Online Feedback Optimization for Gas Compressors

Bachelor Thesis

Author(s):
Degner, Maximilian

Publication date:
2021-07-19

Permanent link:
https://doi.org/10.3929/ethz-b-000502040

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Automatic Control Laboratory

Bachelorthesis

Online Feedback Optimization for Gas Compressors

Maximilian Degner

July 19, 2021

Advisors
Prof. Florian Dörfler
Dr. Saverio Bolognani
Dr. Mehmet Mercangöz
Lukas Ortmann

Abstract

Gas compressor stations are vital components of natural gas pipelines. Their compressors are often arranged in parallel or serial configurations to achieve a designated mass flow or pressure ratio. The machines have different, time-varying performance characteristics and by applying carefully chosen inputs to them, the station's energy consumption of can be reduced. In the industry, a two-step real-time optimization method is frequently used to do so. However, in practice this method rarely achieves optimal plant operation due to structural plant-model mismatch.

To tackle the issues caused by plant-model mismatch, other controllers are needed. The performance of a recently proposed feedback optimization algorithm for this type of optimization problems is investigated with a simplified, generic load sharing problem and a compressor load sharing problem. A setup of three machines arranged in parallel is considered for both problems. In the compressor load sharing problem, plant-model mismatch is included by fitting Gaussian process regression models to efficiency maps from manufacturers to serve as hidden, true plant models in simulations. Second order polynomials, fitted to the same data, are used as the models available to the optimizing controllers. The performance of feedback optimization and the two-step optimization method is compared to an equal load sharing controller.

Feedback optimization achieves lower energy consumptions than the two-step optimization method and converges steadily, whereas the two-step approach exhibits jumps between local minima. By tuning the feedback optimization controller adaptively, good convergence rates are possible for all operating points. Therefore, using feedback optimization to solve load sharing problems that involve mechanical systems and structural plant-model mismatch can lead to significant reductions in energy consumption.

Contents

Notation

Nomenclature

$\mathbb{R}^n$	n-dimensional Euclidean space
$B^\top$	transpose of matrix B
B^{-1}	inverse of matrix B
$\mathbb{I}_p$	identity matrix of size $p \times p$
$\lvert\lvert v \rvert\rvert$	2-norm of the vector $v \in \mathbb{R}^n$
$\lvert\lvert \cdot \rvert\rvert_G = \sqrt{v^\top G v}$	norm induced by $G \in \mathbb{R}^{n \times n}$
∇f	Jacobian matrix of a function $f : \mathbb{R}^n \to \mathbb{R}^m$
$(a)^k$	quantity a at discrete time k
m	physical quantity mass
$\frac{dm}{dt} = \dot{m}$	derivative of the physical quantity mass
$\overset{*}{m}$	mass flow

Acronyms

CT	continuous time
CTRT	continuous time reference tracking
ELS	equal load sharing
FO	feedback optimization
GPR	Gaussian process regression
MAC	model adequacy criterion
PI	proportional-integral (controller)
PID	proportional-integral-derivative (controller)
QP	quadratic programming
RTO	real-time optimization
SPW	set-point weight

List of Figures

List of Tables

Chapter 1

Introduction

Gas pipelines are used to transport natural gas over long distances. Friction and altitude changes (e.g., from sea level to regions a few hundred meters above) cause pressure losses which have to be compensated regularly by compressor stations. These stations usually consist of multiple compressors that are arranged in parallel or serial configurations to achieve a desired station mass flow or pressure increase. Compressors are also used in many industries to provide compressed air, e.g., for control and actuation or to compress process gases. Compressors are therefore responsible for a significant energy consumption and strategies that reduce the energy consumption of parallel and serial configurations are needed.

The performance characteristics of compressors can differ from machine to machine and they are subject to time-dependent influences such as fouling, small damages to the compressor blades, or varying gas mixtures. Variable speed gas turbines or electrical variable frequency drives enable different operating points for different machines. With these drives it is possible to use the differences in efficiency to minimize the compressor station's energy consumption [3]. The operating points that minimize the station's energy consumption are the solution to an optimization problem. By implementing online or offline optimizing controllers, locally minimal power consumption can be achieved.

However, the implementation of offline optimizing controllers requires highly accurate models. Compressors in gas networks are only subject to extensive maintenance after they have been used for years. This in-depth maintenance usually happens after fifteen to twenty years of operation, unless there is a major failure. Measurements that could be used to identify a new offline model are only possible within the scope of these maintenance sessions. Therefore, a feed-forward optimization approach will not always deliver inputs that (locally) minimize the energy consumption.

There are a few additional factors which make the optimization of compressor stations challenging:

- The performance of a compressor is expressed by an efficiency map that depends on compressor mass flow and pressure ratio. It is time-varying and a correct analytical representation of this relationship is not available (only measurement-based maps are available).

- Low-level controllers (implemented to prevent compressor surges) can cause disturbances. The optimal inputs in the presence of disturbances can be different from the ones calculated in the absence of disturbances.

- Compressors have non-linear dynamics with time constants in the range of seconds. The controllers have to run with a sampling time that is larger than this time constant to ensure timescale separation. Ideally, the optimal torques are applied to the compressors at the first iteration after a reference change, such that minimal power consumption is reached quickly.

Similar load sharing problems can be found for pumps in water desalination plants, turbines and generators in power plants, bigger air conditioning systems, and cooling systems of cold storage. The goal of applying optimization methods to these systems is to minimize the total energy consumption. Additional operational constraints on the control inputs or safety relevant boundaries might apply.

The time-varying compressor efficiency maps, disturbances, and possibly changes of the constraints in between maintenance sessions, make the usage of offline optimizing controllers practically impossible. A few online optimization approaches which can be used for this load sharing optimization problem were developed. Examples include modifier adaptation [9], [10], and specialized real-time optimization (RTO) algorithms [3], [8]. However, most of these approaches require high computational efforts as complex non-linear and possibly non-convex optimization problems have to be solved.

Recently, online feedback optimization (FO) has been introduced in the field of power system control [1], [6]. It combines aspects from optimization theory with control theory and aims at delivering controllers that are more robust to disturbances and uncertainties than numerical optimization applied in a feed-forward manner. The approach presented in [6] requires less computational effort than RTO approaches, because only a quadratic program has to be solved.

The focus of this work lays on the application of the feedback optimization algorithm from [6] to gas compressors that are arrange in parallel configurations. The controller is used for reference tracking and a few additions are made regarding the implementation. Therefore, this work is structured as follows: In the second chapter, the industrial standard for compressor load sharing optimization is introduced. This is followed by recalling FO in the third chapter and applying both methods to a generic load sharing problem in chapter four. The fifth chapter contains an introduction to the modeling of gas compressors, the corresponding optimization problem, the application of FO to gas compressors, and the comparison of the different controllers. The thesis is concluded in chapter six. Some additional information regarding the controller implementation can be found in the appendix.

Chapter 2

Current State-of-the-art for (Compressor) Load Sharing Optimization

In this chapter, the industrial standard for the optimization of compressor load sharing will be recalled. The idea of the approach and its properties are presented.

2.1 Two-Step Approach

An intuitive approach to finding the solution of a time-varying optimization problem would be to regularly update the modeled optimization problem with data obtained during operation. With this updated model, it would be possible to search for the minimizer of an objective function, e.g., the energy consumption.

This idea can be transformed into an optimizing controller. A standard approach is to use nonlinear regression to estimate the parameters of a system's model, however linear least squares can be used too. Parametric plant-model mismatch can be diminished as the parameters are adjusted such that the model is locally as close to the plant as structurally possible. Afterwards, a constrained optimization problem is solved [9]. In the most general case, this is a nonlinear program. As this approach consists of two separate steps, we will refer to it as the *two-step approach*, which can be classified as an adaptive RTO method. RTO methods also contain approaches that only solve an optimization problem during plant operation and do not update their model or problem. Thus, "adaptive" indicates, that the constrained optimization problem is adjusted to follow the physical system.

2.2 Properties

A crucial point for every optimization approach is the quality of the calculated set-points, i.e., whether the minimizer is a set-point leading to optimal plant operation. In order to reach such results with the two-step approach, some prerequisites must be met:

1. The system excitations have to be sufficient for the estimation of uncertain parameters [10],
2. the structural plant-model mismatch after the model update must not be significant [10], and
3. the used model must be *adequate* [8].

By default, these conditions are not fulfilled. In this context, "adequate" refers to the *model adequacy condition* (MAC), which demands that the RTO problem matches the Karush-Kuhn-Tucker points at the plant optimum and that it has a positive definite Hessian matrix at this point. Tools to check this condition are introduced in [4].

> *MAC [4, Definition 2.1]:* If a process model can produce a fixed point which is a local minimum of the optimization problem at the (generally unknown) plant optimum, it is adequate for the use in a RTO scheme.

If the MAC are not satisfied, the calculated inputs lead to sub-optimal operation. It is not clear before the parameter estimation, whether the MAC are fulfilled or not. To ensure locally optimal inputs, a check would be needed before every optimization step. Consequently, general guarantees for optimal plant operation cannot be given for this method [8]. In practice, optimality upon convergence is rare [7].

Another notable aspect is the computational effort needed to compute plant inputs with the two-step approach. In general, the RTO problem has to be solved by nonlinear programming. It is one of the most general constrained optimization problems and allows for non-linearity and non-convexity in the objective function and the constraints. Widely used solvers are based on methods that divide the optimization problem into subproblems and exploit certain properties of these. Therefore, even finding a local solution may be computationally demanding.

Chapter 3

Feedback Optimization

3.1 General Approach

The main idea of feedback optimization (FO) is to combine traditional feedback-based control theory and optimization theory. The feedback is used to improve the robustness to uncertainties and disturbances when compared to feed-forward numerical optimization [6]. FO does not have to be based on one specific optimization algorithm, but can be based on gradient flows [6], saddle-flows [2], [11], and possibly other methods. Projections are usually included to enforce constraints strictly. Note that the projections are implemented differently for continuous and discontinuous time.

For a gradient flow in continuous time, the gradient is projected onto the tangent cone of the feasible set. In other words, the gradient is compared to all feasible directions and if it is not feasible, the closest feasible direction will be followed. In discontinuous time, a slightly different concept is needed: The gradient is used to calculate the next point without considering the constraints that are imposed by the projection. Then, the closest feasible point is searched, which is equivalent to projecting the calculated point onto the feasible set.

Within the scope of this work, the discrete-time controller as presented in [6] is used. It will be recalled in this section. To emphasize that the inputs and outputs of the plant are discrete in time, the superscript $(\cdot)^k$ is used for a quantity at the k-th time step.

We consider any constrained optimization problem of the form

$$\begin{aligned} \overset{\diamond}{u} = \arg\min_{u \in \mathbb{R}^p} \; & \Phi(u, y) && \text{with } y = h(u) && \text{(3.1a)} \\ \text{subject to} \quad & A \cdot u \leq b && && \text{(3.1b)} \\ & C \cdot y \leq d, && && \text{(3.1c)} \end{aligned}$$

where $y = h(u)$ is the measured system output, $A \in \mathbb{R}^{q \times p}$, $C \in \mathbb{R}^{l \times n}$, $b \in \mathbb{R}^q$, and $d \in \mathbb{R}^l$.

This optimization problem is solved by the FO algorithm which consists of an integral control law and a projection onto a linearization of the feasible input set. The control law has an integration step size α and reads

$$u^{k+1} = u^k + \alpha \cdot \widehat{\sigma}_\alpha(u^k, y^k) \quad \text{with } y^k = h(u^k). \tag{3.2}$$

The projection can be written as an optimization problem with constraints on inputs u^k and outputs y^k

$$\widehat{\sigma}_\alpha(u,y) = \underset{w\,\in\,\mathbb{R}^p}{\arg\min}\;\; s(w) = \underset{w\,\in\,\mathbb{R}^p}{\arg\min} \left|\left| w + G^{-1}(u)H(u)^\top \nabla\Phi(u,y)^\top \right|\right|^2_{G(u)} \tag{3.3a}$$

$$\text{subject to}\quad A\left(u^k + \alpha w\right) \leq b \tag{3.3b}$$

$$C\left(y^k + \alpha\nabla h(u^k)w\right) \leq d, \tag{3.3c}$$

where $H(u^k)^\top = \left[\mathbb{I}_p \;\nabla h(u^k)^\top\right]$, and $\nabla(\cdot)$ is the Jacobian matrix. Additionally, one has to define a metric on the input space $G(u^k) \in \mathbb{R}^{p\times p}$. In most cases $G = \mathbb{I}_p$ is a reasonable choice. To improve readability, the superscript $(\cdot)^k$ is dropped in (3.3a).

