040$, (right) $F = 0.400$

Discussion:

- *Fig.6.35:*
  The nonlinear filter for gain scheduling F performs approximately as expected. It will need further refinement before it is ready for implementation on a real plant...

- *Fig.6.36:*
  At $F = 0.040$ the overshoot on Q is $\approx +0.30$ at $\approx 400s$ and the response is decayed at $\approx 800s$. The corresponding values for the step response from section 6.4 are $\approx +0.30$, $\approx 300s$, $\approx 700s$. Considering that here the 'step' is in fact a ramp, taking $100s$ to reach the new steady state, this agrees well.

- For the inflow level the overshoot peak is $\approx 0.04m$, that is ≈ 0.016 of D_d. This is for $\Delta Q = 0.4$ of the local steady state $Q = 5m^3/s$. This yields a relative size of 0.40, which agrees well with the value for Δx_in from sections 6.3 and 6.4.

- At $F = 0.40$ the effect of the friction is more pronounced on the overshoots. The relative overshoot in Q is now 0.10, compared to 0.25 from section 6.4. And the relative overshoot on the inflow level is ≈ 0.20, compared to 0.30 from sect. 6.4.

- The decay time however is not affected markedly by the friction. It is $\approx 700s$, which again agrees well with the corresponding values from section 6.4.

- Finally note that at $F = 0.040$ the observed variation $\Delta F = 0.0155$ agrees well with the expected one of 0.016 for the relative flow increase of 0.40 there. And also at $F = 0.40$ the observed variation of $\Delta F = 0.016$ for the relative flow increase of 0.04.

To summarize, for case[A]: The responses for the nonlinear time domain model agree well with the results from the linearized model section 6.4 and thus from the pure delay model section 6.3. The only marked difference is the steady state level and depth spread along the channel length, due to the friction.

6.5.5 Case [B]

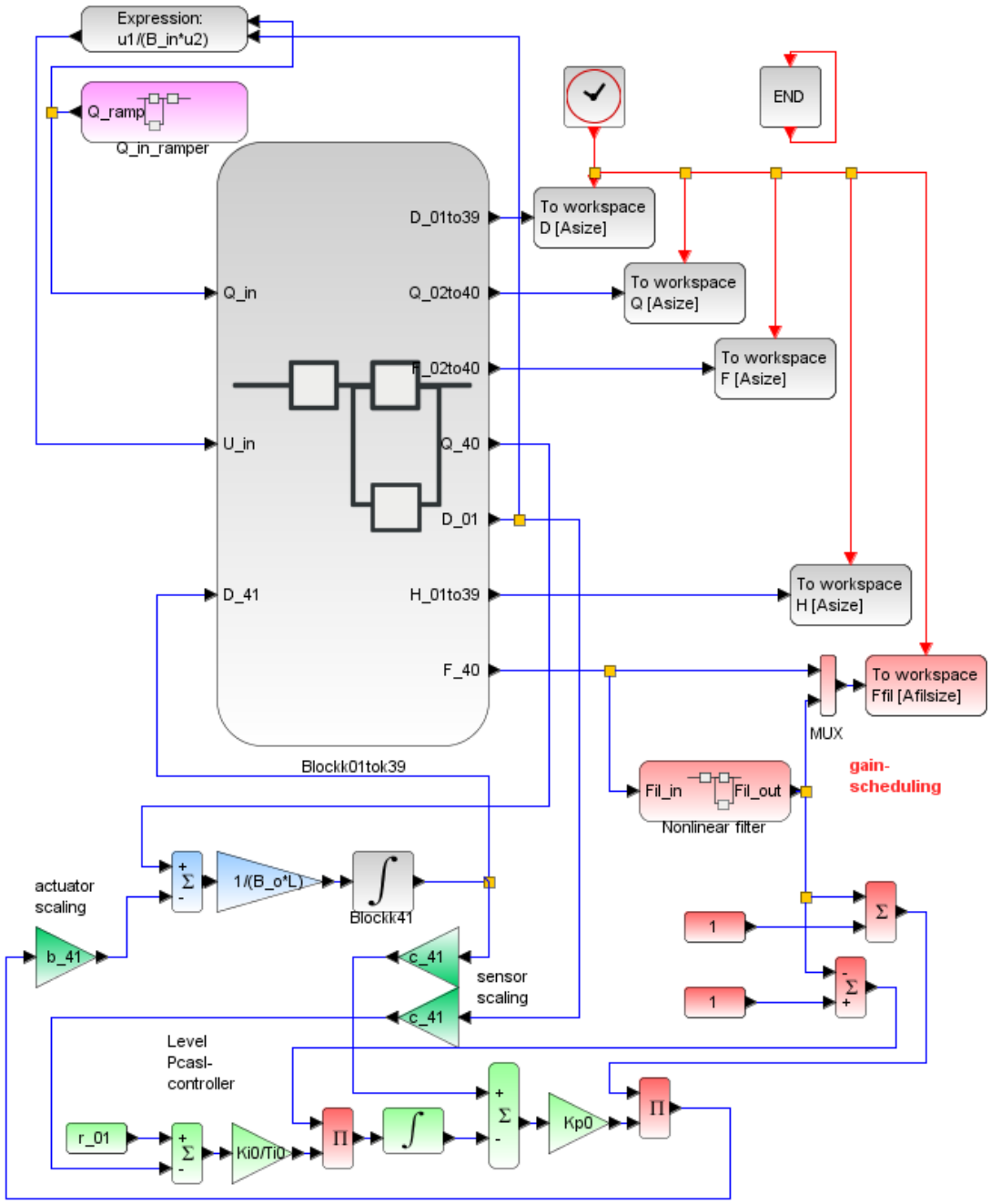


Figure 6.37: Case [B]: top-level diagram for closed loop step responses to inflow z_in

```
// s_c6_03_2_context
// Glf  2017_06.15
N= 20;      // no Volume and Momentum-segments
g = 10.; L_tot = 500.; L = L_tot/(N+1);
kap = 0.5;  // "centered"

// operating point
   Q_d = 50.0; D_d = 2.50;
   B_d = 10.0; S_d = 1*(-1.0);
   U_d = 2.0;  U_F_d = 5.0; F_d = 0.40;

// GMS-friction coefficient
   k_s = 50.0;

// reference bottom slope
  R_d = (B_d*D_d)/(B_d + 2*D_d);
  I_d = (U_d/(k_s*(R_d)^(2/3)))^2;

//working spans for integrators
 D_min = +0.001; D_max = 4*D_d;
 Q_min = +0.001; Q_max = 4*Q_d;

// channel geometry
// Basic layout: constant width
vB = B_d*ones(1,2*N+1));

//  basic layout: bottom 'horizontal'
vS0 = (S_d)*ones(1,(2*N+1));

vd0 = ones(1,(2*N+1)); vD0 = D_d*vd0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_d*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// friction slope of the bottom vS
vi0    = ones(1,(2*N+1));
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));

for kk=2:2:(2*N),
  vI0(kk)    = I_d*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end
// shift bottom zero to outflow point
    Sf_o = vSf(kk+1);
for k5= 1:1:(2*N+1),
    vS(k5)  = vS(k5)  - Sf_o*vi0(k5);
end

//  outflow data
B_o=vB(kk+1);  S_o=vS(kk+1);
D_o=vD0(kk+1); Q_o=vQ0(kk+1);

// inflow generation
  dQ_sm = 0.04; dQ_la = 0.90;
t_st_1 =   100.0; r_1_0 =Q_d; r_1_1=(1.0+dQ_sm)*Q_d;
t_st_2 =  2600.0; r_2_0 = 0.; r_2_1=-(dQ_la+dQ_sm)*Q_d;
t_st_3 =  8200.0; r_3_0 = 0.; r_3_1=    + dQ_sm*Q_d;
t_st_4 = 10200.0; r_4_0 = 0.; r_4_1=+(dQ_la-dQ_sm)*Q_d;
t_st_5 = 15000.0; r_5_0 = 0.; r_5_1=    + dQ_sm*Q_d;
T_fin =  18000.0;
// inflow slew rate
g_st   = 10.0; u_up_st = +1.00; u_dn_st = -1.00;
tau_st = 60.0; Q_st_0  = Q_d;

// outflow by outlevel PcI-control
// ********************************
// sensor & actuator scaling factors
c_41 = 1.00 / D_d;  b_41 = Q_d / 1.00;

// nonlinear filtering of F_02
T_sw = t_st_1; rsw0 = -1.0; rsw1 = +1.0;
b = 0.040; bhi = +0.380; blo = 0.0; bhys = 0.005;
T_fil = 300.0; F_f_0 = F_d ;
g_paral = 0.5;

// controller gain settings
T_E_d = 2.0*(L_tot / U_F_d);
r_01 = 1.0 ;
  Kp0 = 1.0*(1/F_d); // impedance matching

  Ki0 = 1.0;   Ti0 = T_E_d;

  // initial cond. xi0 for I-action
Kp1 = Kp0*(1+F_d); xi0 = (Kp1 - 1.0)/Kp1;

// Data transfer to Plots
CC = 21;   CN = 3600;   delT = T_fin/CN;
Asize = 1.01*CC*CN;   Afilsize=1.01*3*CN;
```

```
// s_c6_03_2_crunplot
//  Glf  2017_06_15
stacksize('max'); exec('s_c6_03_2_context.sce', -1);
importXcosDiagram('s_c6_03_2.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:10,  clf(kfig); end
vcolor = [ 1, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

t_0 = t_st_1 - 100.0; t_e0 = t_st_2-100.0;
f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[t_0,48.,t_e0,54.]);
xtitle(["case [B]: Q_2 to Q_40; F = 0.400;k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[t_0,2.4,t_e0,2.6]);
xtitle(["case [B]: D_1 to D_41; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[t_0,1.0,t_e0,2.0]);
xtitle(["case [B]: H_1 to H_41; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[t_0,0.35,t_e0,0.45]);
xtitle(["case [B]: F_2 to F_40; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

t_6=t_st_3-100.0; t_e6=t_st_3+1200.0;
f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[t_6,3.0,t_e6,12.]);
xtitle(["case [B]: Q_2 to Q_40;  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[t_6,2.4,t_e6,2.9]);
xtitle(["case [B]: D_1 to D_41;  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[t_6,1.7,t_e6,1.9]);
xtitle(["case [B]: H_1 to H_41; F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[t_6,0.01,t_e6,0.14]);
xtitle(["case [B]: F_2 to F_40,  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f9 = scf(9);
plot2d(Q.time,Q.values,vcolor,rect=[0,0,T_fin,60]);
xtitle(["case [B]: Q_2 to Q_40;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f10 = scf(10);
plot2d(Ffil.time,Ffil.values,style=[5,2],...
       rect=[0,0.,T_fin,0.55]);
xtitle(["case [B]: F_40 (red), Ffil (blue);  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);
```

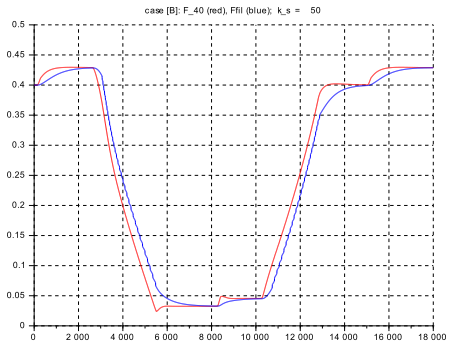


Figure 6.38: Case [B]: closed loop responses to inflow 'steps' z_in, for F_40, $Ffil$

Discussion:

- *Fig.6.38:*
  The nonlinear filter for gain scheduling F performs approximately as expected.

- *Fig.6.39:*
  At $F = 0.040$ the overshoot on Q is $\approx +0.40$ at $\approx 250s$ and the response is decayed at $\approx 700s$. The corresponding values for the step response from section 6.4 are $\approx +0.40$, $\approx 200s$, $\approx 600s$. Considering that here the 'step' is in fact a ramp, taking $100s$ to reach the new steady state, this agrees well.