For $\alpha \to 0^+$, this FO method converges to a continuous time projected gradient flow, and for any $\alpha < \alpha^{\text{upper}}$, it converges to a steady-state feasible solution which corresponds to a minimum of the objective function [6, Theorem 3]. The computation of the optimal inputs for a given plant benefits largely from two properties: Firstly, the numerical optimization problem in (3.3a) can be rewritten as a quadratic programming (QP) problems and therefore solved in real-time by established QP-solvers with little computational effort. Second, only the steady-state input-output sensitivities $\nabla h(u^k)$, rather than the full model, are required to calculate $\widehat{\sigma}_\alpha(u,y)$. As shown in [11], an estimation of these sensitivities can be enough to obtain feasible solutions.

3.2 Reformulation as Quadratic Programming Problem

The general structure of a QP reads

$$\overset{\diamond}{x} = \underset{x}{\arg\min}\, \frac{1}{2}x^\top Qx + c^\top x + e \tag{3.4a}$$

$$\text{subject to}\quad A_1 x \leq b_1 \tag{3.4b}$$

$$A_2 x = b_2, \tag{3.4c}$$

where $A_1 \in \mathbb{R}^{m_1\times n_1}$, $A_2 \in \mathbb{R}^{m_2\times n_2}$, $Q = Q^\top \in \mathbb{R}^{n_3\times n_3}$, $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$, and $e \in \mathbb{R}$. Without loss of generality, $e = 0$ can be chosen, because the constant term does not change the minimizer $\overset{\diamond}{x}$.

To obtain the corresponding FO formulation, the norm has to be used in its square-root representation $||\zeta||^2_G = \sqrt{\zeta^\top G\zeta}^2$. The following equality (3.5) holds if the number under the square root is positive and real, i.e., $0 \leq \left(\zeta^\top G\zeta\right) \in \mathbb{R}$. This is always the case if G is positive definite.

$$\left(\sqrt{\zeta^\top G\zeta}\right)^2 = \zeta^\top G\zeta \tag{3.5}$$

By using $\zeta = w + \beta$ with $\beta = G^{-1}(u^k)H(u^k)^\top\nabla\Phi(u^k,y^k)^\top$ from (3.3a) we get

$$\left(\sqrt{(w+\beta)^\top G(w+\beta)}\right)^2 = \left(\sqrt{w^\top Gw + 2\beta^\top Gw + \beta^\top G\beta}\right)^2, \tag{3.6}$$

where the term $\beta^\top G\beta$ does not depend on w. Consequently, it is not relevant to find the minimizer of $s(w)$ from (3.3a) and $\beta^\top G\beta$ can be omitted. By combining (3.5), (3.6), and the fact that G is a metric, we get

$$
\begin{aligned}
\widehat{\sigma}_\alpha &= \arg\min_w \frac{1}{2} w^\top (2 \cdot G)\, w + \left[\left(G^{-1} H(u^k)^\top \nabla\Phi(u^k, y^k) \right)^\top G \right]^\top w && (3.7a) \\
&= \arg\min_w \frac{1}{2} w^\top (2 \cdot G)\, w + \left[G^\top G^{-1} H(u^k)^\top \nabla\Phi(u^k, y^k) \right] w && (3.7b) \\
&\text{subject to} \quad A \left(u^k + \alpha w \right) \leq b && (3.7c) \\
&\qquad\qquad\quad C \left(y^k + \alpha \nabla h(u^k) w \right) \leq d. && (3.7d)
\end{aligned}
$$

When comparing the general QP problem with the QP formulation of FO, one notices that inputs and outputs of the plant are present in the constraints for the latter formulation. In the general problem, the constraints are formulated as constraints on x. It is possible to meet this form, as the variable w (now used instead of x) is part of both inequalities. The constraints in (3.7) have to be rearranged such that all terms that do not depend on w are on the right hand side and only terms with w are on the left hand side. When using a QP solver, the constraints have to be passed as one matrix and one vector which can be obtained by concatenating A, C and b, d. To emphasize that the constraints have to hold at the next point, they are not presented in a rearranged and concatenated form.

3.3 Additions

Applying FO to the load sharing problems presented in Chapters 4 and 5 requires a few considerations. These include the tuning of the integration step size and the tracking of an external reference.

Feedback optimization is a discrete-time method and has (in addition to the sampling time T_s) one intrinsic tuning parameter. This parameter is the integration step size α. The sampling time has an influence on the integration behavior as it determines how often a step with step size α is performed per unit of time. A lower sampling time leads to a faster integration (even for a constant α) and therefore faster dynamics of the optimizing controller. To ensure stable operation, one has to find a combination for α and T_s, that satisfies the timescale separation and the stability bound defined in [6, Lemma 5]. Timescale separation refers to the assumption that the plant has fast-decaying dynamics which do not have to be considered for the implementation of an optimizing controller. If this assumption holds, the plant can be described by its steady-state input-output relationship.

However, a suitable combination of these two parameters does not guarantee convergence to an optimal set-point in a sufficiently small amount of time for all operating points. For one part of the operating range of the considered load sharing problems, the convergence rate is sufficient, but for another range it is not. Therefore, the value of α is varied according to the current operating point and when a new load target is given (i.e., the station reference changes). In our implementation we used a heuristic approach consisting of if-statements.

This heuristic increases α if the operating point is in a certain region, e.g., the generic load is below 250. The integration step size is also increased if a set-point change occurs; i.e., α is doubled for the first calculation of new inputs after the station reference changed. This leads to better reference tracking behavior because the plant gets an input that is closer to the final value without causing oscillations.

Compressor stations can work with station mass flow references, that is a reference which has to be tracked by all machines together. The optimizer has to find inputs for the individual machines which not only minimize the energy consumption of the station, but also produce the requested mass flow. To the best of the author's knowledge, there is no literature available that used the FO algorithm from section 3.1 for reference tracking. The solution is straight forward: The reference can be included as a constraint of the optimization problem. If this constraint is formulated on the inputs, a disturbance could alter the stations output such that it is not equal to the reference. In this case, the controller is not able to reject the disturbance. To diminish this problem, the constraint can be formulated on the outputs. This is only possible if one of the outputs is a quantity that can be compared to the station reference.

Another way to implement reference tracking and disturbance rejection focuses on using a continuous-time PI or PID controller for continuous time reference tracking (CTRT). These controllers track the reference in an equal load sharing manner and the optimizing part is used to compute modifications to the CTRT inputs. Compared to the version where FO is the sole controller, a slightly different set of constraints is necessary.

Chapter 4

Generic Load Sharing Problem

4.1 Background and System Structure

The performance of feedback optimization is examined with the aid of a generic load sharing problem as it could be found in the load sharing problems of air conditioning systems, pumps, or turbines. All of these applications have in common, that machines with different performance characteristics are used. These differences can result, e.g., from manufacturing or different capacities. Different capacities might be beneficial as, for example, the required amount of cooling of a cold storage changes with the outside air temperature. Here, an example of three machines working in a parallel configuration is studied where the machines have slightly different capacities. Compared to the compressor load sharing optimization discussed in Chapter 5, the generic problem is a simplified version that focuses on key concepts rather than application specific details.

4.1.1 Plants

The three plants consist of two parts: Dynamics based on a generator set and a low-level PI controller for tracking the individual machine's reference. The dynamics are simplified as a first order low-pass element. The machine's individual reference l_i^{ref} is, e.g., one third of the station reference in the equal load sharing case. The PI controller is necessary, because the optimizing controller is a high-level controller and should not track the reference of an individual machine, but the station reference. The input of each plant is a load reference $u_i = l_i^{\text{ref}}$. The output of each plant consists of its actual load l_i and its efficiency $\eta_i = f_i^{\eta}(l_i)$, i.e., $y_i = [l_i, \eta_i]$.
To include the different performance characteristics, all three machines have different dynamics and different efficiency curves. The details can be found in Appendix A.

4.1.2 Objective Function and Constraints

The cost function of this generic problem comprises the energy consumption of the three machines $\Phi(l_i) = \sum_{i=0}^{3} P_i(l_i, \eta_i)$.
The load range of the individual machines is bounded by a box constraint and the sum of the three loads l_i should follow the station reference $r(t)$. A formalization can be found in equations (4.1) and (4.2).

$$l_i \in \left[l_i^{\text{lower}}, l_i^{\text{upper}}\right] \tag{4.1}$$

$$\sum_{i=0}^{3} l_i \approx r(t) \tag{4.2}$$

The constraint (4.1) is implemented according to the different machine capacities

$$\begin{aligned} l_1^{\text{lower}} &= l_2^{\text{lower}} = l_3^{\text{lower}} = 0 \\ l_1^{\text{upper}} &= 100, \quad l_2^{\text{upper}} = 95, \quad l_3^{\text{upper}} = 90. \end{aligned}$$

4.2 Controller Implementation

In this section, the implementation of different controllers will be presented. The equal load sharing controller serves as a base case, where no optimization is used. Three optimizing controllers are implemented, of which one is the two-step approach and two are based on feedback optimization. One FO implementation incorporates reference tracking by using constraints (section 4.2.3), the other one uses a PID controller for continuous time reference tracking (CTRT) (section 4.2.4).

Common among the different implementations is that a disturbance signal can be passed to each compressor directly, and that the feedback to the optimizing controllers is done with Simulink's "GoTo"-block. Thus, the feedback is not visible in the provided block diagrams.

4.2.1 Equal Load Sharing

The equal load sharing case is the base case, where no optimization is used, but all machines get the same input. Here, a PI controller is used to track the station reference $r(t)$. Each compressor is supplied with one third of the calculated input. Therefore, if a disturbance is present and the sum of the outputs differs from the station target, the PI controller increases the load on all three machines.

The block diagram in Figure 4.1 visualizes the equal load sharing setup. The total load and the total power consumption are passed to a higher level, where a comparison with other controllers is made.

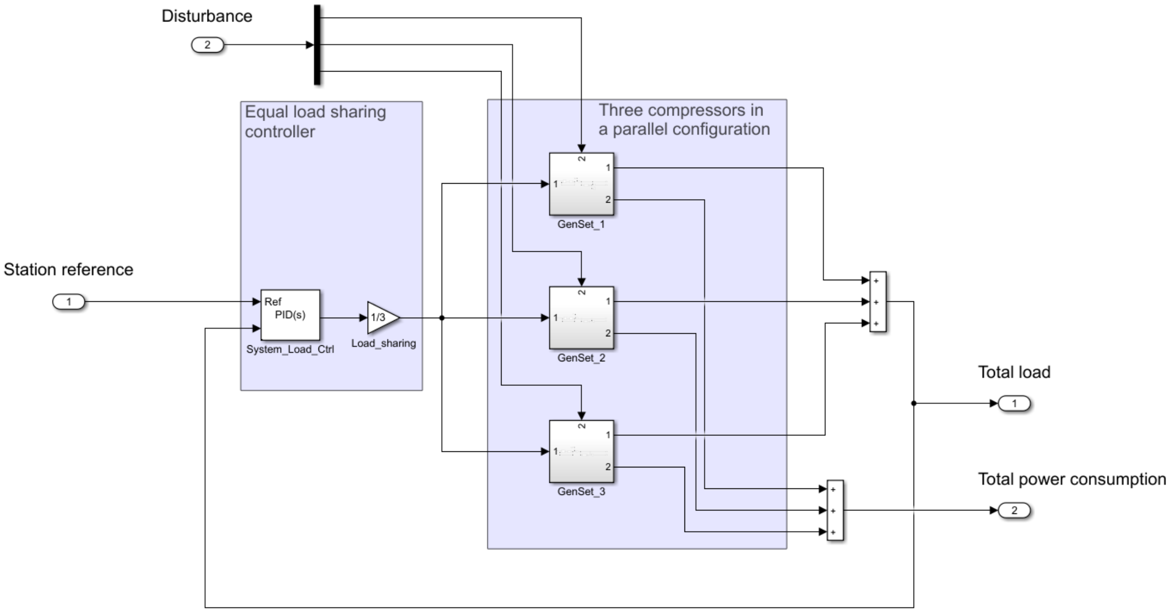


Figure 4.1: Block diagram showing the system in the equal load sharing case

4.2.2 Two-Step Approach

The industrial standard is implemented based on the concept described in Chapter 2. The controller consists of two parts, as shown in the block diagram in Figure 4.2. The parameter estimation and the calculation of the optimal plant inputs are done in the upper block. This block contains an s-function which allows the usage of customized algorithms. A continuous-time PI controller acts in parallel to track the station reference. Similar to the implementation of equal load sharing, one third of the PI controller's output is passed to each machine. The signals are added within the three plant sub-blocks. The controller receives the individual loads and efficiencies as feedback.

The performance characteristics of the plants are contained in their efficiency curves which are described by a polynomial function $\eta_i = f_i^{\eta}(l_i)$. The parameter estimation step uses measurements (actual loads and efficiencies) and linear least squares, to approximate this curve with a polynomial $\eta_{\mathrm{e},i} = f_{\mathrm{e},i}^{\eta}(l_i)$ of a different order. By choosing a polynomial with a different order than the plants' curves, structural plant-model mismatch is introduced. For this implementation, a third order polynomial is used, whereas the actual efficiency polynomials in the efficiency-calculation parts of the three plants have an order of four.

The objective function is the total energy consumption. It includes the estimated efficiency

$$\Phi_{\text{two-step}}(l_i, \eta_{\mathrm{e},i}(l_i)) = \sum_{i=0}^{3} P_i = \sum_{i=0}^{3} \frac{l_i}{\eta_{\mathrm{e},i}}. \tag{4.3}$$

The optimization step only runs if the residuals of the linear least squares regression are below a defined threshold. The plants' inputs are calculated with the nonlinear optimization algorithm `fmincon` from MATLAB's Optimization Tool Box. The constraints from (4.1) and (4.2) are passed to this optimization algorithm. Even though the plant inputs are calculated, only the difference to the current loads l_i is passed to the plants. This is due to the PI controller acting in parallel.

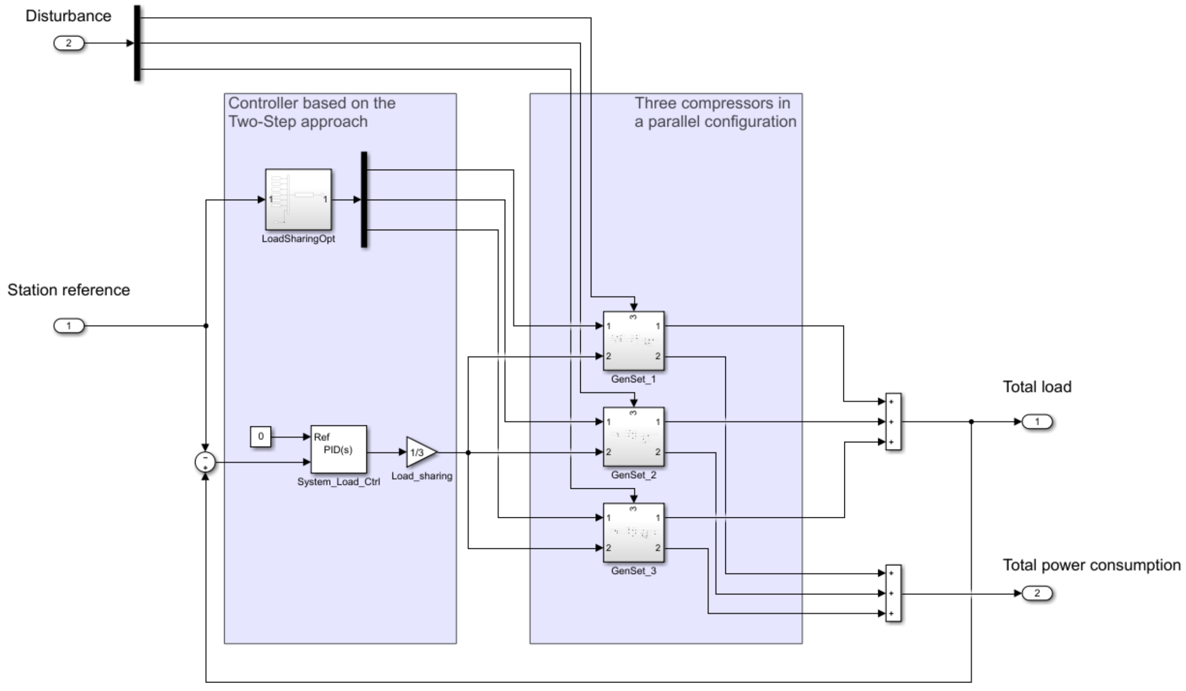


Figure 4.2: Block diagram of the plants and the controller based on the two-step approach

4.2.3 Feedback Optimization

This implementation of FO follows the approach described in Chapter 3 closely because the task of tracking the station reference is included within the formulation of the optimization problem as a constraint. The controller is written as an s-function, which calls an additional function that returns the solution of the QP from (3.7). A block diagram of the system can be found in Figure 4.3.

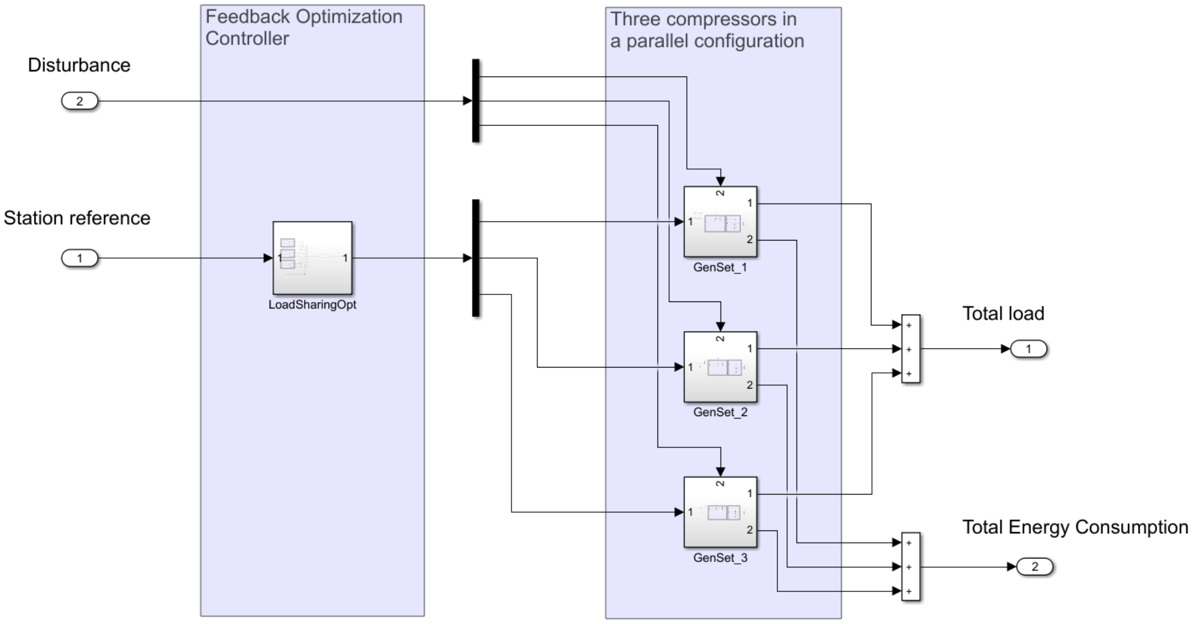


Figure 4.3: Block diagram of the plants and the feedback optimization controller

The objective function for this controller only depends on the three loads l_i; i.e., the efficiency polynomials are directly included in the objective function (4.4). Here, it is possible to use the true efficiency curves and therefore assume that an ideal plant model is available. It is also possible to use an approximation of the true efficiency curves (e.g., polynomials similar to the ones used by the two-step approach). If the coefficients differ from the true coefficients, parametric plant-model mismatch is present, if the polynomial has a different order, structural mismatch is present.

$$\Phi_{FO}(l_i) = \sum_{i=0}^{3} \left(\frac{l_i}{f_{e,i}^{\eta}(l_i)} \right) \tag{4.4}$$

The objective function only depends on the loads (the efficiencies are not considered as plant outputs here) and low-level controllers are used to track the machines' individual reference. This leads to the simple steady-state input-output relationship

$$y = h(u) = u, \tag{4.5}$$

where $y = [l_1, l_2, l_3]^\top$ and $u = [l_1^{\text{ref}}, l_2^{\text{ref}}, l_3^{\text{ref}}]^\top$.
FO requires a gradient of the objective function. This gradient is computed analytically by using symbolic expressions. It is then evaluated for the measured loads l_i. Further, the steady-state input-output sensitivities are required. These can easily be calculated to $\nabla h(u^k) = \mathbb{I}_3$ when deriving (4.5) by the inputs u_i.

The box constraint on the loads (4.1) is implemented as inequality constraints on the inputs and adapted according to (3.7c). The reference tracking constraint from (4.2) is implemented as two inequality constraints on the plants' outputs, leading to the narrow box constraint $\sum_{i=0}^{3} l_i \in [0.9999 \cdot r(t),\ 1.0001 \cdot r(t)]$. Formulating the constraint on the outputs enables the usage of measurements (i.e., actual loads) when adapting the constraint according to (3.7d). Together with (3.7b), the constraints are given to a QP solver which calculates the inputs.

4.2.4 Feedback Optimization with Continuous Time Reference Tracking

This implementation of FO is inspired by the two-step approach and includes CTRT. FO is responsible for calculating the optimal inputs, whereas tracking the station reference $r(t)$ is handled by a high-level continuous-time PID controller which acts in parallel to the optimizing part. It is implemented as an s-function and as an additional function, like in the previous subsection. The block diagram in Figure 4.4 visualizes the system's structure and shows the reference tracking part and the optimizing part of the controller. As in the two-step implementation, the two input signals are added within each plant sub-block.

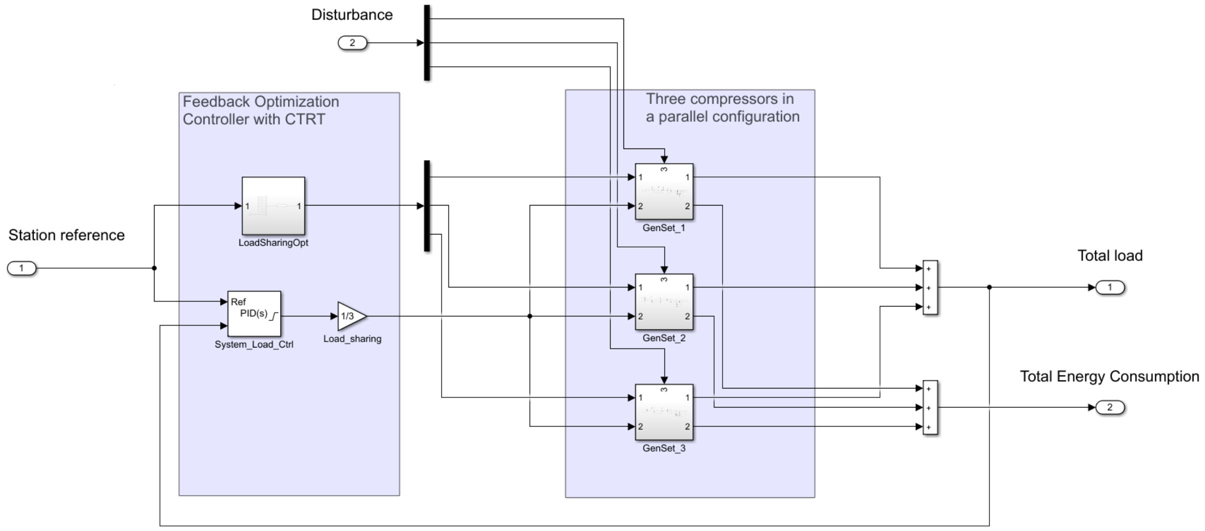


Figure 4.4: Block diagram of the plants and the FO controller with CTRT

The objective function for this implementation is (4.4). Again, its gradient is computed symbolically and the steady-state input-output sensitivities are $\nabla h(u^k) = \mathbb{I}_3$.