- For the inflow level the overshoot peak is $\approx 0.04m$, that is ≈ 0.016 of D_d. This is for $\Delta Q = 0.4$ of the local steady state $Q = 5m^3/s$. This yields a relative size of 0.40, which agrees well with the value for Δx_in from sections 6.3 and 6.4.

- There is a small undershoot $\Delta Q \approx 0.05$, which is due to the stronger integral action $Ki0 = 1.40$ instead of 1.0 in section 6.4. This re-tuning has been done to reduce the settling time for$F = 0.40$, see below.

- At $F = 0.40$ the effect of the friction is more pronounced on the overshoot size. The relative overshoot in Q is now 0.25 at $300s$, compared to 0.20 from section 6.4. And the relative overshoot on the inflow level is ≈ 0.40, compared to 0.30 from sect. 6.4.

- The decay time however is affected strongly by the friction. It is $\approx 1100s$, with $750s$ for the corresponding values from section 6.4.

- Finally note that at $F = 0.040$ the observed variation $\Delta F = 0.016$ agrees well with the expected one of 0.016 for the relative flow increase of 0.40 there. And also at $F = 0.40$ the observed variation of $\Delta F = 0.0165$ for the relative flow increase of 0.04.

To summarize, for case [B]: the responses for the nonlinear time domain model agree acceptably with the results from the linearized model section 6.4 and thus from the pure delay model section 6.3. Note the (small) re-tuning on the integral action ($Ki \rightarrow 1.40 \cdot (1 + F)$) to reduce the even larger settling time for $F = 0.40$.

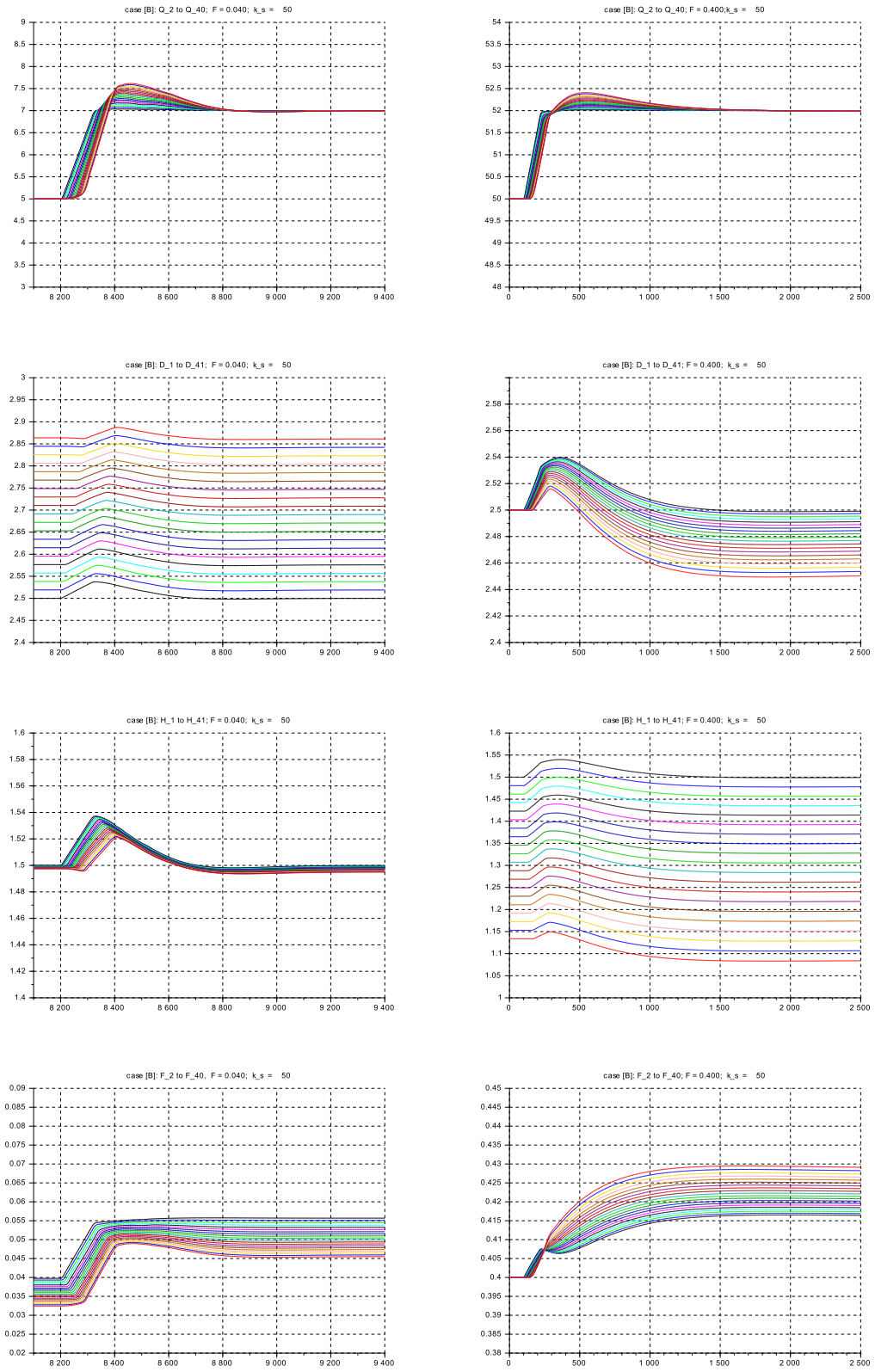


Figure 6.39: Case [B]: zoomed-in responses of Q, D, H, F to inflow 'steps' z_in, (left) $F = 0.040$, (right) $F = 0.400$