The constraint (4.1) is used to limit the inputs of each compressor to a certain range. It is implemented with inequality constraints on the inputs. The reference tracking constraint (4.2) is not necessary here, as this task is fulfilled by the PID controller. However, it is necessary to ensure that the sum of the individual load references l_i^{ref} is equal to zero. This is done by applying a constraint on the inputs that demands that the sum of the calculated values is approximately zero. To ensure that the plants get feasible inputs from the controller, the PID block has an output saturation corresponding to the box constraint given above.

4.3 Results and Data Analysis

4.3.1 Reference Tracking

The reference tracking behavior of the different controllers is compared by looking at four properties of the step responses.
The rise time t_r is defined as the time it takes for the system's output to get from 10 % to 90 % of the step height κ. The settling time t_s is the time the output needs to stay within ±1 % of the step height. The overshoot M_p is defined as the difference between the reference and the highest peak. The peak time t_p is the time between the reference step and the output reaching its highest value. It only exists if an overshoot is present.
Based on a step at time $t = 240\,s$ with step height $\kappa = 50$, the four parameters were measured for all controllers. The results can be found in Table 4.1 and the system output is depicted in Figure 4.5.

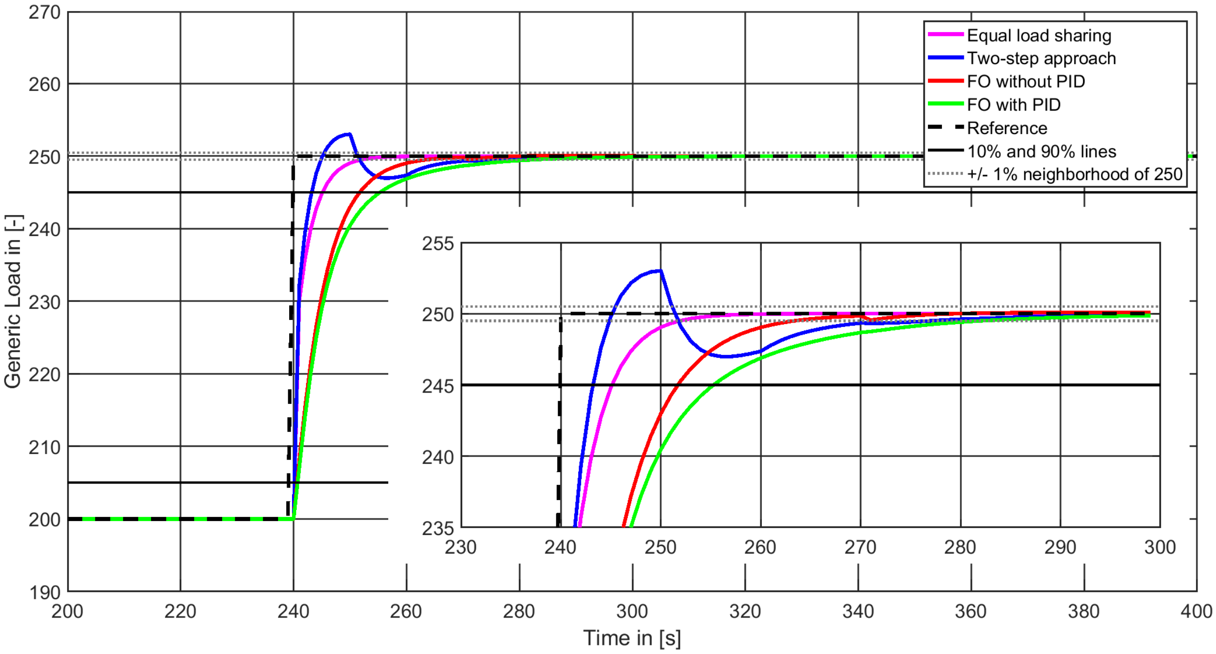


Figure 4.5: System outputs for a step of height $\kappa = 50$ at time $t = 240\,s$

Controller	Rise time t_r [s]	Settling time t_s [s]	Overshoot M_p [-]	Peak time t_p [s]
Equal load sharing	5.1	11.9	0	–
Two-step approach	3.2	37.0	3.02	10.0
Feedback optimization	11.1	23.3	0.10	60.0
Feedback optimization with CTRT	14.3	41.7	0	–

Table 4.1: Parameters characterizing the step response of the different systems for a step with height $\kappa = 50$

The system outputs and the rise times in the table show that the two-step approach leads to fast responses. The rise time is in the range of equal load sharing. However, the two-step

controller also produces overshoots of more than 6 % of the step's value. The two FO-based methods rise slower and do not cause significant overshoots. The settling time for all optimizing controllers is higher than for the equal load sharing case and ranges from 23.3 s to 41.7 s. The FO implementation without CTRT has the lowest settling time among the optimizing controllers.
Another aspect of reference tracking involves the rejection of disturbances. To qualitatively study the behavior of the different controllers, a disturbance signal consisting of steps and a ramp function is applied to the compressors. This signal is depicted in Figure 4.6, the system outputs in Figure 4.7. The disturbances start after 250 s to give the system enough time to reach steady state.

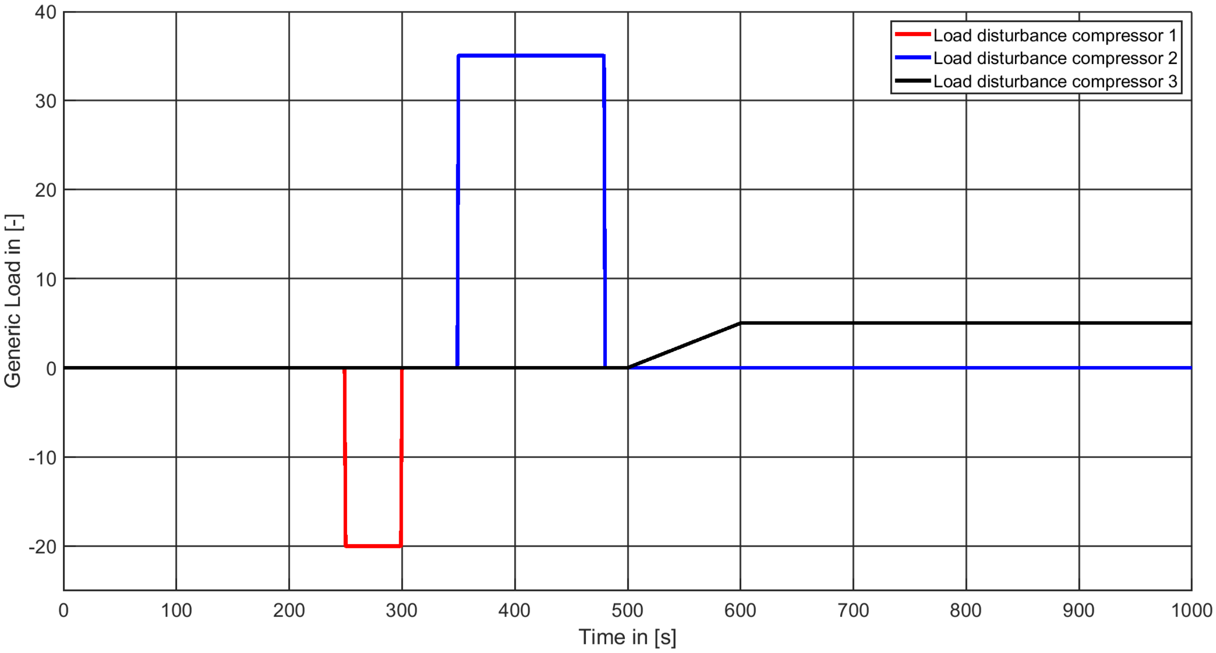


Figure 4.6: Disturbances used to check the disturbance rejection

All controllers are able to react to the disturbances and adjust the inputs such that the station reference is reached. The controllers that use CTRT react faster to the disturbances than the FO controller without a parallel PID controller. Notable in this context is the disturbance on compressor 1 from 250 s to 300 s because the system output shows that the FO controller without CTRT does not react to the disturbance for a prolonged period of time. The controller cannot react earlier, because it runs in discrete time. The controller runs at 240 s and 270 s which leads to the sampling delay of 20 s. All controllers handle the ramp disturbance best and with little to no visible influence on the total station load.

4.3.2 Convergence

The considered generic load sharing problem leads to different convergence behavior based on the value of the reference. Figure 4.8 shows the resulting power consumption if no plant-model mismatch is present for the FO-based controllers. The reference signal used to obtain these trajectories is the sine wave $r(t) = 100 \cdot \sin(0.025 \cdot t + 1) + 150$, sampled every 540 s. For reference, the power consumption of equal load sharing is given. A numerical optimum is calculated using MATLAB's function `fmincon` and included as well.
Qualitatively, the differences in convergence speed are clearly visible from the graph in Figure 4.8. For most reference changes, FO has good convergence speeds, only for the changes at 1620 s, 2700 s, and 3240 s, abnormalities are visible. At the beginning of the simulation, the two-step

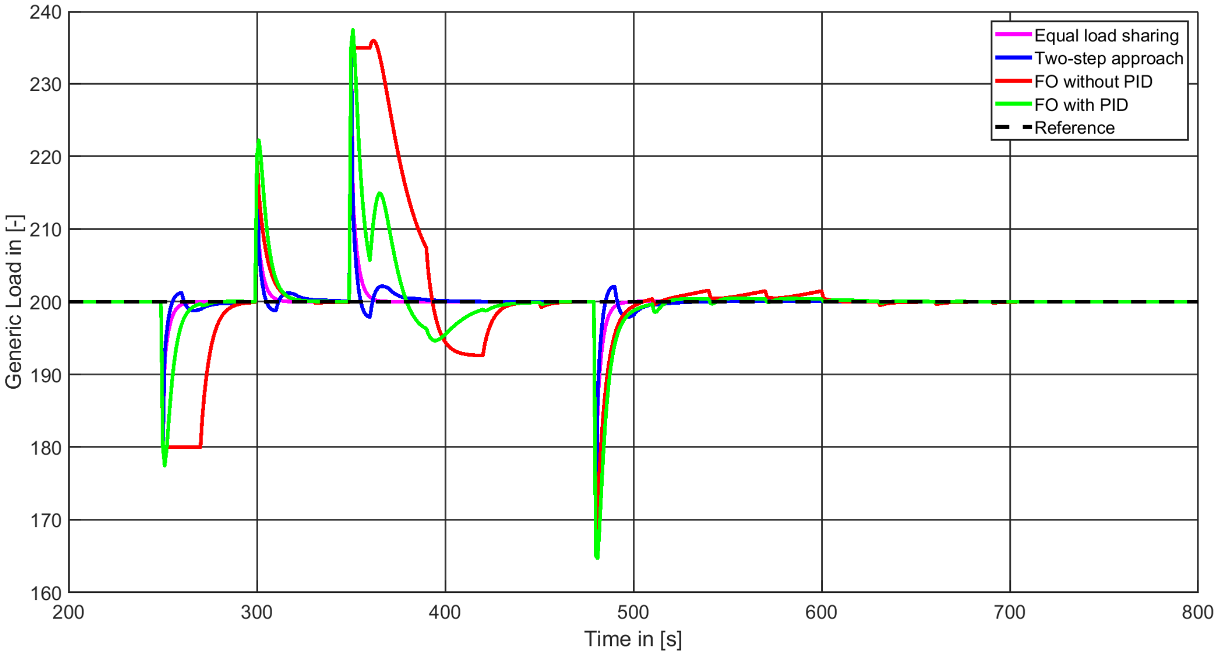


Figure 4.7: System output for the disturbance signals from Figure 4.6

controller causes a large increase in power consumption. Except from this reference change, the two-step approach causes only small visible spikes in the power consumption.