6.5.6 Case [C]

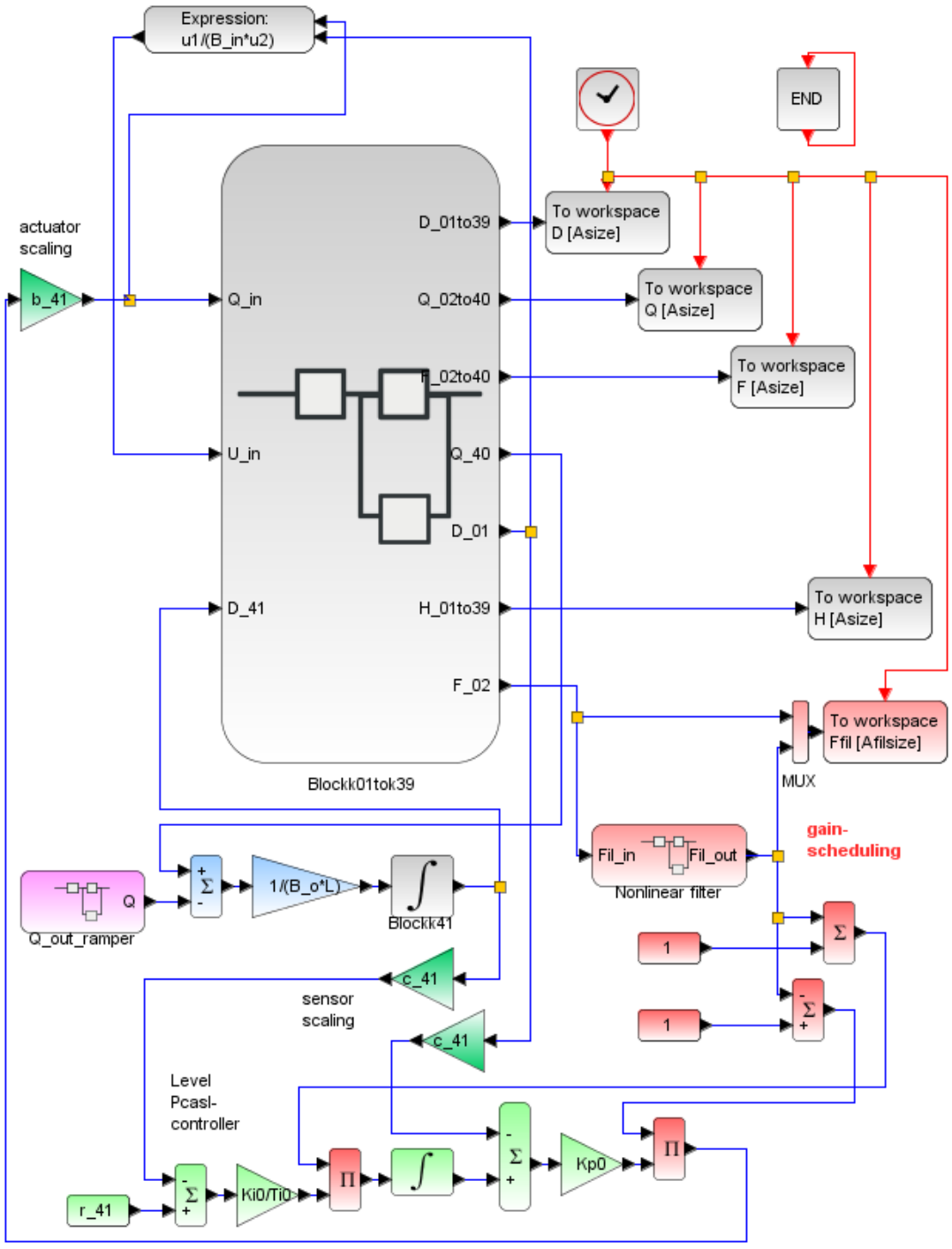


Figure 6.40: Case [C]: top level diagram for closed loop responses to outflow 'steps' *z_out*

```
// s_c6_03_3_context
// Glf  2017_06_15

N= 20;      // no Volume and Momentum-segments
g = 10.; L_tot = 500.; L = L_tot/(N+1);
kap = 0.50;  // "centered"

// operating point
   Q_d = 50.0; D_d = 2.50;
   B_d = 10.0; S_d = 1*(-1.0);
   U_d = 2.0;  U_F_d = 5.0; F_d = 0.40;

// GMS-friction coefficient
// k_s = 50.0;
   k_s = 100.0;

// reference bottom slope
  R_d = (B_d*D_d)/(B_d + 2*D_d);
  I_d = (U_d/(k_s*(R_d)^(2/3)))^2;

// working spans for integrators
 D_min = +0.001; D_max = 4*D_d;
 Q_min = +0.001; Q_max = 4*Q_d;

// channel geometry
//*****************
// Basic layout: constant width
vB = B_d*ones(1,(2*N+1));

//  basic layout: bottom 'horizontal'
vS0 = (S_d)*ones(1,(2*N+1));
//*************************

vd0 = ones(1,(2*N+1)); vD0 = D_d*vd0;
vq0 = ones(1,(2*N+1)); vQ0 = Q_d*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// friction slope of bottom vS
vi0    = ones(1,(2*N+1));
vdelSf = zeros(1,(2*N+1));
vSf    = zeros(1,(2*N+1));
for kk=2:2:(2*N),
  vI0(kk)    = I_d*vi0(kk);
  vdelSf(kk) = - L*vI0(kk);
  vSf(kk)    = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1)  = vSf(kk)   + 0.5*vdelSf(kk);
end
```

```
// s_c6_03_3_crunplot
//  Glf  2017_06_15

stacksize('max'); exec('s_c6_03_3_context.sce', -1);
importXcosDiagram('s_c6_03_3.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************
for kfig = 1:1:10,  clf(kfig); end
vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 1];

t_0 = t_st_1 - 100.0; t_e0 = t_st_1+1200.0;
f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[t_0,48.,t_e0,54.]);
xtitle(["case [C]: Q_2 to Q_40; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[t_0,2.4,t_e0,2.6]);
xtitle(["case [C]: D_1 to D_41; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[t_0,1.0,t_e0,2.0]);
xtitle(["case [C]: H_1 to H_41; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[t_0,0.38,t_e0,0.45]);
xtitle(["case [C]: F_2 to F_40; F = 0.400;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

t_6=t_st_3-100.0; t_e6=t_st_3+1200.0;
```

```
for k4= 1:1:(2*N+1),
    vS(k4)  = vS0(k4)  + vSf(k4);
end
//shift bottom zero to outflow point
    Sf_o = vSf(kk+1);
for k5= 1:1:(2*N+1),
    vS(k5)  = vS(k5)  - Sf_o*vi0(k5);
end

//  outflow data
B_o=vB(kk+1);S_o=vS(kk+1);D_o=vD0(kk+1);Q_o=vQ0(kk+1);

// inflow generation
  dQ_sm = 0.04; dQ_la = 0.90;
t_st_1 =  2000.0; r_1_0 =Q_d; r_1_1=(1.0+dQ_sm)*Q_d;
t_st_2 =  3400.0; r_2_0 = 0.; r_2_1=-(dQ_la+dQ_sm)*Q_d;
t_st_3 =  9000.0; r_3_0 = 0.; r_3_1=    + dQ_sm*Q_d;
t_st_4 = 11000.0; r_4_0 = 0.; r_4_1=+(dQ_la-dQ_sm)*Q_d;
t_st_5 = 15800.0; r_5_0 = 0.; r_5_1=    + dQ_sm*Q_d;
T_fin =  18000.0;
// inflow slew rate
g_st   = 10.0; u_up_st = +1.00; u_dn_st = -1.00;
tau_st = 60.0; Q_st_0  = Q_d;

// outlevel PcI-control by inflow
// ******************************
// sensor & actuator scaling factors
c_41 = 1.00 / D_d;  b_41 = Q_d / 1.00;

// nonlinear filtering of F_02
T_sw = t_st_1, rsw0 = -1.0; rsw1 = +1.0;
b = 0.050; bhi = +0.380; blo = 0.0; bhys = 0.005;
T_fil = 300.0; F_f_0 = F_d ;
g_paral = 0.5;

// controller gain settings
   T_E_d = 2.0*(L_tot/U_F_d);
   r_H_o = (D_d + S_o)/D_d;

   Kp0 = 1.0*(1/F_d); // impedance matching

   Ki0 = 1.00;   Ti0 = T_E_d;

// initial condition xi0 for I-action
Kp1 = Kp0*(1-F_d);   xi0 = (Kp1 + (1.0+(S_o/D_d)))/Kp1;

// Data transfer to Plots
CC = 21;   CN = 3600;   delT = T_fin/CN;
Asize = 1.01*CC*CN;   Afilsize=1.01*3*CN;
```

```
f5 = scf(5);
plot2d(Q.time,Q.values,vcolor,rect=[t_6,3.0,t_e6,12.0]);
xtitle(["case [C]: Q_2 to Q_40;  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f6 = scf(6);
plot2d(D.time,D.values,vcolor,rect=[t_6,1.8,t_e6,2.9]);
xtitle(["case [C]: D_1 to D_41;  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f7 = scf(7);
plot2d(H.time,H.values,vcolor,rect=[t_6,1.3,t_e6,1.9]);
xtitle(["case [C]: H_1 to H_41;  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f8 = scf(8);
plot2d(F.time,F.values,vcolor,rect=[t_6,0.02,t_e6,0.14]);
xtitle(["case [C]: F_2 to F_40,  F = 0.040;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f9 = scf(9);
plot2d(Q.time,Q.values,vcolor,rect=[t_0,0,T_fin,60]);
xtitle(["case [C]: Q_2 to Q_40;  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);

f10 = scf(10);
plot2d(Ffil.time,Ffil.values,style={5,2],...
       rect=[t_0,0.,T_fin,0.5]);
xtitle(["case [C]: F_02 (red), Ffil (blue);  k_s  = ",...
 msprintf('%4.0f', k_s)]); xgrid(1);
```

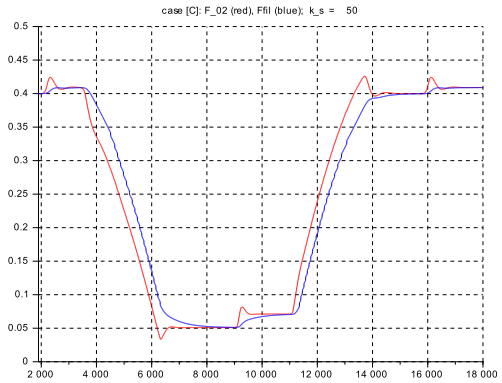


Figure 6.41: Case [C]: closed loop responses of F_02, $Ffil$ to outflow 'steps' z_out

Discussion:

- *Fig.6.41:*
  Again the nonlinear filter for gain scheduling F performs approximately as expected (to be refined before any real world application...).

- *Fig.6.42:*
  At $F = 0.040$ the overshoot on Q is $\approx +0.35$ at $\approx 300s$ and the response is decayed at $\approx 700s$. The corresponding values for the step response from section 6.4 are $\approx +0.45$, $\approx 100s$, $\approx 400s$. Considering that here the 'step' is in fact a ramp, taking $100s$ to reach the new steady state, this is still about acceptable.

- For the outflow level the overshoot peak is $\approx -0.04m$, that is ≈ -0.016 of D_d. This is for $\Delta Q = 0.4$ of the local steady state $Q = 5m^3/s$. This yields a relative size of -0.40, which agrees well with the value for Δx_out from sections 6.3 and 6.4. The settling time is $\approx 700s$, in contrast to $\approx 400s$ from section 6.4.

- At $F = 0.40$ the effect of the friction is more pronounced on the overshoot size. The relative overshoot in Q is now 0.60 at $500s$, compared to 0.70 at $\approx 220s$ from section 6.4. And the relative overshoot on the outflow level is ≈ -0.85 at $\approx 250s$, compared to ≈ -0.65 at $\approx 80s$ from sect. 6.4.

- The settling time however is affected strongly by the friction. It is $\approx 1500s$ (!), in contrast to $\approx 500s$ from section 6.4.

- There is a weakly damped response with the first relative undershoot $\Delta Q \approx -0.15$, resulting in a period of $\approx 1'800s$. Note that the integral action has been de-tuned already here to $Ki = 0.70 \cdot (1 + F)$.

- Finally note that at $F = 0.040$ the observed variation $\Delta F = 0.016$ agrees well with the expected one of 0.016 for the relative flow increase of 0.40 there. And also at $F = 0.40$ the observed variation of $\Delta F = 0.0165$ for the relative flow increase of 0.04.

- Further note the strong oscillation on F. This makes the nonlinear filtering of F necessary. Otherwise an even stronger oscillation results due to the additional feedback through the gain scheduling loop (simulation results not shown here).

To summarize, for case [C]: the responses for the nonlinear time domain model agree well with the results from the linearized model section 6.4 and thus from the pure delay model section 6.3., but only for $F = 0.040$. For $F = 0.40$ the weakly damped oscillation is marginally acceptable. A possible cause may be the larger water level inclination which leads to a higher acceleration and thus to a higher kinetic energy of the channel content, which has to be countered by the control action [10].

[10] But this shall remain an open point here...

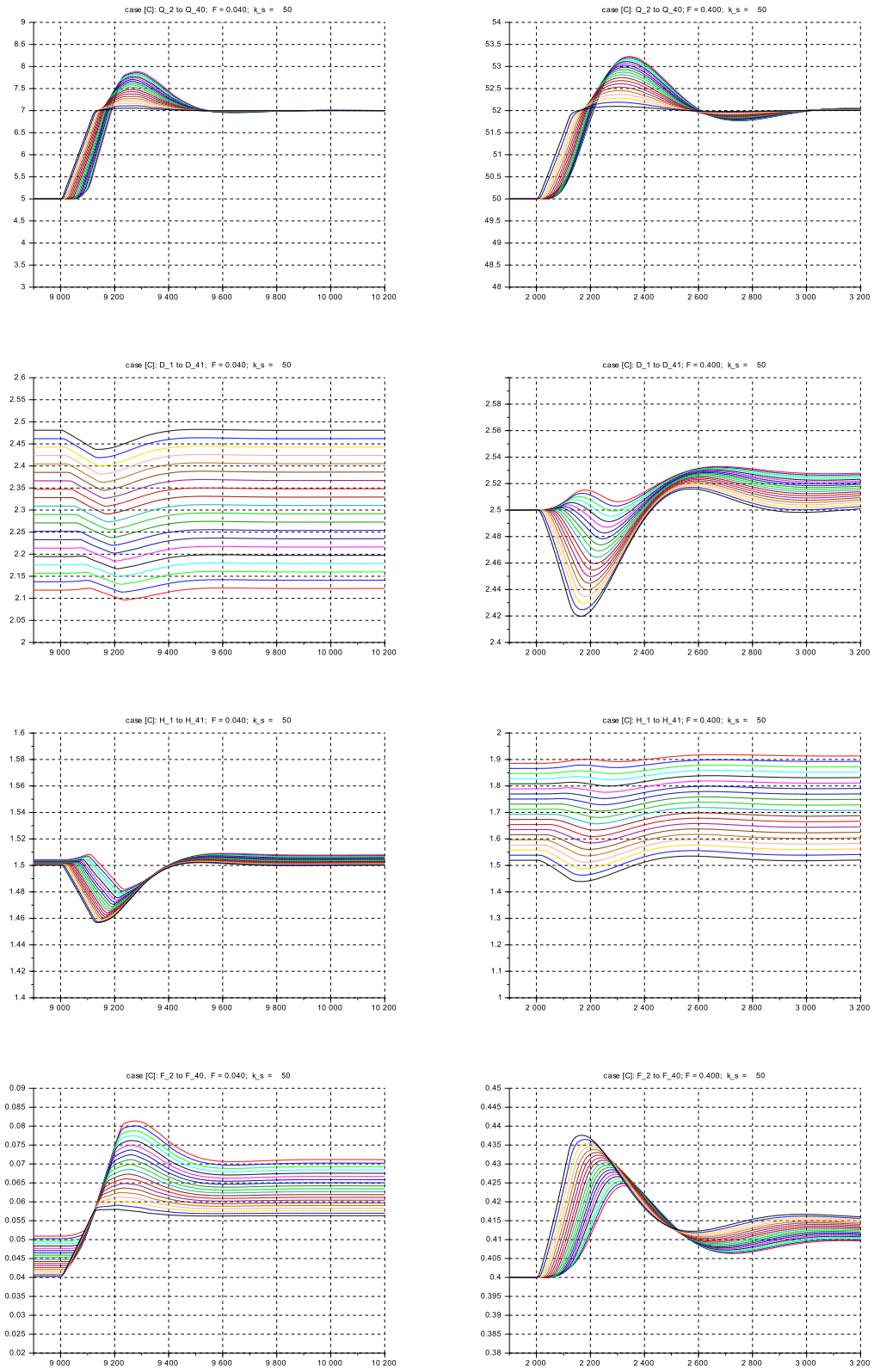


Figure 6.42: Case [C]: zoomed-in responses of Q, D, H, F to outflow 'steps' z_out, (left) $F = 0.040$, (right) $F = 0.400$

Chapter 7

Application Studies

The next step is to verify the control design method on two typical run-of-river plants, one rather large (“ Birsfelden”), and a small one (“NeueWelt”). The open loop plant model for ‘Birsfelden’ has been discussed in section 3.9.

7.1 Control Design Verification for case study “Birsfelden”

(WGS 84 location: 47.559308 7.626752)

7.1.1 Modelling

The upstream basin model

The plant model is taken from sect. 3.9 in its nonlinear time-domain form with all coefficients taken over from there. Note the variable depth along the river reach, at $Q\_d := 1'000m^3/s$ from $D\_in \approx 3m$ to $D\_out = D\_d := 10.60m$, while the river width is $B\_d := 157m$ is approx. constant. Therefore the Froude number will change considerably along the reach.

Design of the experiment sequence

The transients for the simulations are modelled as follows:

- The simulation is started at the original level profile before the power station was built. And the outflow level controller is switched ON. Then the simulation runs until the steady state depth profile is reached at $t > 9'000s$. This transient is suppressed in the following plots which start at $t = 9'800s$.
- At $t = 10'000s$ a step $dQ = +100m^3/s$ is applied on the inflow to verify the response at design/nominal flow.
- At $t = 20'000s$ after the transient has decayed sufficiently a ramp reduction down to $Q = 500m^3/s$ is applied on the inflow, taking $2'700s$, that is $-0.0926m^3/s$ per sec.
  Note the *Remark* below.
- After the new equilibrium has been reached at $t = 38'000s$ a step $dQ = +100m^3/s$ is applied to verify the response at low flow.
- At $t = 48'000s$ a rising flow ramp is applied up to $Q = 1'500m^3/s$ with $+0.0926m^3/s$ per sec.
- After equilibration a final step $dQ = +100m^3/s$ is applied to verify the response at high flow.

- The total river flow exceeds this value several times per year, up to around $3'000m^3/s$. Then the weirs parallel to the power station must be lowered. This would require an additional total flow controller implementing a strategy of splitting flow to the different actuators. This is not considered further here.

Remark:
Analysis from the available data[1] on the river Rhine flow at Birsfelden shows that the maximum rate of change is approx. $+1600m^3/s$ per day that is approx. $+0.0185m^3/s$ per sec. This is a factor of 5 slower than the value used in the simulation. Also note that the down rate is even smaller than this value. So the value for the simulation is on the safe side...

Level controller design

The value of the design parameter F_d needed for the controller gains is taken at the outflow end where the control loop is situated.

$$F_d = \frac{U_d}{U_{F_d}}|_o = \frac{Q_d}{B_d \cdot D_d}\frac{1}{\sqrt{g \cdot D_d}} := \frac{1'000}{157 \cdot 10.6} \cdot \frac{1}{\sqrt{106}} = 0.05836$$

And the echo travelling time at near zero flow conditions is from sect. 3.9 $T\_e\_0 := 1'000s$.

The level controller structure is applied from sect. 6.5 without change. Both input and output scalings need to be applied,
$c\_L\_o = 1.0/D_d$ and $b\_L\_o = Q_d/1.0$.
Then the design value for the depth reference is $refD\_d := 1.0$. Finally the initial condition for the integral action is from sect. 6.5, case A: $xi\_0 = (kp\_L - 1.0)/kp\_L$.

[1]Swiss Federal office for environment, Hydrology division,
https://www.bafu.admin.ch/bafu/de/home/themen/wasser/...
...publikationen-studien/publikationen-wasser/hydrologisches-jahrbuch-schweiz-2015.html

7.1.2 Implementation

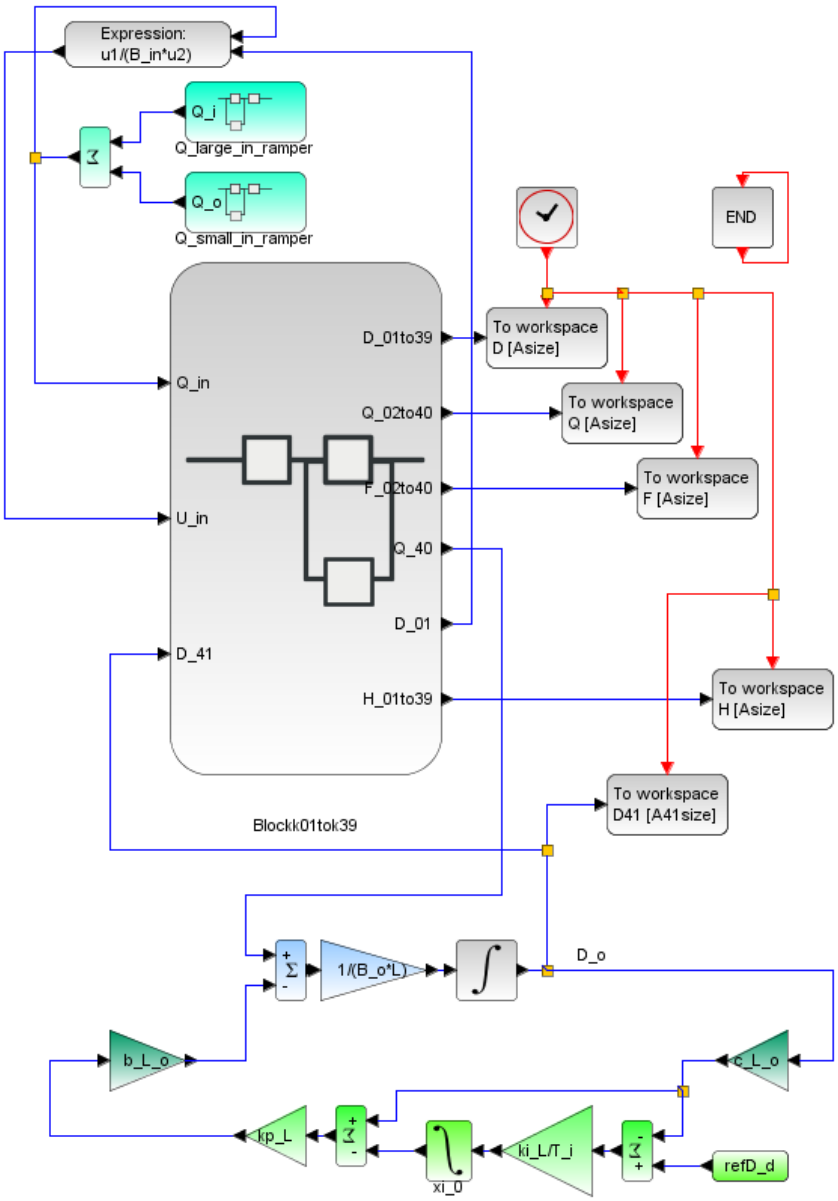


Figure 7.1: Case study "Birsfelden": top level diagram for closed loop step responses to inflow *z_in*

```
// s_c7_01_10_context
// Glf  2017_06_15
// Birsfelden 1000m^3/s

g = 10.;
// reference op-point
Q_r=1000.0; H_r=2.2; S_r=0.80*(-1.0);
D_r=H_r - S_r; B_r = 157.;
U_r = 2.02; F_r = 0.368;  L = 375.;
kap = 1.0;   N= 20;

// select design/initial op-point
 Q_0 = 1.00*Q_r;
 H_0 = H_r; S_0 = S_r; B_0 = B_r;
 U_0 = Q_r/(B_r*D_r); D_0 = D_r;
 D_min = +0.0001; D_max = 40*D_r;
 Q_min = +0.0001; Q_max = 40*Q_r;
```

```
// channel geometry
// Basic element: constant (nominal) width
vb =[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
     1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
     1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
     1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
     1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vB = B_0*vb;

// basic layout: horizontal bottom
vs=[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
    1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
    1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
    1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,...
    1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0,1.0];
vS0 = S_0*vs;

vd0=ones(1,(2*N+1)); vD0=D_0*vd0; vH0=vD0 + vS0;
```

```
vq0 = ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1);S_in=vS0(1);D_in=vD0(1);Q_in=vQ0(1);

// GMS-coefficient
   k_s = 32.5;
// friction slope of the bottom vS and of the surface vH
vdelSf = zeros(1,(2*N+1));  vSf = zeros(1,(2*N+1));
vDr = D_r*vd0;              vQr = Q_r*vq0;
vdelHf = zeros(1,(2*N+1));  vHf = zeros(1,(2*N+1));
vH00   = ones(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
vdelSf(kk) = - L*vIr(kk); vdelHf(kk) = - L*vIH(kk);
  vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
  vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
 vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
 vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);    end;
for k4= 1:1:(2*N+1),
vS(k4)= vS0(k4)+ vSf(k4); vH(k4)= vH0(k4)+ vHf(k4);
vH0(k4) = vH(k4) ; vH0(k4) = 2.20*vH00(k4) ;
vD0(k4) = vH0(k4) - vS(k4);  end;

//  outflow data
B_o=vB(kk+1); S_o=vS(kk+1); D_o=vD0(kk+1);
H_o=vH0(kk+1); Q_o = vQ0(kk+1);
```

```
dQ = 0.10; // small step
// inflow LARGE sequence
ti_st_1= 10000.0;ri_1_0=Q_0;  ri_1_1=Q_0;
ti_st_2= 20000.0;ri_2_0=0.;ri_2_1=-(0.50+dQ)*Q_0;
ti_st_3= 48000.0;ri_3_0=0.;ri_3_1=+(1.00-dQ)*Q_0;
// inflow slew rate
gi_st = 1.0; ui_up_st = +1.0; ui_dn_st = -1.0;
taui_st = 10.8; Qi_st_0 = Q_0;

// inflow SMALL sequence
dQ = 0.10;
t_st_1= 10000.0;r_1_0 = 0.;  r_1_1=(0.0+dQ)*Q_0;
t_st_2= 38000.0;r_2_0 = 0.;  r_2_1 = +dQ*Q_0;
t_st_3= 70000.0;r_3_0 = 0.;  r_3_1 = +dQ*Q_0;
// inflow slew rate
g_st = 10.0; u_up_st = +1.0; u_dn_st = -1.0;
tau_st = 1.0; Q_st_0 = 0.;

T_fin = 80000.0;

// Outflow Level control
c_L_o = 1.0/10.60;  b_L_o = 1000.0/1.0;
T_e_0 = 2000.0;  F_d = 0.0583;
kp_L = 1.0*(1.0/F_d);
ki_L = 1.0; T_i = T_e_0;
refD_d = 1.0; xi_0 = (kp_L - 1.0)/kp_L;

// Data transfer to Plots
CC=21;  CN=8000; delT=T_fin/CN;
Asize=1.01*CC*CN; A41size=1.01*2*CN;
```

```
// s_c7_01_10_crunplot
//  Glf  2017_06_15
//  Birsfelden 1000 m^3/s

stacksize('max');exec('s_c7_01_10_context.sce', -1);
importXcosDiagram('s_c7_01_10.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

for kfig = 1:1:9,  clf(kfig); end
vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

T_0=0.0;T_1=9500.;T_2=20000;T_3=37500;
T_4=48000.;T_5=69500.;   vc = [1];

f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[T_1,400,T_fin,1700]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[T_1,2.0,T_fin,12.0]);
plot2d(D41.time,D41.values,vc,rect=[T_1,2.0,T_fin,12.0]);
xtitle("D_1 to D_41"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[T_1,+1.8,T_fin,4.2]);
xtitle("H_1 to H_39"); xgrid(1);
```

```
f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[T_1,0.0,T_fin,0.4]);
xtitle("F_2 to F_40"); xgrid(1);

//*****************************************************
eX = L*(1:1:N); eXe = 21*L;
for k6= 1:1:N,  vSr(k6) = vS(2*k6-1); end

f5 = scf(5); yH = H.values; oH = yH(999,:);
plot2d(eX',[oH',vSr],style={5,1},rect=[0.,-9.0,eXe,+4.0]);
xtitle("Longitudinal H (red) at 10000 s, bottom (black)");
xgrid(1);

v = [5]; T_f = T_fin;
f6 = scf(6);
plot2d(D41.time,D41.values,v,rect=[T_1,10.46,T_f,10.74]);
xtitle("D_41, zoom-in"); xgrid(1);

f7 = scf(7);
plot2d(D41.time,D41.values,v,rect=[T_1,10.56,T_2,10.66]);
xtitle("D_41, zoom-in, at Q=1000 m3/s"); xgrid(1);

f8 = scf(8);
plot2d(D41.time,D41.values,v,rect=[T_3,10.56,T_4,10.66]);
xtitle("D_41, zoom-in, at Q= 500 m3/s"); xgrid(1);

f9 = scf(9);
plot2d(D41.time,D41.values,v,rect=[T_5,10.56,T_f,10.66]);
xtitle("D_41, zoom-in, at Q=1500 m3/s"); xgrid(1);
```

Discussion:

- Note that the Froude number F_40 relevant for controller tuning stays close to zero for the whole flow range. Thus no gain scheduling is needed here.
- The overall closed loop performance is as expected, the responses on D_41 are well damped. No further manual tuning was needed. Note that this eliminates a very substantial effort and time-on-site during commissioning (due to the long settling time of $\approx 7'200\ s$).
- The tracking error on D_41 along the (much faster than real) flow ramps is $\pm 12\ cm$ and still conforms to the proscribed limits given in sect.3.9.
- And the peak error on D_41 for the (large and fast) flow steps $dQ = +100\ m^3/s$ is 4.5 cm, which is also conforming.