To compare the convergence behavior of the three optimizing controllers quantitatively, the settling time t_s of the power consumption and a possible static difference d_s to the optimum are analyzed for the ten changes in the station reference presented above. Here, the settling time expresses the time it takes for the power consumption to stay within ±1 % of its final value. The mean values and the corresponding standard deviation can be found in Table 4.2.

Controller	$\frac{t_s}{\kappa}$ $[s/\frac{kg}{s}]$ mean	$\frac{t_s}{\kappa}$ $[s/\frac{kg}{s}]$ std. deviation	d_s $[W]$ mean	d_s $[W]$ std. deviation
Two-step approach	0.5	0.2	10.2	9.7
Feedback optimization	1.0	0.8	-0.4	1.71
Feedback optimization with CTRT	1.2	0.7	-0.4	1.7

Table 4.2: Parameters characterizing the convergence behavior of different systems

Overall, the results show that FO has higher settling times than the two-step approach. However, in most cases the two FO-based controllers converge similarly fast and the station power consumption plot in Figure 4.8 shows that the higher settling times are mainly due to a few reference changes. This is also evident from the standard deviation. The high values indicate, that the settling times are spread for the FO-based controllers.

The static difference to the optimum shows that the two-step approach does not reliably find the inputs that correspond to a locally minimal power consumption. The FO controllers achieve an unexpected result for the chosen reference: They are able to find a minimum that has a lower power consumption than the computed numerical optimum. In Figure 4.8, this can be seen for the time interval $2160\,s - 2700\,s$. This leads to the negative static difference mean value for the two implementations of feedback optimization. Constraints are not violated by FO and the

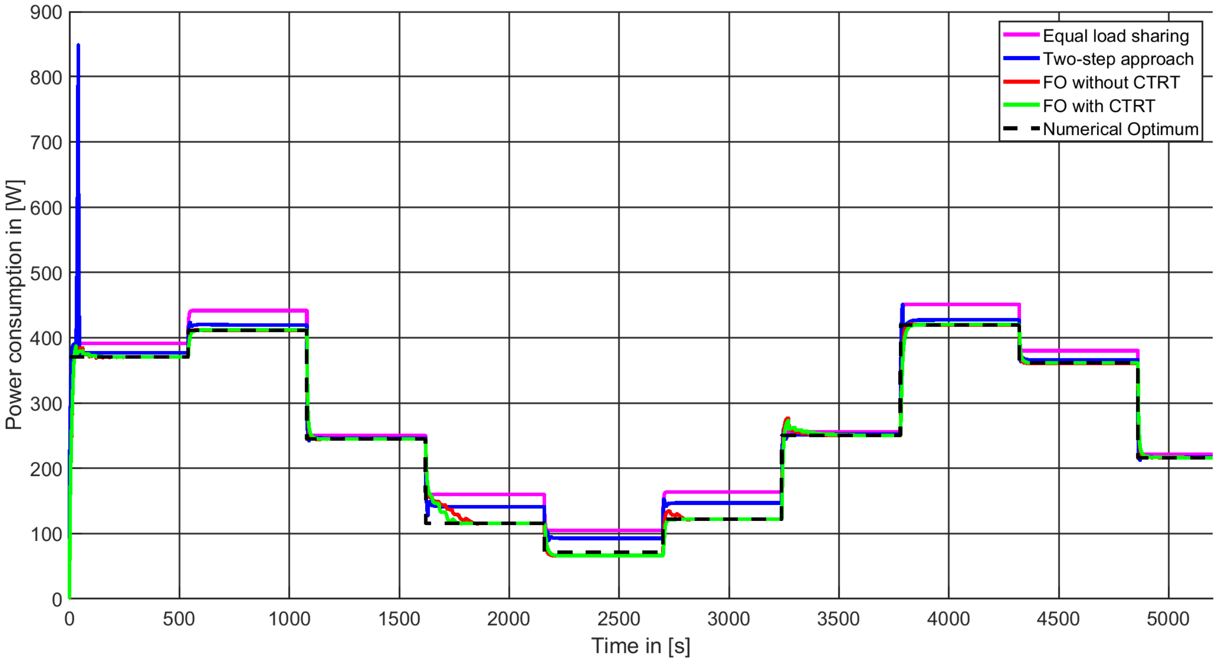


Figure 4.8: Power consumption for different controllers, no plant-model mismatch is present for the FO-based controllers

numerical optimizer jumps to the solution of FO if a different initial point is used.

4.3.3 Plant-Model Mismatch and Energy Savings

There are two forms of plant-model mismatch, structural and parametric mismatch. Here, structural plant-model mismatch is intentionally introduced by using polynomials of a different order for the estimation of the efficiency curves of the three plants. The true curves can be described by polynomials of fourth order, whereas the models use third order polynomials.

The resulting power consumption graphs for the case where no structural plant-model mismatch is present in the FO-based controllers can be found in Figure 4.8, a case where mismatch is present in Figure 4.9. The total energy consumption for the different controllers and the two cases is shown in Table 4.3.

Controller	No mismatch		Structural mismatch	
	Energy consumption [Wh]	Improvement to ELS [%]	Energy consumption [Wh]	Improvement to ELS [%]
Equal load sharing (ELS)	410.11	–	410.11	–
Two-step approach	–	–	391.22	4.61
Feedback optimization	375.89	8.34	385.53	5.99
Feedback optimization with CTRT	375.44	8.45	382.11	6.83
Numerical optimum	375.33	8.48	375.33	8.48

Table 4.3: Energy consumption for different controllers, with and without plant-model mismatch

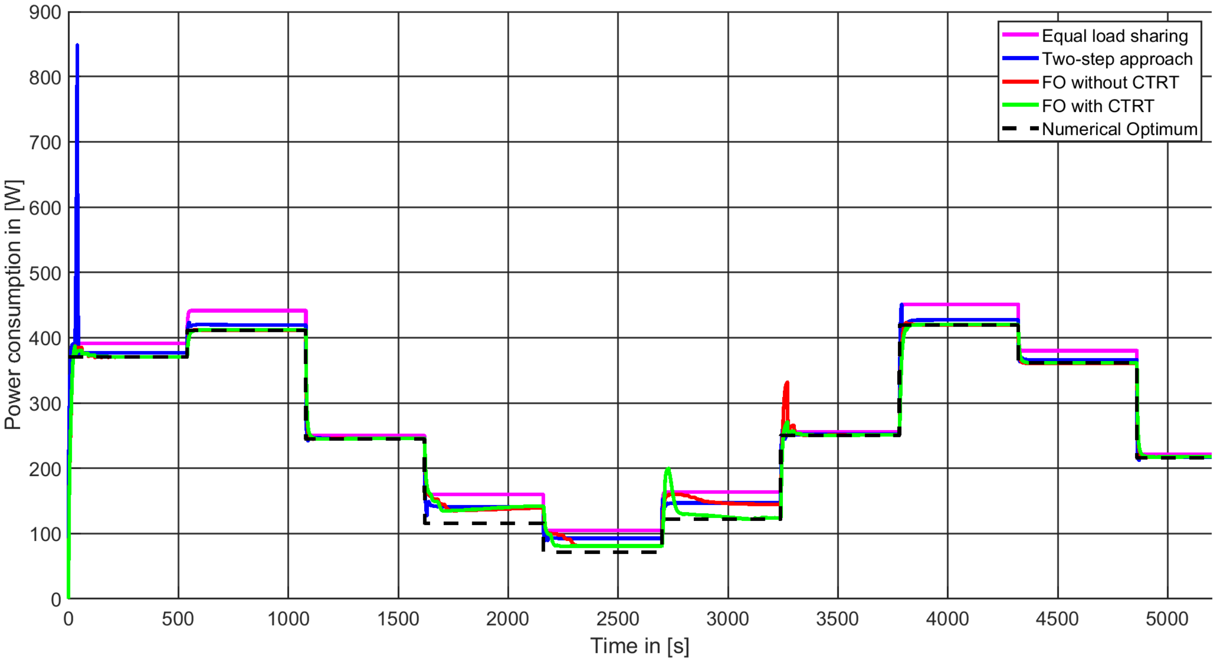


Figure 4.9: Power consumption for different controllers, plant-model mismatch is present in the FO-based controllers

When comparing the energy consumption of the different systems, it is apparent that the FO-based methods achieve better results if the gradient of the objective function is correct and no plant-model mismatch is present. With plant-model mismatch, the energy savings are still greater than for the two-step approach. The system outputs of the FO controllers show that the convergence to a solution is slower if plant-model mismatch is present. There are also two spikes in the power consumption that are not present in the case of no plant-model mismatch. The responses to the reference changes at $2160\,s$ and $2700\,s$ show, that FO with CTRT converges faster to a solution than the plain FO controller if plant-model mismatch is present.

4.4 Discussion

As expected, all controllers achieve closed-loop stability. The reference tracking behavior of the different controllers is similar, but the two-step approach is more aggressive. One of the important parameters for load sharing is the settling time. It is considerably higher for all optimizing controllers than for the equal load sharing approach. However, when only looking at the optimizing controllers, FO without CTRT produces the best result. Noteworthy is the smooth trajectory of the two FO implementations.

Disturbances are rejected by the FO implementations and show that the reference tracking constraint works well in conjunction with the measured loads l_i. However, the FO controller without a PID can have a sampling delay when reacting to disturbances. The controller only applies new inputs after the sampling time elapsed, because it does not contain a continuous time block like the FO implementation with the parallel PID. The sampling delay can be reduced by choosing a lower sampling time. Here, it is not possible because it leads to problems with the timescale separation. Therefore, the controllers that contain continuous time elements have an advantage regarding the reaction to disturbances.

Even though there is one reference (from $1620\,s$ to $2160\,s$ in Figure 4.9) where the two FO implementations are not capable of finding a better solution than the two-step approach, they exhibit some robustness to plant-model mismatch, because they can still calculate inputs that lead to a significant decrease of the station's energy consumption. The plant-model mismatch leads to slower convergence rates which increases the total energy consumption. The influence of the slower convergence rates on the total energy consumption can be reduced by using reference signals that are constant for longer periods of time. If an objective function was used that is structured such that measurements of the power consumption are present in the gradient, the two FO-based controllers might be able to find better inputs when plant-model mismatch is present.

Chapter 5

Compressor Load Sharing Problem

5.1 Dynamics of Compressors

This section covers the modeling of compressors and their interconnection in parallel configurations. The mathematical description of multi-compressor configurations is based on the model of one simplified compressor, like the one shown in Figure 5.1. This model consists of the most important input-output dynamics and is derived by using physical first principles. The model presented below is a slightly adapted version of the ones presented in [9], [10], and was originally derived in [5].

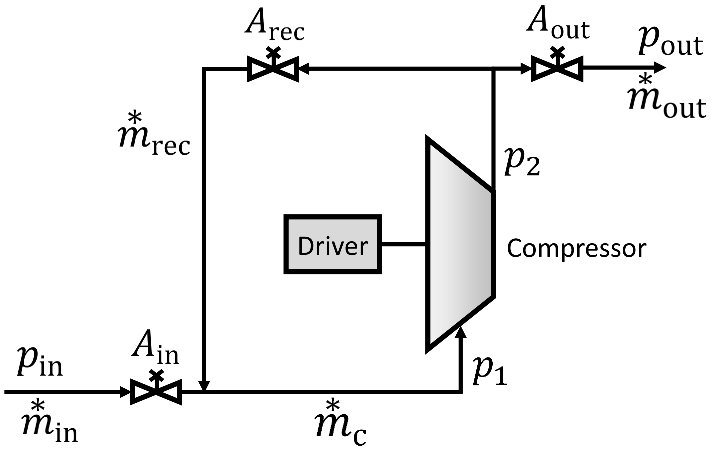

A_{rec}
A_{out}
p_{out}
$\overset{*}{m}_{\text{out}}$
$\overset{*}{m}_{\text{rec}}$
p_2
Driver
Compressor
p_{in}
A_{in}
p_1
$\overset{*}{m}_{\text{in}}$
$\overset{*}{m}_{\text{c}}$

Figure 5.1: Diagram of one compressor with surrounding tubing and valves

The inlet, outlet and recycle parts of the compressor are denoted by 'in', 'out', and 'rec'. Consequently, $\overset{*}{m}_{\text{in}}$, $\overset{*}{m}_{\text{out}}$, $\overset{*}{m}_{\text{rec}}$, and $\overset{*}{m}_{\text{c}}$ are the inlet, outlet, recycle and compressor mass flows, and p_{in}, p_{out}, p_1, and p_2 represent the inlet pressure, outlet pressure, and the compressor in- and outlet pressures.