7.1.3 Simulation results

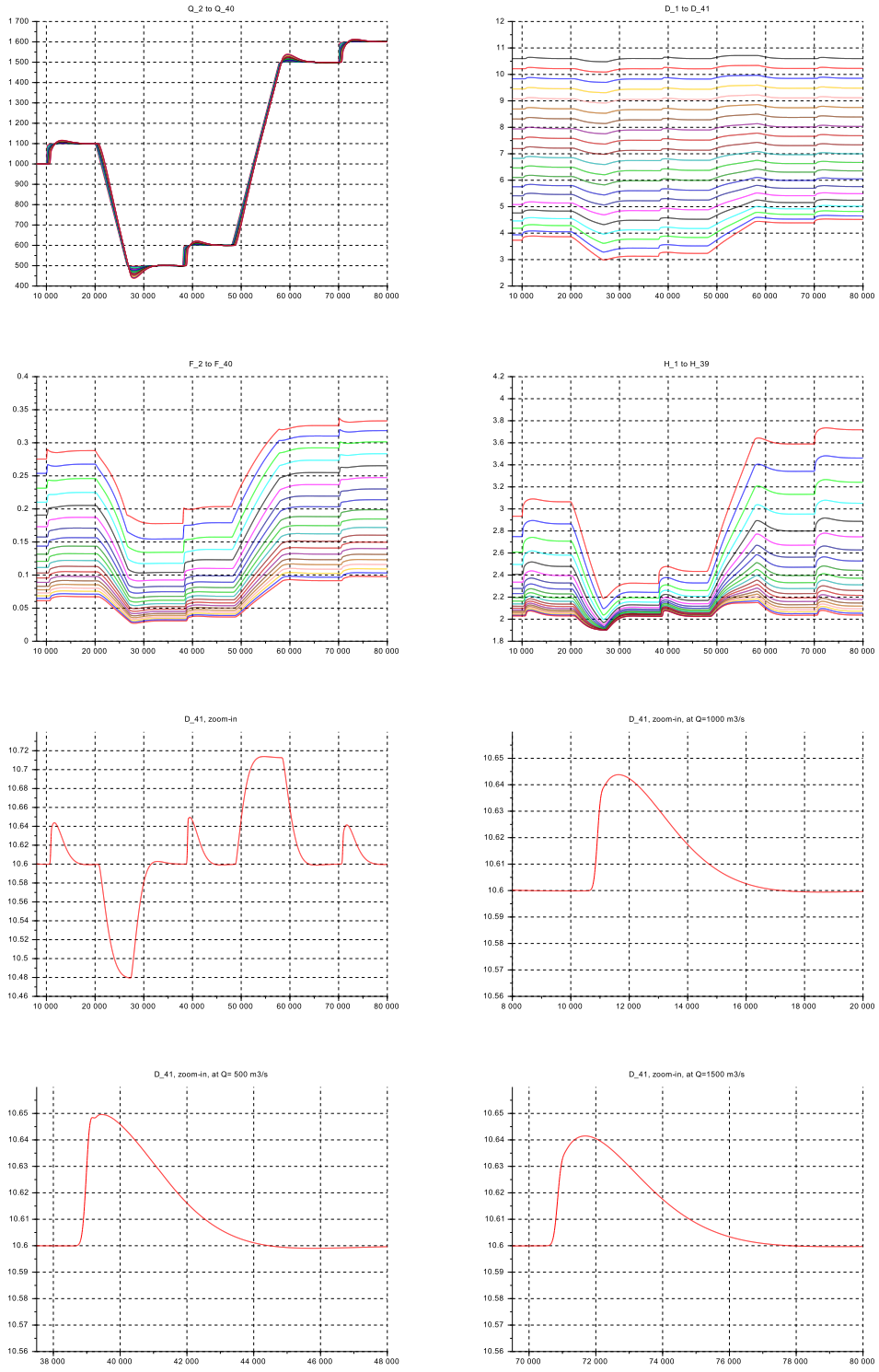


Figure 7.2: top group of four: overview of closed loop step responses to inflow z_in
lower group of four: zoom-in on closed loop step responses of D to inflow z_in
(upper left) for full time sequence (upper right) at $Q = 1000m^3/s$,
(lower left) at $Q = 500m^3/s$, (lower right) at $Q = 1500m^3/s$

On this particular plant there are two locks for river barges. The filling takes approx. 500 sec at a flow of approx. 70 m^3/s. To show the effect on the depth control deviation the previous diagram with its .sce files is suitably modified (not shown here), but the simulation results are given in Fig.7.3.

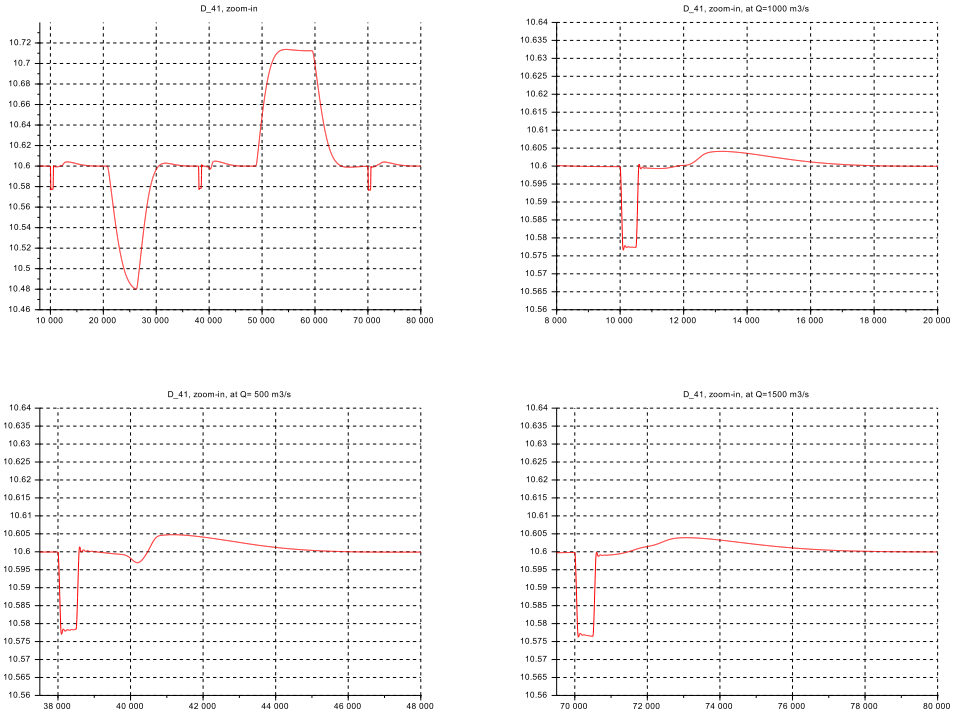


Figure 7.3: zoom-in on closed loop step responses of D to outflow z_out due to lock filling
(upper left) for full time sequence (upper right) at $Q = 1000m^3/s$,
(lower left) at $Q = 500m^3/s$, (lower right) at $Q = 1500m^3/s$

Discussion:
The max. deviation on D_41 is $-2.2\ cm$ and of rectangular shape. And the settling time is again $\approx 7'200\ s$ with an overshoot of approx. $+0.5\ cm$ and no undershoot.

7.2 Control Design for case study "Neue Welt"

(WGS 84 location: 47.525863 7.621635)

This application case is about a much smaller hydro power station 'NeueWelt' on the river Birs near Basel. The reference flow is $Q\_r = 20m^3/s$. The upstream river reach has two distinct parts. From the inflow end there is a rather steep natural river bed of approx. 300 m length and with approx. constant width 8 m. The second part has an approx. flat bottom and its width increases approx. linearly from 8 m to approx. 50 m. There is a wide overfall at the outflow end in parallel to one turbine-generator unit, as well as some fixed small flows to a park nearby and a fish stairs. And the region of transition between river flow and basin flow is not fixed in space but will move downstream with increasing river flow Q.
The control layout here is the 'industry standard' (layout case A).
As to the mode of operation, the turbine flow must be maximized up to a specified residual overfall flow by way of a level control loop. Below a given threshold in river flow, the turbine flow is to be switched off. And if the river flow exceeds this threshold again the turbine is to be started and loaded appropriately by the level control loop.
The hydrological model is taken from chap. 3 in the nonlinear time-domain form. From its particular shape the Froude number will change considerably along the reach. Its design value F_d needed for the controller gain design is taken at the outflow end where the level sensor and controller are situated.

7.2.1 Data Assembly and Modelling

The Basin Geometry

This is approximated from the actual geometry (from Swiss topo maps) as follows

Top view
From the inflow to $240m := 6 \cdot L = 6 \cdot 40m$ downstream the width is set constant to $B = B_r = 8.0m$. From there on down to the end of the basin at the overfall at $800m$, the width increases linearly to $50m$ that is from $1.0 \cdot B_r$ up to $6.25 \cdot B_r$.

Lateral view
Starting at the inflow, the bottom shall start at $+2.0m$ above the (arbitrary) zero reference elevation. The bottom shape is then generated by the GMS-friction by setting the Strickler coefficient $k_s := 25.0$ (as for large pebbles and medium stones [2]). At reference flow $Q_r = 20m^3/s$ and $D_r = 1.25m$ and thus $U_r = 2.0m/s$ in the natural river part, this yields a slope value $I_r \approx 64 \cdot 10^{-4}$. Then the bottom would go down to the zero reference elevation at $320m \approx 8 \cdot L$ from the inflow.
However as the width starts to increase before, the bottom will round out and it will finish at the downstream end (at $\approx -0.05m$) below the zero horizon level, see the longitudinal profile plot.

Flow Distribution

Fixed outflow: $Q = 2.8m^3/s$
Residual Overfall flow (as by Specification): $Q_{min} = 5\%$ of $20m^3/s = 1.0m^3/s$
Turbine flow: $Q_T = 6.0 \ldots 18.0m^3/s$

For a total inflow of $Q = 20m^3/s$ and the fixed and minimum overfall outflow as given above, this leaves for the turbine flow $Q_T = 16.2m^3/s$ that is $1.8m^3/s$ less than the full turbine flow. Thus there is a sufficient maneuvering margin for the level controller.

Overfall flow calculation

The crown of the overfall is considered flat and sufficiently long that the flow may attain the critical velocity $U_{F=1.0}$. For the overfall flow Q_of, with H_of as the height of the overfall crown above zero

[2] from photos taken on site

reference elevation, and with H_41 as the level just upstream of the overfall:

$$\begin{aligned} \Delta H &:= H\_41 - H\_of \\ U &:= U_{F=1.0} = \sqrt{g \cdot \frac{2}{3}\Delta H} \\ Q\_of &= U_{F=1.0} \cdot B\_of \cdot \frac{2}{3}\Delta H = B\_of \cdot \sqrt{g} \cdot \left(\frac{2}{3}\Delta H\right)^{3/2} \\ \rightarrow \quad \Delta H &= \frac{3}{2} \cdot \left(\frac{Q\_of}{B\_of \cdot \sqrt{g}}\right)^{2/3} \end{aligned}$$

Using $B\_of := 40m$ (leaving $10m$ for the turbine and fixed outflow inlets);

$$\begin{aligned} \text{for } \; Q = 1.0m^3/s \quad &\rightarrow \quad \Delta H \approx 0.060m \\ \text{for } \; Q = 20.0 - 2.8m^3/s \quad &\rightarrow \quad \Delta H \approx 0.37m \end{aligned}$$

Note that for the latter case the crown is too short and the upstream flow to the overfall is far from being at rest. So this will be a very approximate value to the real one. But this is of no great importance, as it supposes the level control loop to be switched out, where it normally is switched in.
Finally the crown level has to be at $+1.14m$ above the zero reference elevation in order to obtain the required minimum overfall flow with the reference of the level-flow controller set at $refH_o := +1.20m$.

Design of the Experiment Sequence

The inflow to the system is modelled as follows, see also Fig.7.7:

- It starts at its design/nominal value $Q\_in = Q\_d = 20\ m^3/s$ with the turbine at shutdown, that is level control OFF, up to $t = 2'000\ s$. The reference value of the depth/level controller tracks to its actual value, see lower diagram of Fig.7.5.
- At $t = 2'500\ s$ the level control is enabled/switched ON. It starts the turbine and loads it transiently up to its upper constraint $Q\_T\_HI = 18\ m^3/s$ and stabilises at $Q\_T = 16.2\ m^3/s$. Note that the integrator time constant in the reference tracker module is set such that the transient outflow to the downstream river bed does not rise beyond approx. 22 m^3/s. Without this additional module the downstream outflow would peak dangerously higher (at nearly 30 m^3/s, which is not shown here).
- At $t = 5'500\ s$ the inflow is reduced in a ramp with $3'600\ s$ for Q_d that is $-\ 5.56\ 10^{-3}\ m^3/s$ per sec.
  The test a is for the endpoint at $Q\_in = 4.0\ m^3/s$ which will result in an automatic turbine shutdown.
  And test b will stop at $Q\_in = 10.0\ m^3/s$ such that the turbine will not be stopped and the level control loop continues operating at its low end, see Fig.7.6.
- After equilibrium has been attained, at $t = 11'200\ s$ the inflow ramps up with $6'000\ s$ for Q_d that is $+3.33 \cdot 10^{-3}\ m^3/s$ per sec. .
- After full design flow has been attained at $t = 14'000\ s$ the final transient to equilibrium is shown. Total elapsed time is 5 hours.

Note that the ramp down and ramp up of inflow is estimated to be a factor of 5...10 higher than actual values. Thus the steady state tracking error of the level controller in the simulation will be quite larger than for the real installation.

Level Controller Design

The level controller is active during turbine operation only and has to maintain a target overfall flow $Q\_of = 1m^3/s$ by manipulating turbine flow Q_T.

The operating range of Q_T is constrained from 6.0 m^3/s to 18 m^3/s. This implies that the turbine must be stopped for low river flow and must be re-started automatically if the river flow increases again, see Fig.7.5. Due to the high lower constraint value some switching bumps will be unavoidable, even if anti-windup and reference tracking schemes are added to the controller structure, see Fig.7.4. Note that these add-on's require the cascaded control structure from above to be transformed into the equivalent parallel structure. Also note that all details of turbine startup and shutdown shall be modelled/masked by a first order ramp filter with ramp time from 0 to 6 m^3/s of 90 s.

The level controller design procedure from chap.6 is used.

Thus the main parameter for the controller gains is the Froude number at design conditions Q_d, D_o and F_d.

$$F_d = \frac{U_d}{U_{F_d}}|_o = \frac{Q_d}{B_o \cdot D_o} \frac{1}{\sqrt{g \cdot D_o}} := \frac{20.0}{50.0 \cdot 1.25} \cdot \frac{1}{\sqrt{12.5}} = 0.09051$$

. Thus no gain scheduling is required, as $\approx 0.03 \leq F \leq \approx 0.09$.

And the echo travelling time at near zero flow conditions is
for $L := 500\ m$ and $U\_F = 3.535\ m/s$ $\quad T\_e\_0 := 286s$.

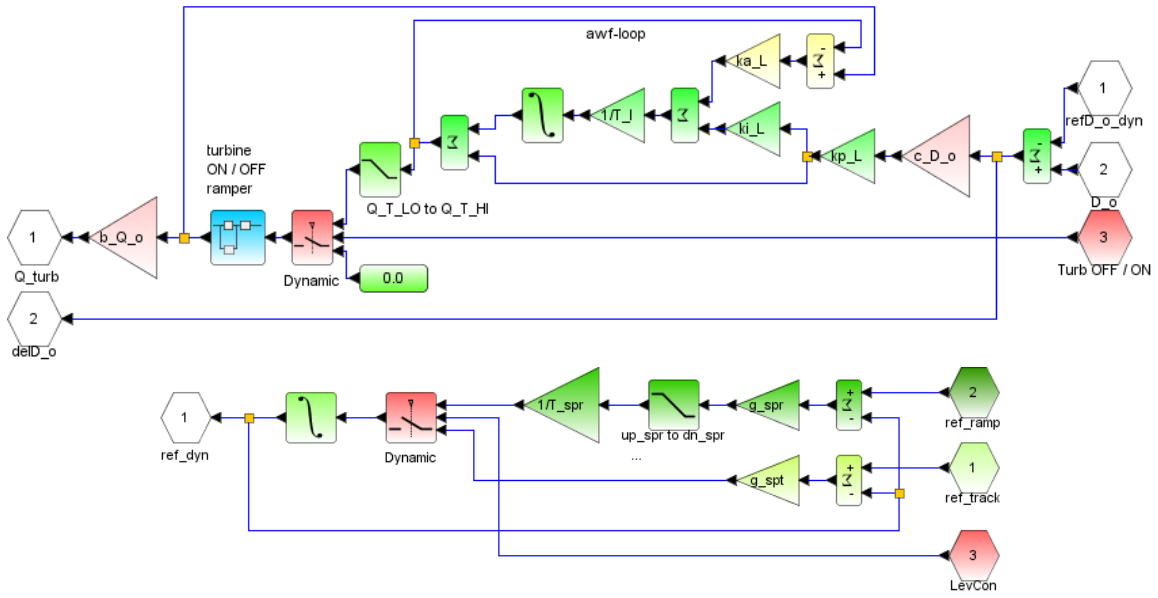


Figure 7.4: Case study "NeueWelt":
(top) diagram for the Level Controller with anti-windup feedback and turbine on/off ramper
(bottom) diagram for reference tracker and ramper

7.2.2 Implementation

```
// s_c7_02_context
// Glf   2017_06_21

g = 10.;
// reference op-point
Q_r = 20.0; D_r = 1.25; S_r = (-2.0)*(-1.0);
H_r = D_r + S_r; B_r = 8.; U_r = 2.00; L = 40.;
kap = 1.0;  N= 20;

// select actual op-point
 Q_0 = 1.00*Q_r; // design riverflow 20 m^3/s;
 H_0=H_r; S_0=S_r; B_0=B_r; U_0=U_r; D_0=1.0*D_r;
 D_min = +0.0001; D_max = 40*D_r;
 Q_min = +0.0001; Q_max = 40*Q_r;

// channel width
vb =[1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,...
     1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,...
     1.1875, 1.3750, 1.5625, 1.7500, 1.9375, 2.1250,...
     2.3125, 2.5000, 2.6875, 2.8750, 3.0625, 3.2500,...
     3.4375, 3.6250, 3.8125, 4.0000, 4.1875, 4.3750,...
     4.5625, 4.7500, 4.9375, 5.1250, 5.3125, 5.5000,...
     5.6875, 5.8750, 6.0625, 6.2500, 6.25];
vB = B_0*vb;

// channel bottom initialization
vs = ones(1:1:(2*N+1));
vS0 = S_0*vs;
vd0=ones(1,(2*N+1)); vD0=D_0*vd0; vH0=vD0 + vS0;
vq0=ones(1,(2*N+1)); vQ0 = Q_0*vq0;

// Inflow data
B_in=vB(1); S_in=vS0(1); D_in=vD0(1); Q_in=vQ0(1);

// GMS-coefficient
  k_s = 25.0;

// friction slope of bottom vS and surface vH
vdelSf = zeros(1,(2*N+1)); vSf = zeros(1,(2*N+1));
vDr = D_r*vd0; vQr = Q_r*vq0;vdelHf = zeros(1,(2*N+1));
vHf = zeros(1,(2*N+1));  vH00 = ones(1,(2*N+1));
for kk=2:2:(2*N),
vRtilda(kk)=((vB(kk)*vDr(kk))/(vB(kk)+2*vDr(kk)))^(2/3);
vIr(kk) = (vQr(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
vIH(kk) = (vQ0(kk)/(vB(kk)*vDr(kk)*k_s*vRtilda(kk)))^2;
vdelSf(kk) = - L*vIr(kk); vdelHf(kk) = - L*vIH(kk);
vSf(kk)   = vSf(kk-1) + 0.5*vdelSf(kk);
vSf(kk+1) = vSf(kk)   + 0.5*vdelSf(kk);
vHf(kk)   = vHf(kk-1) + 0.5*vdelHf(kk);
vHf(kk+1) = vHf(kk)   + 0.5*vdelHf(kk);
end;
for k4= 1:1:(2*N+1),
vS(k4) = vS0(k4)+ vSf(k4);vH(k4) = vH0(k4)+ vHf(k4);
vH0(k4) = vH(k4); vD0(k4) = D_0*vd0(k4);
vH0(k4) = vD0(k4) + vS(k4);
end;

// outflow data
B_o=vB(kk+1);S_o=vS(kk+1);D_o=vD0(kk+1);H_o=vH0(kk+1);
Q_o=vQ0(kk+1);

// inflow generation
dQ = 1.0; t_st_1 = 5500.0; t_st_2 = 11000.0;
// test a :
//     r_1_0 = Q_0; r_1_1=(1.0-0.80*dQ)*Q_0;
//     r_2_0 = 0.0; r_2_1 = +0.80*dQ*Q_0;
// // test b :
   r_1_1=(1.0-0.50*dQ)*Q_0;
   r_2_1 = +0.50*dQ*Q_0;

// inflow slew rate
g_st = 20.0; u_up_st = +0.6; u_dn_st = -1.0;
tau_st = 180.0; Q_st_0 = Q_0;

T_fin  = 18000.0;

// overfall outflow
B_of = 40.0;  refH_of = 1.14;  g_of = B_of*(g^(1/2));

// fixed outflow
Q_o_fixed = 2.80;

// Outflow Level control PI-awf
c_D_o = 1.0/D_0;  b_Q_o = Q_0/1.0;
F_d = (Q_0/(B_o*D_0))/((g*D_0)^(1/2));

Q_T_LO = 0.29; Q_T_HI = 0.90;
  kp_L=1.0*(1/F_d); ki_L=1.0; T_I=283.; ka_L=kp_L/2.0;
refD_o = 1.2566;

// level control ENABLE
log_OFF=-1.0; log_ON=+1.0; Q_of_off=0.50; Q_of_on=6.0;
T_LevConENABLE = 2700.; ulevcon0=-1; ulevcon1=+1;

// turbine ON/OFF ramp
T_fil=270.0; g_fil=30.0; slewup=1.0; slewdn=-1.0;

// setpoint tracking and ramper
up_spr=0.1;dn_spr=-0.1; g_spr=10.;g_spt=1.; T_spr=360;

// gain for zoom-in on depth deviation
g_delD = 100.0; // 1.0 is 1 cm

// Data transfer to Plots
CC=26; CN=900; delT=T_fin/CN;
Asize=1.01*CC*CN; FLsize=1.01*(6+1)*CN;
```

```
// s_c7_02_crunplot
//  Glf  2017_06_21

stacksize('max');exec('s_c7_02_context.sce', -1);
importXcosDiagram('s_c7_02.zcos');
typeof(scs_m); scs_m.props.context;
Info=list(); Info=scicos_simulate(scs_m,Info);
//*********************************

 for kfig = 1:1:10,  clf(kfig); end
vcolor = [ 5, 2, 3, 4, 1, 6, 9,11,13,15,...
          17,19,21,22,25,27,29,32, 2, 5];

T_0 = 0.; T_1 = 2000.; T_2 = T_fin;
f1 = scf(1);
plot2d(Q.time,Q.values,vcolor,rect=[T_1,0.,T_2,30.]);
xtitle("Q_2 to Q_40"); xgrid(1);

f2 = scf(2);
plot2d(D.time,D.values,vcolor,rect=[T_1,.0,T_2,2.0]);
xtitle("D_1 to D_39"); xgrid(1);

f3 = scf(3);
plot2d(H.time,H.values,vcolor,rect=[T_1,+1.0,T_2,4.0]);
xtitle("H_1 to H_39"); xgrid(1);

f4 = scf(4);
plot2d(F.time,F.values,vcolor,rect=[T_1,0.0,T_2,1.0]);
xtitle("F_2 to F_40"); xgrid(1);

f5 = scf(5);
vcolor2=[5,14,2,22,32,1,27];
plot2d(FL.time,FL.values,vcolor2,rect=[T_1,-2,T_2,32]);
xtitle("controlled Water Depth and Flows"); xgrid(1);
h1=legend(['100*(D_o - refD_o)','100*delD_o',...
'Q_turbine','Q_overfall','Q_fixed','Q_in','Q_downriver']);

T_3 = 7000.; T_4 = 9200;
f6 = scf(6);
plot2d(Q.time,Q.values,vcolor,rect=[T_3,3.0,T_4,10.0]);
xtitle("Q_2 to Q_40, Turbine OFF, zoom-in"); xgrid(1);

f7 = scf(7);
plot2d(D.time,D.values,vcolor,rect=[T_3,0.4,T_4,1.4]);
```