Mass only flows between two volumes if they have different energy states. In this context, it only flows if a pressure gradient is present. The influence of the three valves on the respective mass flow is incorporated by the valve gains $k_{\text{in}}, k_{\text{out}}, k_{\text{rec}}$ (taking values between 0 and 1) and the valve orifice areas $A_{\text{in}}, A_{\text{out}}, A_{\text{rec}}$. The mass flow can be obtained when assuming incompressible flow (constant density ρ) and using Bernoulli's law. The kinetic energy $\frac{1}{2}\rho v^2$ (where v is the velocity of the gas) can be neglected compared to the pressure energy term (e.g., $p_2 - p_1$). The mass flows are positive (or zero in the case of the recycle flow) for stable compressor operation.

$$\overset{*}{m}_{\text{in}} = k_{\text{in}} A_{\text{in}} \sqrt{|p_{\text{in}} - p_1|} \tag{5.1a}$$

$$\overset{*}{m}_{\text{out}} = k_{\text{out}} A_{\text{out}} \sqrt{|p_2 - p_{\text{out}}|} \tag{5.1b}$$

$$\overset{*}{m}_{\text{rec,ss}} = k_{\text{rec}} A_{\text{rec}} \sqrt{|p_2 - p_1|} \tag{5.1c}$$

The compressor dynamics are captured by a system of coupled differential equations:

$$\frac{dp_1}{dt} = K_1 \left(\overset{*}{m}_{\text{rec}} + \overset{*}{m}_{\text{in}} - \overset{*}{m}_{\text{c}} \right) \tag{5.2a}$$

$$\frac{dp_2}{dt} = K_2 \left(\overset{*}{m}_{\text{c}} - \overset{*}{m}_{\text{rec}} - \overset{*}{m}_{\text{out}} \right) \tag{5.2b}$$

$$\frac{d\overset{*}{m}_c}{dt} = K_3 \left(p_1 \Pi - p_2 \right) \tag{5.2c}$$

$$\frac{d\overset{*}{m}_{\text{rec}}}{dt} = K_4 \left(\overset{*}{m}_{\text{rec,ss}} - \overset{*}{m}_{\text{rec}} \right) \tag{5.2d}$$

$$\frac{d\omega}{dt} = K_5 \left(\tau_{\text{ext}} - \tau_{\text{comp}} \right) \tag{5.2e}$$

The torque *absorbed by* the compressor is $\tau_{\text{comp}} = \sigma r \omega \overset{*}{m}_c$ with σ as the slip factor, r as the radius of the shaft, and ω as the rotational velocity. The external torque *applied to* the compressor is denoted by τ_{ext}, and various constants by K_i $(i = 1, ..., 5)$. Some dynamics of the recycle flow $\overset{*}{m}_{\text{rec}}$ are too complex to be modeled using physical first principles and are added empirically by using their steady-state flow $\overset{*}{m}_{\text{rec,ss}}$ in (5.2d).

The pressure ratio Π depends on the rotational velocity and the compressor mass flow in a nonlinear way, i.e., $\Pi = f_\Pi(\omega, \overset{*}{m}_{\text{c}})$. This relationship is called the *compressor map* and usually represented as a Π vs. $\overset{*}{m}_c$ plot, where ω is used as a parameter.

The polytropic efficiency η_p of a compressor depends on its rotational velocity, the pressure ratio, and the compressor mass flow. This is signified by $\eta_p = f_\eta(\omega, \Pi, \overset{*}{m}_c)$. This relationship is also nonlinear and often shown as contour lines in a compressor map. Here, the function f_η is also called *efficiency map.*

An example of a compressor and efficiency map can be found in Fig. 5.2. These maps are based on measurements and created for inlet conditions corresponding to a standardized reference atmosphere (e.g., inlet pressure $p_{\text{in}} = 10^5\,Pa$ and inlet temperature $T_{\text{in}} = 293.15\,K$) and for a specific gas (i.e., a specific molecular weight). If the compressor is operated at different inlet conditions, the compressor mass flow has to be corrected to use the map. In Fig. 5.2, this *corrected compressor mass flow* $\overset{*}{\mu}_c$ is used. The surge line is the line that connects the left ends of the ω-lines.

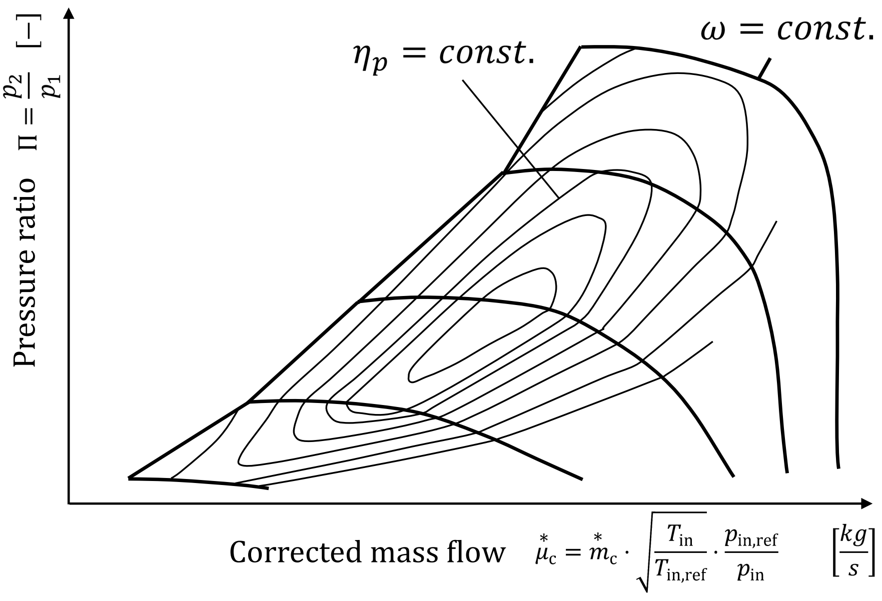
The power consumed by a compressor is equal to its shaft power P and can be calculated with the help of the polytropic efficiency, the compressor mass flow and the polytropic head y_p as

$$P = \frac{y_p}{\eta_p} \overset{*}{m}_{\text{c}} \tag{5.3}$$

$$\text{with} \quad y_p = \frac{Z_{\text{in}} R T_{\text{in}}}{M_w} \frac{n_\nu}{n_\nu - 1} \left[\Pi^{\frac{n_\nu - 1}{n_\nu}} - 1 \right], \tag{5.4}$$

where R is the universal gas constant, n_ν is the polytropic index, T_{in} is the inflow temperature, Z_{in} is the inlet compressibility factor, and M_w is the molecular weight of the gas.

Depending on the thermodynamic process (described by n_ν), different amounts of work are needed to compress gas to a certain pressure. This is captured by the polytropic head. Even



$\eta_p = const.$
$\omega = const.$
Pressure ratio $\Pi = \frac{p_2}{p_1}$ [–]
Corrected mass flow $\overset{*}{\mu}_c = \overset{*}{m}_c \cdot \sqrt{\frac{T_{in}}{T_{in,ref}}} \cdot \frac{p_{in,ref}}{p_{in}}$ $\left[\frac{kg}{s}\right]$

Figure 5.2: Schematic compressor and efficiency map

though gas properties appear in (5.4), it is mainly influenced by the pressure ratio because the gas properties are changing slowly or only when a different gas is used. Therefore, they are treated as parameters of the function $y_p = f_y(\Pi)$.

5.2 Formulation of the Constrained Optimization Problem

5.2.1 Operational Constraints

To ensure stable and safe compressor operation, several constraints have to be satisfied. The mathematical formulation of surge, choke, and mechanical constraints is adopted from [10].

$$s_{0,i} - s_{1,i}\overset{*}{m}_{c,i} + \Pi_i \leq 0 \tag{5.5a}$$

$$c_{0,i} + c_{1,i}\overset{*}{m}_{c,i} - \Pi_i \leq 0 \tag{5.5b}$$

$$\tau_{ext,i} \cdot \omega_i = P_{ext,i} \in \left[0, P^{upper}_{ext,i}\right] \tag{5.5c}$$

$$\overset{*}{m}_{c,i} \in \left[\overset{*}{m}^{lower}_{c,i}, \overset{*}{m}^{upper}_{c,i}\right] \tag{5.5d}$$

$$A_{rec,i} \in \left[0, A^{upper}_{rec,i}\right] \tag{5.5e}$$

$$\sum_{i \in \mathcal{N}} \overset{*}{m}_{out,i} \approx r(t) \tag{5.5f}$$

In these equations, i is the compressor index with $i \in \mathcal{N} = \{1, ..., n\}$ and n is the number of compressors; $s_{0,i}$, $s_{1,i}$, $c_{0,i}$ and $c_{1,i}$ are positive constants. A violation of the first three constraints can lead to serious damages [10], the others are mainly placed to emphasize physical boundaries or a control task:

- Inequality (5.5a) represents the *surge constraint.* If the flow through the compressor is too low for a certain pressure ratio, a surge occurs. It causes reverse flow in the compressor, heavy vibrations, and mechanical damage. Eventually, a compressor surge could lead to machine failure.

- The *choke constraint* is formulated in (5.5b). It is equivalent to the maximum flow through the compressor for a given rotational velocity or pressure ratio. It depends on the aerodynamic properties of the compressor and the piping on the outlet side.

- The constraint on the applied power (5.5c) is placed to ensure that no over-speeding happens. Over-speeding can lead to damaged compressor blades due to increased mechanical stress.

- *Mechanical limits* can be found in (5.5d) and (5.5e) and apply to compressor mass flow and recycle valve orifice area respectively.

- The *reference tracking constraint* (5.5f) emphasizes that one task of the compressor station's control structure is to follow a station reference.

For the formulation of the optimization problem given to the optimizing controllers, not all of the mentioned constraints are relevant. The surge, choke and over-speeding constraints are usually handled by specialized low-level controllers. The constraint on the actuation limit in (5.5e) can be important for the implementation of the anti-surge controllers as the saturation could lead to integrator windup problems.

5.2.2 Influences of Parallel Configurations

When compressors are arranged in multi-machine configurations, interconnection constraints can arise and influence the choice of the decision variable for the optimization problem. In the case of a parallel configuration, one-way valves are used to prohibit unwanted reverse flows that can occur if the machines produce different compressor outlet pressures. Therefore, constraints on the inflow and outflow pressures are not necessary.

The recycle valve orifice area $A_{\mathrm{rec},i}$ and the applied torque $\tau_{\mathrm{ext},i}$ can be controlled and used as inputs for parallel configurations. p_{in} and p_{out} are not available as degrees of freedom, because they are deterentermined by the dispatch system at a higher level of the control hierarchy. $\overset{*}{m}_{\mathrm{c},i}$, $\overset{*}{m}_{\mathrm{out},i}$, P_i, Π_i, η_i, and ω_i are quantities that can be measured as outputs of the parallel configuration.

5.2.3 Influences of Low-Level Controllers

In the following implementation of the optimizing controllers, low-level flow controllers are used. These calculate the external torque $\tau_{\mathrm{ext},i}$ based on a mass flow reference $\overset{*}{m}^{\mathrm{ref}}_{\mathrm{c},i}$ and the actual compressor mass flow $\overset{*}{m}_{\mathrm{c},i}$. In steady state, the compressor mass flow reference is equal to the actual compressor mass flow. From the standpoint of the optimizing controllers, this compressor mass flow reference serves as a control input and constraints on the inputs can be formulated as constraints on the compressor mass flow references. All controllers require the actual compressor mass flows as feedback. Additionally, the two-step approach needs the compressor efficiencies and their pressure ratio (i.e., $y_i = [\overset{*}{m}_{\mathrm{c},i}, \Pi_i, \eta_i,]$), while feedback optimization uses the pressure ratios and the individual power consumptions as well (i.e., $y_i = [\overset{*}{m}_{\mathrm{c},i}, \Pi_i, P_i]$).

A closed recycle valve is beneficial for the minimization of the compressor's energy consumption. However, some operating points are only reachable when recycling some mass flow. In the following, only operating points that can be reached without any recycle flow and that are far away from the surge line will be considered. Additionally, the increase and especially the decrease of the applied torque $\tau_{\mathrm{ext},i}$ has to be sufficiently smooth, because the anti-surge controller also acts if these changes happen faster than a predetermined rate limit. This is enforced by limiting the slope of the flow controllers' output. With these restrictions in place, the optimizing controllers do not use the recycle valve as an input, but only the compressor mass flow references. To show that the optimizing controllers can achieve stable operation that does not require any action by the anti-surge controllers, they are still active and would actuate the recycle valve if necessary. With a permanently closed recycle valve and without disturbances, the compressor mass flow is the same as the compressor outlet flow.

5.2.4 Constrained Optimization Problem

After the previous considerations and by using (5.3), the steady-state constrained optimization problem can be defined as

$$\Phi(\overset{*}{m}_{\text{c},1}, \overset{*}{m}_{\text{c},2}, \overset{*}{m}_{\text{c},3}) = \sum_{i=0}^{3} P_i + \frac{1}{2}\sum_{i=0}^{3}(P_i)^2 \quad \text{with} \quad P_i = \frac{y_{\text{p},i}}{\eta_{\text{p},i}}\overset{*}{m}_{\text{c},i} \tag{5.6a}$$

$$\text{subject to} \quad \text{steady-state equations derived from (5.2)} \tag{5.6b}$$

$$\sum_{i=0}^{3} \overset{*}{m}_{\text{c},i} \approx r(t) \tag{5.6c}$$

$$\overset{*}{m}_{\text{c},i} \in \left[\overset{*}{m}_{\text{c},i}^{\text{lower}}, \overset{*}{m}_{\text{c},i}^{\text{upper}}\right] \tag{5.6d}$$

where $r(t)$ is the station mass flow reference. In contrast to the generic load sharing problem, the objective function is extended with quadratic terms. This is done to improve the numerical behavior of the optimizing controllers.
The box constraint on the compressor mass flow $\overset{*}{m}_{\text{c},i}$ is chosen such that the anti-surge controller does not have to act due to a too low mass flow:

$$\overset{*}{m}_{\text{c},1}^{\text{lower}} = \overset{*}{m}_{\text{c},2}^{\text{lower}} = \overset{*}{m}_{\text{c},3}^{\text{lower}} = 66\,\frac{kg}{s}$$
$$\overset{*}{m}_{\text{c},1}^{\text{upper}} = \overset{*}{m}_{\text{c},2}^{\text{upper}} = \overset{*}{m}_{\text{c},3}^{\text{upper}} = 120\,\frac{kg}{s}$$

5.3 Implementation

In the following subsections, the implementation of the compressor model and the different controllers will be discussed. After presenting the system's structure and the implementation of the efficiency maps, the equal load sharing controller, the two-step based controller, and two implementations of FO are described in detail. One FO implementation follows the idea of [6] more closely and incorporates reference tracking into the optimization problem, the other implementation uses a PID controller for continuous time reference tracking.

5.3.1 System Structure and Low-Level Controllers

The model presented in Section 5.1 is implemented in an s-function for each compressor. The implementation of the efficiency maps is discussed in the following subsection in detail. If the inflow and outflow valve orifice areas are subject to changes during the simulation, a property of the s-function has to be set such that MATLAB might detect an algebraic loop even if none is actually present[1]. In this case, a unit delay after the s-function greatly improves the simulation

[1] If the output function `mdlOutputs` of the s-function uses the inputs to compute the outputs at the same time step, the "direct feedthrough" property has to be set to true. This causes MATLAB to assume that all outputs are depending on the inputs. When looking for algebraic loops, it does not verify whether the signal in question actually depends directly on the inputs.

speed. Here, the valve orifice areas and the inflow temperature are constant throughout the simulation and the unit delay is not needed. Measurements that are needed by the optimizing controllers as feedback are passed to them by using Simulink's "GoTo"-block. Thus, these signals are not visible in any of the following block diagrams.
One layer above the model are the low-level flow and anti-surge controllers, as shown in Figure 5.3. The anti-surge controller compares the current compressor mass flow to a minimum flow which is based on the current pressure ratio. If the compressor mass flow is too low, the recycle valve is opened to increase the flow through the compressor and thereby prevent a surge. Details on the implementation of the anti-surge controller can be found in Appendix A. Ramp constraints on the flow controllers' output are used to prevent fast changes of the applied torque $\tau_{\text{ext},i}$ and thereby compressor surge. This ramp constraint can be interpreted as the dynamics of the driver that applies a torque to the compressor (e.g., a variable frequency drive).
The top layer consists of the station reference signal, the (optimizing) station controller, and the three plants. Disturbance signals can be given to the individual compressors directly and act on the model level, as shown in Appendix A.

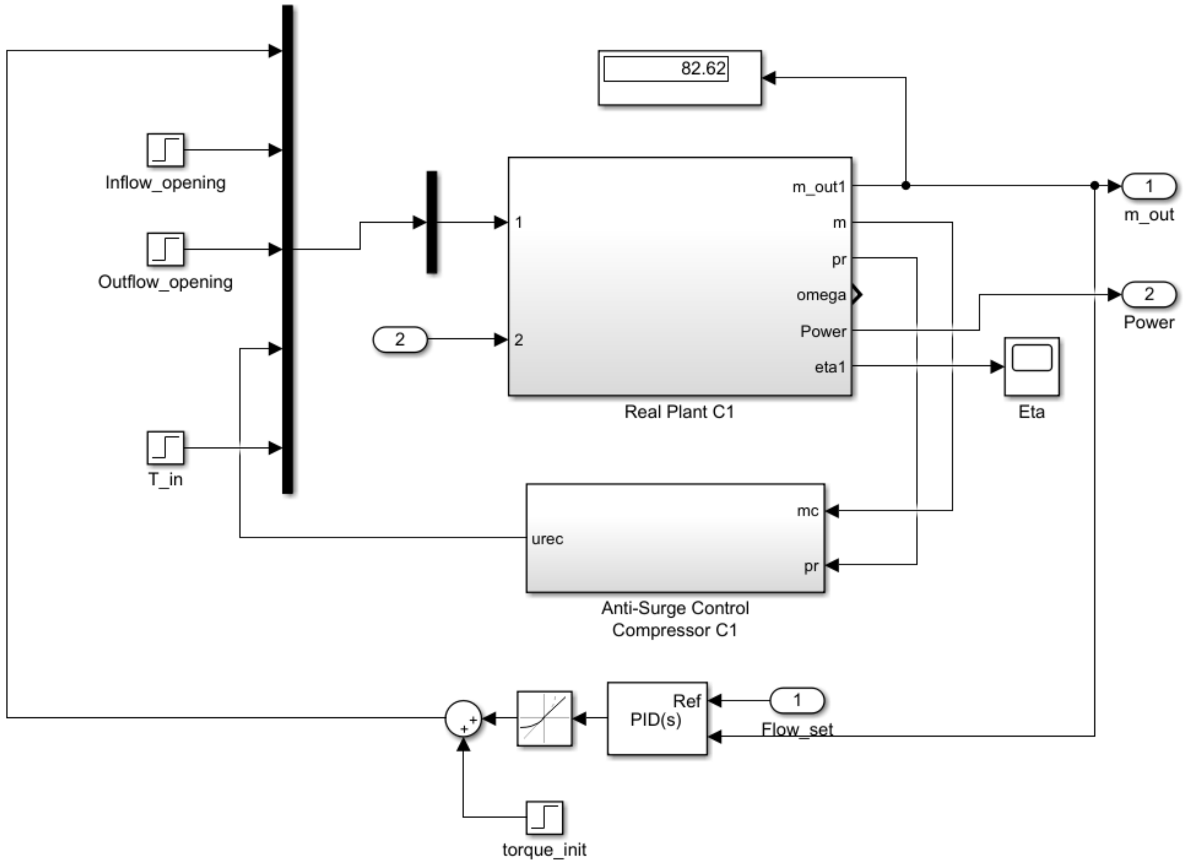


Figure 5.3: Block diagram of one compressor and its low-level controllers that are responsible for reference tracking and anti-surge control

5.3.2 True and Estimated Efficiency Maps

In realistic field conditions the true compressor map is unknown but compressor power consumption and compressor efficiency can be measured. In our simulation study we use Gaussian process regression (GPR) models, which are fitted to existing compressor maps to serve as the data generator for the compressor power and compressor efficiency, mimicking the behavior in

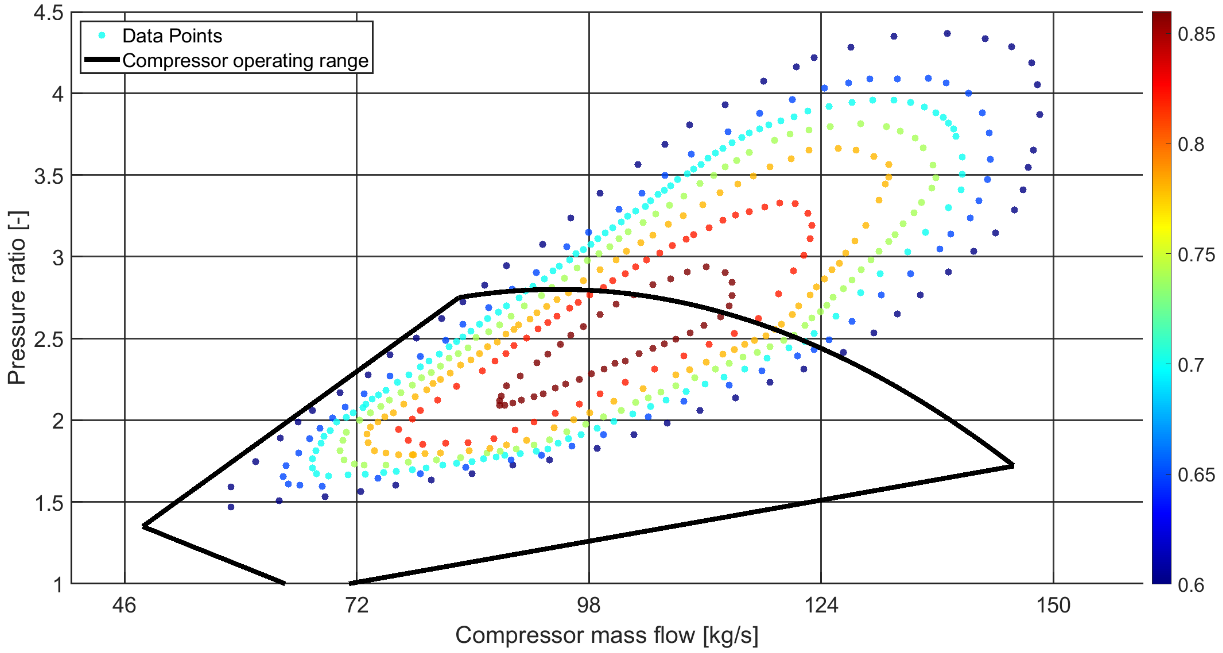


Figure 5.4: Original compressor and efficiency map. The different colors indicate different efficiencies.

actual compressors. These GPR models are only used for simulating the compressors and are not revealed to the optimization algorithms.