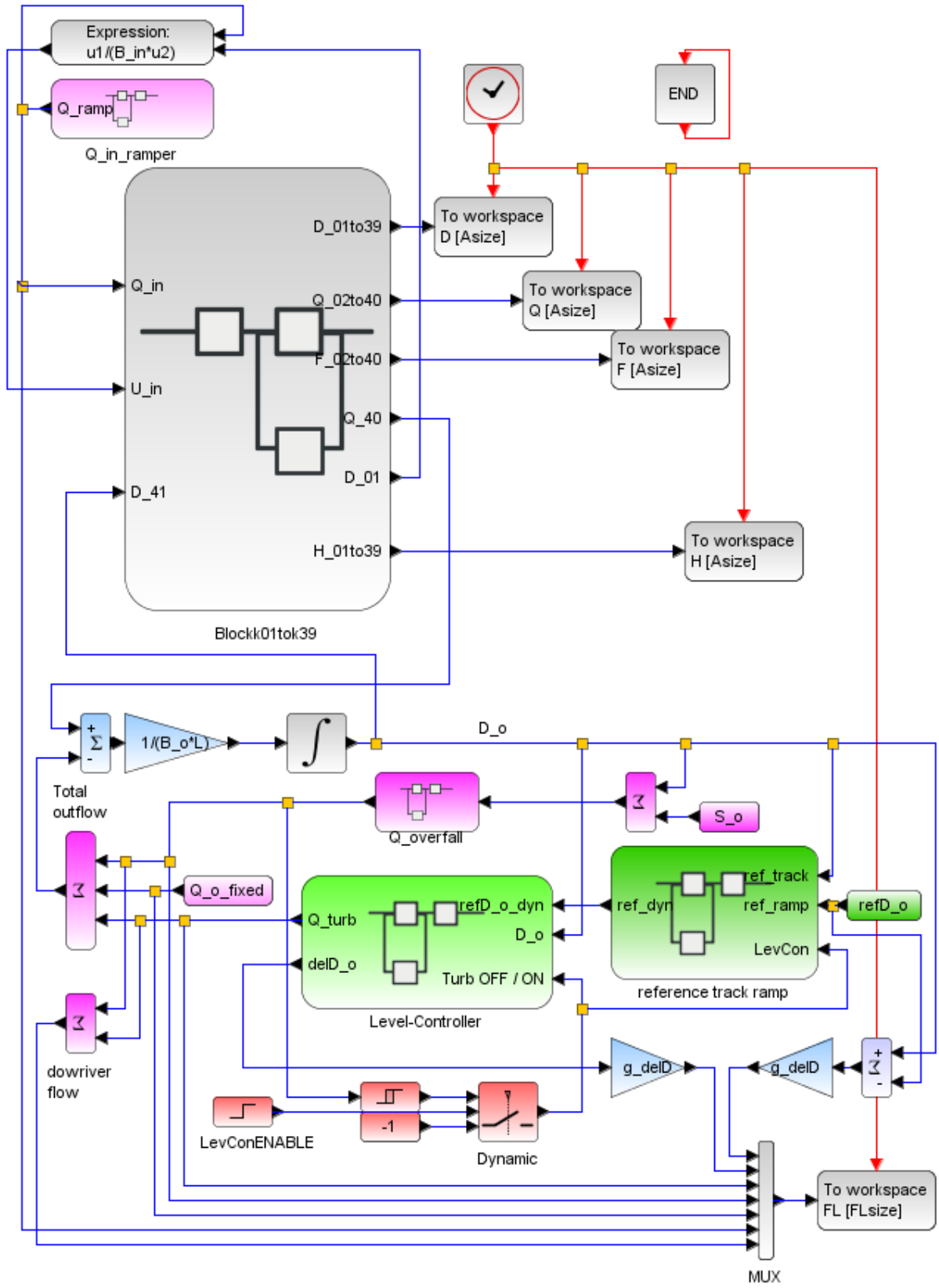


Figure 7.5: Case study "NeueWelt": top level diagram for closed loop responses to **inflow** *z_in*

```
xtitle("D_1 to D_39, Turbine OFF, zoom-in");xgrid(1);
```

```
T_5 = 12200.; T_6 = 14400.;
f8 = scf(8);
plot2d(Q.time,Q.values,vcolor,rect=[T_5,8.,T_6,16.]);
xtitle("Q_2 to Q_40, Turbine ON, zoom-in"); xgrid(1);

f9 = scf(9);
plot2d(D.time,D.values,vcolor,rect=[T_5,0.6,T_6,1.6]);
xtitle("D_1 to D_39, Turbine ON, zoom-in"); xgrid(1);

//*****************************************************
eX = L*(1:1:N); eXend = 21*L;
```

```
for k6= 1:1:N,  vSr(k6) = vS(2*k6-1); end

f10 = scf(10);
vcolL = [9,5,13,1];  oH = H.values;
oH02 = oH(100,:); oH05 = oH(250,:); oH10 = oH(500,:);
plot2d(eX',[oH02',oH05',oH10',vSr],vcolL,rect=[0,-1.0,eXend,+4.0]);
xtitle('Longitudinal Profiles of levels H and bottom S');
h1=legend(['H at  2 000s','H at  5 000s','H at 10 000s','S']);
xgrid(1);
```

7.2.3 Simulation results

Longitudinal profiles

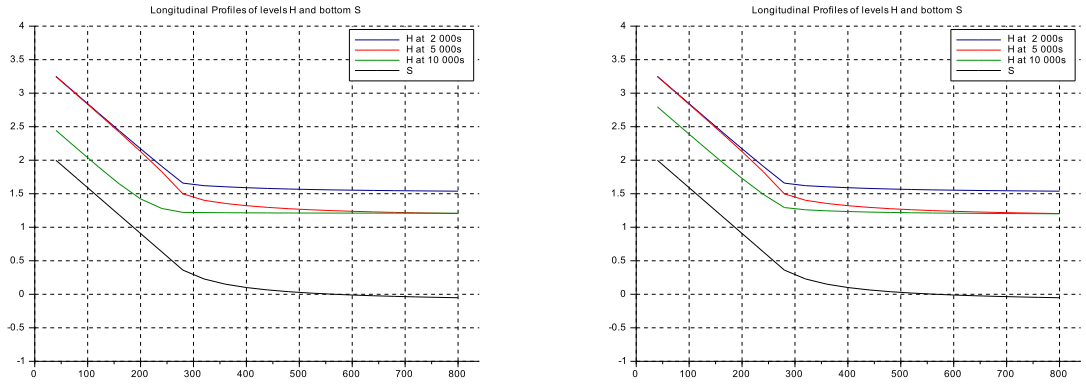


Figure 7.6: Longitudinal profiles of level H and bottom S, (left) test a, (right) test b

Discussion

- Note that the controller gains have been carried over from the previous sections without any additional changes being necessary.
- Note also that the 'reference tracking mode' is enabled/ON when the turbine is OFF. Then the error signal $delD\_o$ of the controller is forced to near zero. Therefore the actual deviation $D\_o(t) - refD\_o$ is plotted in Fig.7.7 as well.
- The closed loop depth response is well damped.
- The overfall flow $Q\_of(t)$ is acceptably close to its specified value 1 m^3/s even for the (very fast) inflow ramps.
- The 'turbine OFF' transient on Q shows a weakly damped oscillation with period of approx. 380 s. Thus the overfall seems not to be very effective in damping the sloshing.
- The 'turbine ON' transient shows a marked overshoot in $Q\_downriver$ as expected. This must be due to the much too fast ramp-up on the inflow.
- Test b shows a good performance of the depth control loop over its operating range. Note that the Froude number F_o relevant for the controller gains stays small (in fact ≤ 0.10). Therefore no gain scheduling is required. Accordingly this add-on has not been used here.

Overviews of Transients

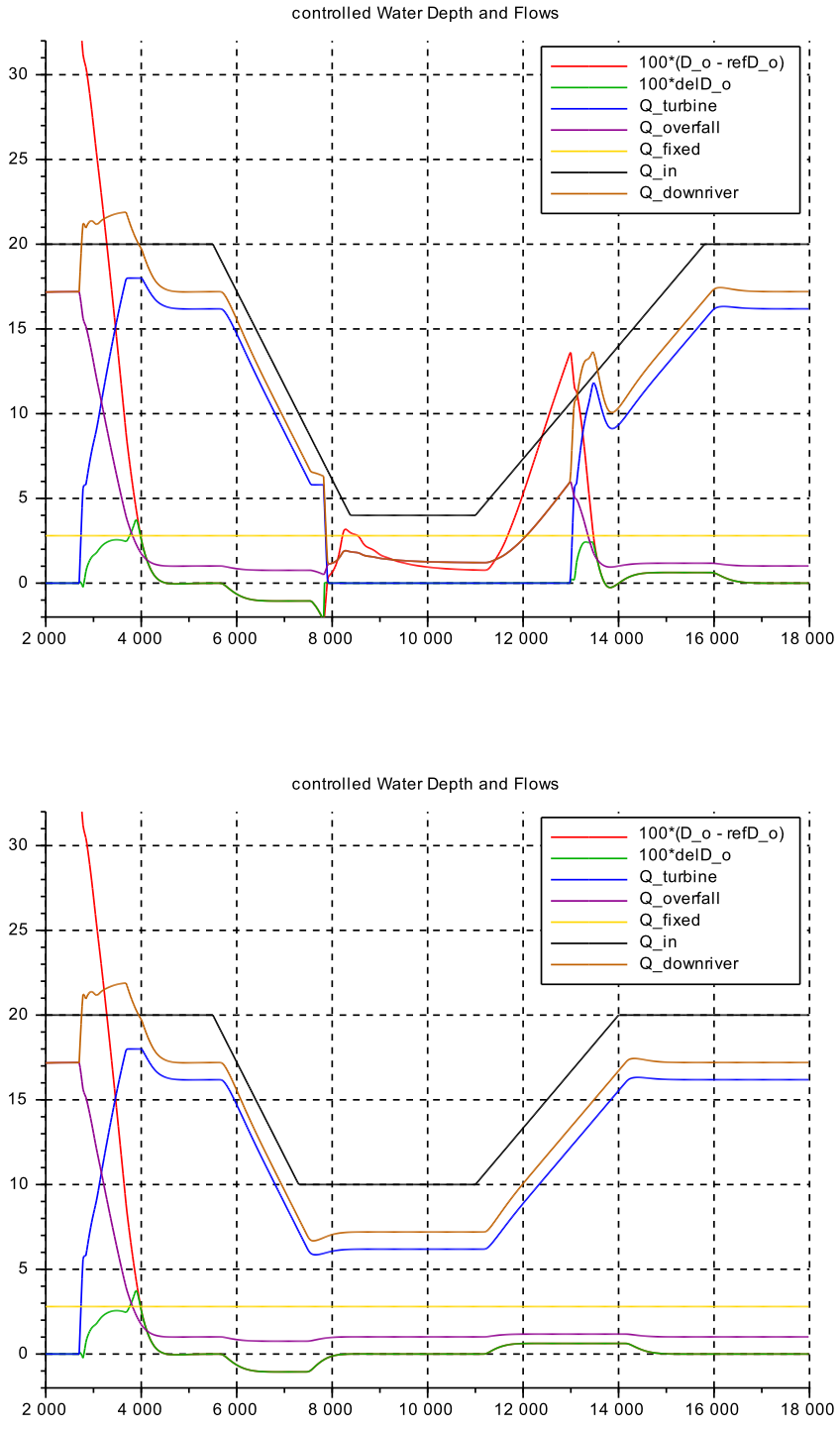


Figure 7.7: Transients on depth and flows
(top) test a, down to $Q\_in = 4\ m^3/s$, (bottom) test b, down to $Q\_in = 10\ m^3/s$
Note: 1.0 units in the plot is 1.0 cm actual depth variation

Details along the reach for test *a*

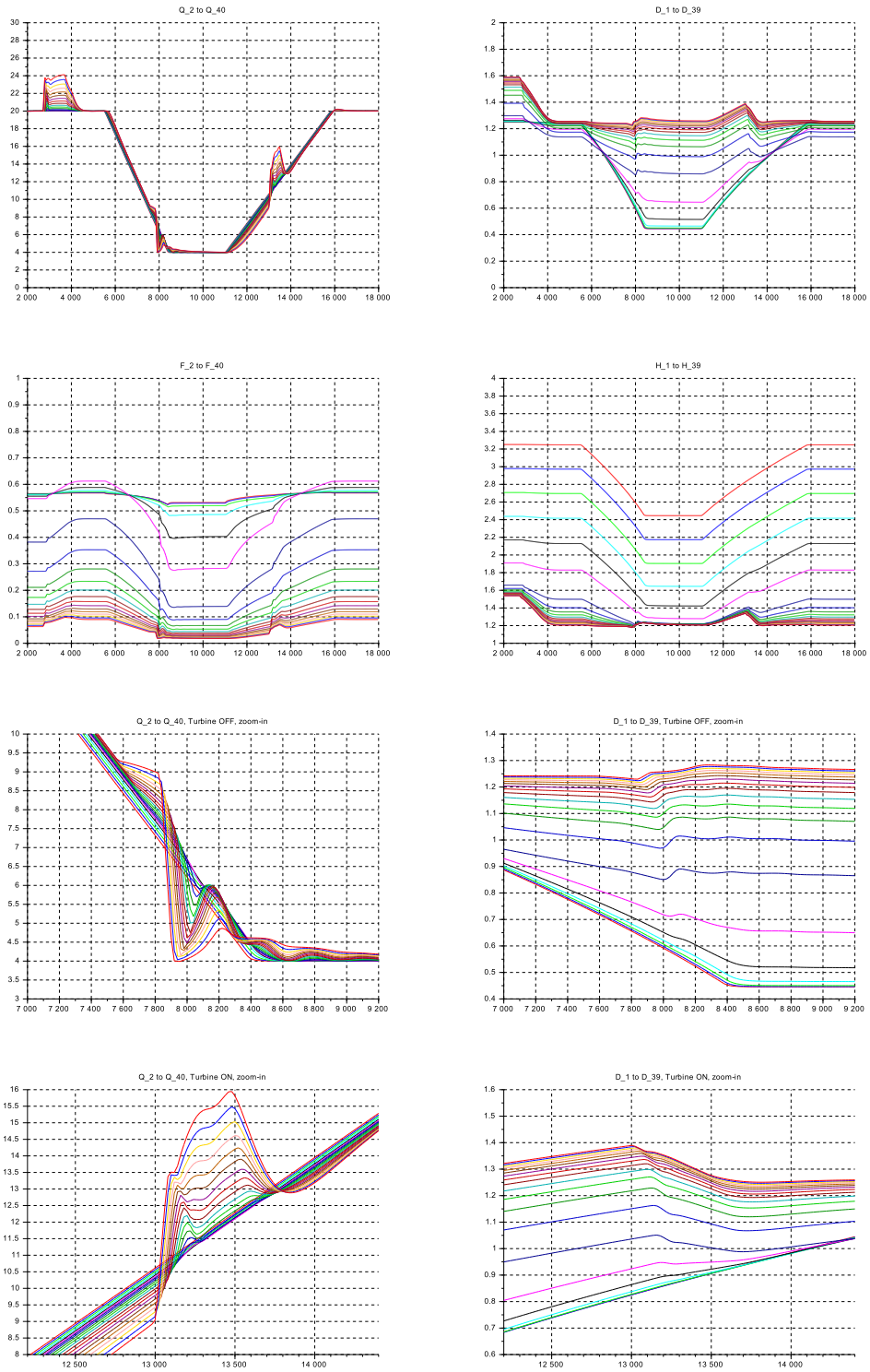


Figure 7.8: Transients for test A
upper block of four: Q, D, F, H for full time sequence
center block of two: zoom-in on Q, D for turbine OFF
lower block of two: zoom-in on Q, D for turbine ON

Chapter 8

Scilab/Xcos ‘s_cX_ ’ and Matlab/Simulink ‘m_cX_ ’

Chapter 2: Basic Longitudinal Element

$Q\_in \rightarrow D\_I$	s_c2_03_00_iter	
rampQ_in	s_c2_03_01	m_c2_01
varied F	s_c2_03_02	m_c2_02
conical cross section	s_c2_03_03	m_c2_03
inlet channel varQ_out	s_c2_03_04	m_c2_04

Chapter 3: Long Channel

colour code	s_c3_41_0x_colorcode	—
Basic Case, slow rampQ_in, N=20	s_c3_41_00	m_c3_41_00
varied N (number of mom.compartm.)	s_c3_41_00_N04	m_c3_41_00_N04
	s_c3_41_00_N08	m_c3_41_00_N08
	s_c3_41_00_N16	m_c3_41_00_N16
	s_c3_41_00_N32	—-
fast rampQ_in	s_c3_41_00_N20	m_c3_41_00_N20
Low and High Q_in	s_c3_41_01	m_c3_41_01
Low to High F	s_c3_41_02	m_c3_41_02
Variable cross section:		m_c3_41_03_s.slx
confusor	s_c3_41_03_1	& m_c3_41_03_1.m
diffusor	s_c3_41_03_2	& m_c3_41_03_2.m
bottom slope	s_c3_41_03_3	& m_c3_41_03_3.m
Power station inlet channel, varied Q_out		
no overfall	s_c3_41_04_2	m_c3_41_04_2
with overfall at inlet	s_c3_41_04_3	m_c3_41_04_3
Application ‘Birsfelden’, open loop		m_c3_41_05_s.slx
1000 m^3/s	s_c3_41_05_10	& m_c3_41_05_10
500 m^3/s	s_c3_41_05_05	& m_c3_41_05_05
200 m^3/s	s_c3_41_05_02	& m_c3_41_05_02

Chapter 4: Wide Channel

Channel & Floodplain, $Q \rightarrow D\_I$	s_c4_caseA_iter.sce	m_c4_caseA_iter.m
Channel & Floodplain, varQ_in	s_c4_caseA	m_c4_caseA

Chapter 5: Short Channel

Spillway contour	s_c5_spillway_contour.sce	m_c5_spillway_contour.m
Case 1: Outflow flap varH_out	s_c5_01_b & _contassemb.sce s_c5_01_c & _contassemb.sce & _contassemb_scaled.sce	m_c5_01_b & _contassemb.m m_c5_01_c & _contassemb.m & _contassemb_scaled.sm
Case 2: inflow level	s_c5_02_b & _contassemb.sce	m_c5_02_b & _contassemb.m
Case 3: 'dam break'	s_c5_03	m_c5_03
Case 4: Surge waves	s_c5_04	m_c5_04
Case 5: HydrJump after weir	s_c5_05	m_c5_05
Case 6: HydrJump after dam	s_c5_06 & contourassembly.sce	m_c5_06 & contourassembly.m

Chapter 6: Control Design

PDE, ***'sys_inf'***

open loop stepresp.	s_c6_01_0_01	m_c6_01_0_01
open loop freqresp.	s_c6_01_0_02.sce	m_c6_01_0_02.m
case A, closed loop, P, freqresp.	s_c6_01_1_01.sce	m_c6_01_1_01.m
case A, closed loop, P, stepresp.	s_c6_01_1_02	m_c6_01_1_02
case A, closed loop, PcI, freqresp.	s_c6_01_1_03.sce	m_c6_01_1_03.m
case A, closed loop, PcI, stepresp.	s_c6_01_1_04	m_c6_01_1_04
case B, closed loop, PcI, stepresp.	s_c6_01_2_01	m_c6_01_2_01
case C, closed loop, PcI, stepresp.	s_c6_01_3_01	m_c6_01_3_01

linear state space, ***'sys_ss'***

open loop step- and freq-resp. N = 100 and N = 20	s_c6_01_0.sce	m_c6_01_0.m
case A, closed loop, PcI, step- & freq-resp.	s_c6_02_1.sce	m_c6_02_1.m
case B, closed loop, PcI, step- & freq-resp.	s_c6_02_2.sce	m_c6_02_2.m
case C, closed loop, PcI, step- & freq-resp.	s_c6_02_3.sce	m_c6_02_3.m

'nonlinear time domain'

open loop step-resp.	s_c6_03_0	m_c6_03_0
case A, closed loop, PcI, step-resp.	s_c6_03_1	m_c6_03_1
case B, closed loop, PcI, step-resp.	s_c6_03_2	m_c6_03_2
case C, closed loop, PcI, step-resp.	s_c6_03_3	m_c6_03_3

Chapter 7: Application Studies

'Birsfelden'		
slow ramping on Q_in	s_c7_01_10	m_c7_01_10
lock cycling by Q_out	s_c7_01_10_lock	m_c7_01_10_lock
'NeueWelt'		
slow ramping on Q_in		
select for low min. flow or intermediate min. flow	s_c7_02	m_c7_02

Bibliography

[1] Aldrighetti, Elisa (2007)
Computational hydraulic techniques for the de Saint Venant Equations in arbitrarly shaped geometry
Ph.D. Thesis, Dipartimento di Matematica, Universita degli Studi di Trento (IT)
download from http://core.ac.uk/download/pdf/11829733.pdf

[2] Crossley, Amanda Jane (1999)
Accurate and efficient numerical solutions for the Saint Venant Equations of open channel flow
Ph.D. Thesis, University of Nottingham (GB)
download from http://eprints.nottingham.ac.uk/10109/1/ajc_thesis.pdf

[3] Beffa, Cornel (2008)
2D-Shallow Water Equations – Basics - Solutions - Applications
download from: http://www.ifu.ethz.ch/EFM/education/Numerical_Hydraulics/NHY_Kapitel_8_EN.pdf

[4] Iaccarino, Gianluca (2015)
Solution methods for the incompressible Navier-Stokes Equations
download slides from: http://stanford.edu/class/me469b/handouts/incompressible.pdf
lecturer: https://profiles.stanford.edu/gianluca-iaccarino?tab=teaching

Method of Lines

[5] Hamdi S, Schiesser W E, Griffiths G W (2009)
Method of Lines
Scholarpedia, 2(7):2859 at www.scholarpedia.org

[6] Schiesser W E (2000)
Partial Differential Equations
Lecture notes from LeHigh University download at: http://www.lehigh.edu/ wes1/apci/28apr00.pdf

Frequency domain

[7] Litrico X, Fromion V (2004)
Frequency modeling of open channel flow
Journal of Hydraulic Engineering, 130, 8, 806-815

[8] Litrico X, Fromion V, Baume JP, Arranja C, Rijo M (2005)
Experimental validation of a methodology to control irrigation canals based on Sait-Venant equations
Control Engineering Practice, 13, 1425 - 1437

[9] Litrico X, Fromion V (2009)
Boundary control of hyperbolic conservation laws using a frequency domain approach
Automatica 45 (2009) 647-656

[10] Bastin G, Coron J-M (2016)
Stability and Boundary Stabilization of 1-D Hyperbolic Systems
Birkhäuser-Springer International Publishing Switzerland 2016, ISBN 978-3-319-32060-1