The three compressor and efficiency maps are variations of the one depicted in Figure 5.4. Twenty-five points of each map are chosen randomly and used by GPR to identify a model. This GPR model is used as the true efficiency map and can be seen in Figure 5.9.

The same twenty-five data points are used to fit a polynomial of the following structure with the help of linear least squares. Each polynomial represents the estimated efficiency map of one compressor.

$$\eta_{\mathrm{p},i} = f^{\eta}_{c,i}(\overset{*}{m}_{\mathrm{c},i}, \Pi_i) = \alpha + \beta \cdot \overset{*}{m}_{\mathrm{c},i} + \gamma \cdot \Pi_i + \delta \cdot \overset{*}{m}_{\mathrm{c},i}\Pi_i + \varepsilon \cdot (\overset{*}{m}_{\mathrm{c},i})^2 + \lambda \cdot (\Pi_i)^2 \tag{5.7}$$

The index c represents the set of coefficients $c = \{\alpha, \beta, \gamma, \delta, \varepsilon, \lambda\}$. The resulting estimated efficiency maps can be seen in Figure 5.10. These maps are available to each controller and ensure that no controller has an advantage by using a better model from the beginning of the simulation.

5.3.3 Equal Load Sharing

In this case, only a PI controller is used to track the station reference. The total network mass flow is fed back to the PI controller. One third of the calculated input is applied to each plant. The setup is visualized in Figure 5.5. A controller which applies one third of the station reference to the plants in a feed-forward manner without feedback would yield similar ramp responses. However, this would make disturbance rejection on the station level impossible.

5.3.4 Two-Step Approach

The industrial standard is implemented by using an s-function for the optimizing part. The reference tracking part is made up of a PID controller. The whole system can be seen in Figure 5.6. The objective function of the optimization problem is (5.6a). One part of it is the compressors' efficiency $\eta_{\mathrm{p},i}$ which is calculated by evaluating the estimated efficiency map at a given compressor

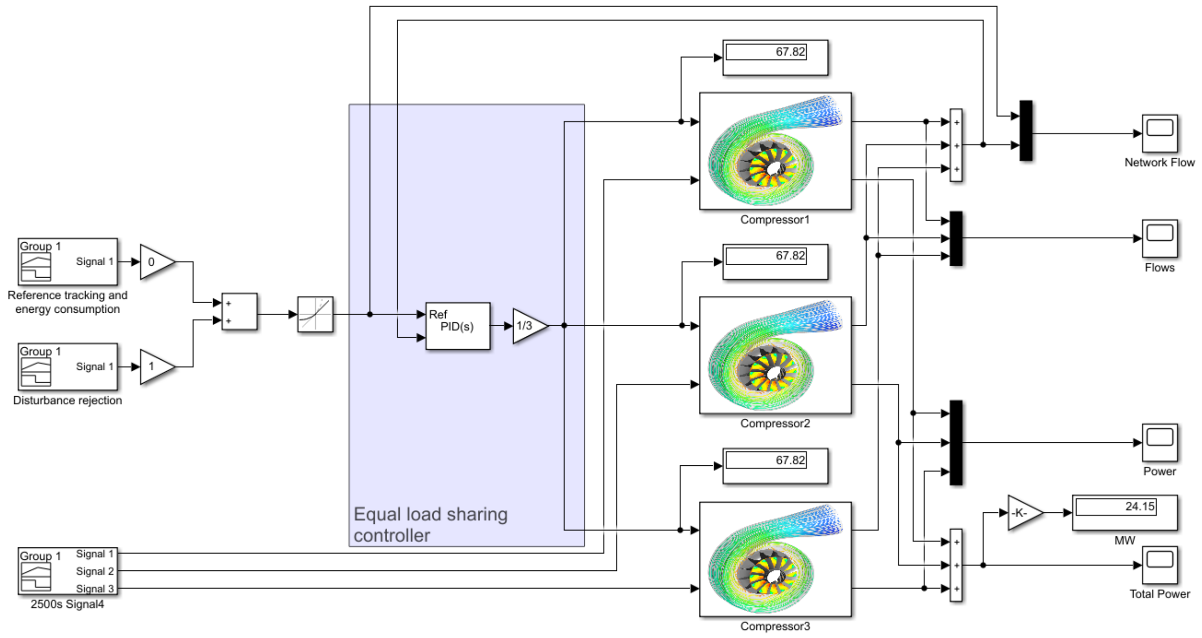


Figure 5.5: Block diagram of the three compressors and the equal load sharing controller

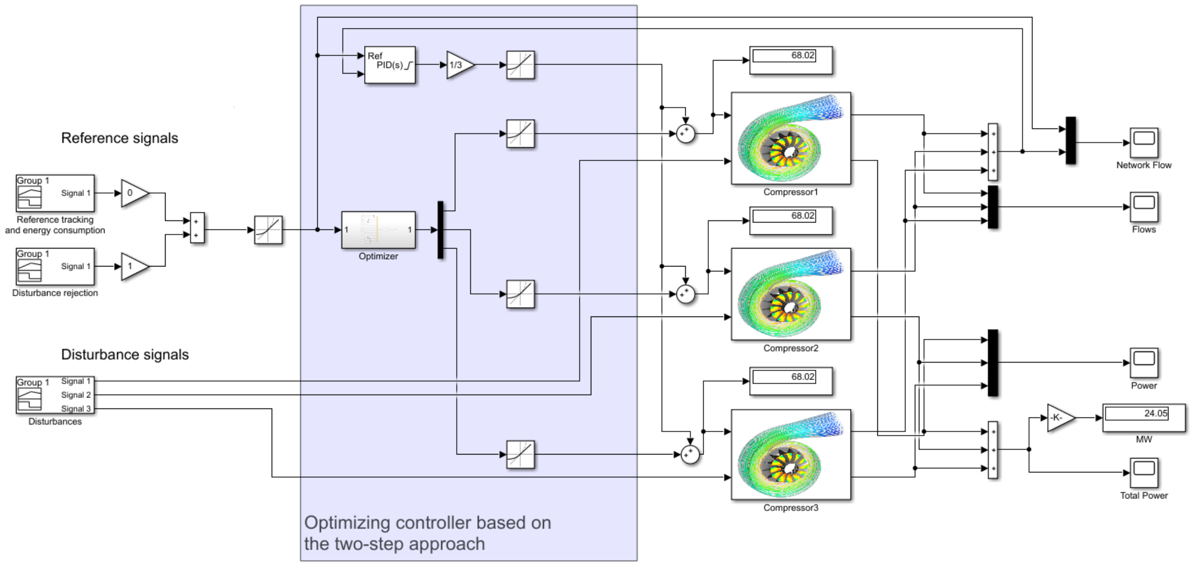


Figure 5.6: Block diagram of the three compressors and the two-step based optimizing controller

mass flow and pressure ratio. As described in Section 5.3.2, the estimated maps are represented by polynomials of the form (5.7). The parameter estimation step uses measurements of the pressure ratio, the compressor mass flow, and the efficiency in conjunction with linear least squares to find a better fit of the polynomial.

The twenty-five data points that are used to identify the true efficiency map with GPR and to obtain the estimated map are given to the two-step approach as historical data. Therefore, the polynomial obtained by the parameter estimation step at the beginning of the simulation is exactly the same as the one used by FO.

The optimization step uses MATLAB's nonlinear optimization algorithm `fmincon` to compute the solution of the constrained optimization problem, like in the generic load sharing problem (Section 4.2.2). The inputs are limited by the box constraint (5.6d). Additionally, the constraint (5.6c) is implemented as a box constraint on the inputs. The difference to the actual mass flows is then passed to the plant. This last step is necessary due to the PID controller that acts in parallel.

5.3.5 Feedback Optimization

This implementation of FO follows the idea of [6] closely and the optimizing controller has the task to track the station reference. Figure 5.7 shows a block diagram of the system.

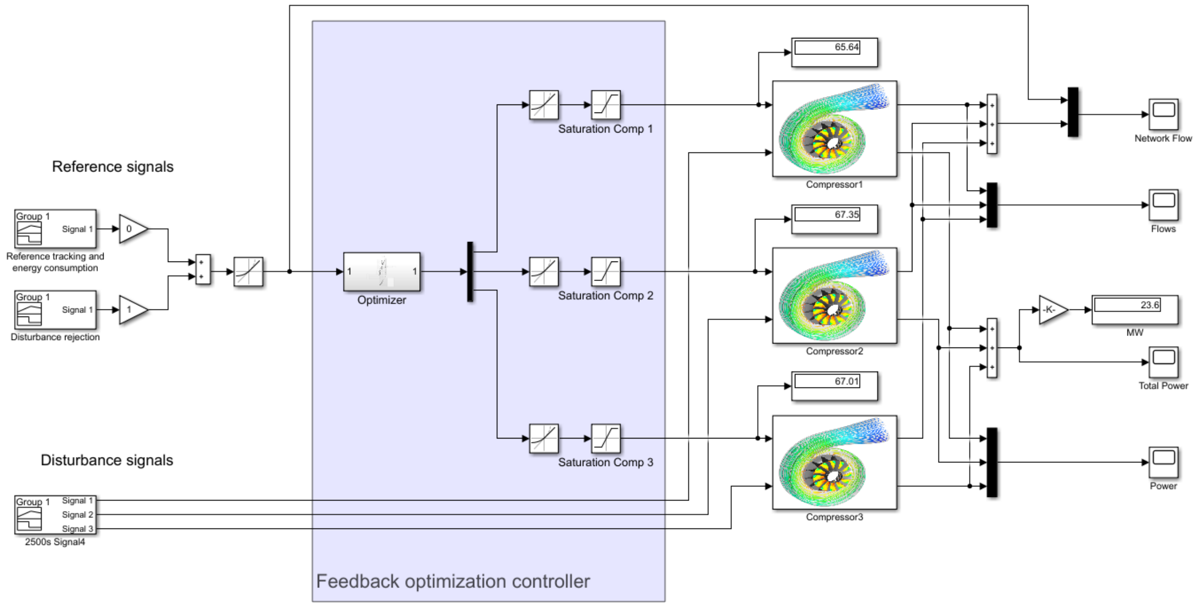


Figure 5.7: Block diagram of the three compressors and FO controller without CTRT

The objective function used by FO is (5.6a). The box constraint on the compressor mass flows (5.6d) is implemented as inequality constraints on the inputs. The reference tracking constraint from (5.6c) is rewritten as a narrow box constraint and implemented as a constraint on the outputs (again using $\sum_{i=0}^{3} y_i = \sum_{i=0}^{3} \overset{*}{m}_{\mathrm{c},i} \in [0.9999 \cdot r(t), 1.0001 \cdot r(t)]$).
The low-level flow controllers enable a similar steady-state system description as in the generic load sharing case

$$y = \begin{pmatrix} \overset{*}{m}_{\mathrm{c},1} \\ \overset{*}{m}_{\mathrm{c},2} \\ \overset{*}{m}_{\mathrm{c},3} \\ P_1 \\ P_2 \\ P_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{y_{\mathrm{p},1}}{\eta_{\mathrm{p},1}} & 0 & 0 \\ 0 & \frac{y_{\mathrm{p},2}}{\eta_{\mathrm{p},2}} & 0 \\ 0 & 0 & \frac{y_{\mathrm{p},3}}{\eta_{\mathrm{p},3}} \end{pmatrix} \cdot \begin{pmatrix} \overset{*}{m}{}_1^{\mathrm{ref}} \\ \overset{*}{m}{}_2^{\mathrm{ref}} \\ \overset{*}{m}{}_3^{\mathrm{ref}} \end{pmatrix} = F \cdot u \tag{5.8}$$

where $u_i = \overset{*}{m}{}_i^{\mathrm{ref}}$ are the inputs that are supplied to the low-level flow controller. The pressure ratios are omitted as outputs here, because there is no useful steady-state relationship available and they are only used as parameters to compute the polytropic efficiencies $\eta_{\mathrm{p},i}$. The input-output sensitivities can be calculated analytically to $\nabla h(u^k) = F$. The gradient of the objective function is calculated analytically. Both, the sensitivities and the gradient are then evaluated with the actual compressor mass flows and the pressure ratios of the current operating point.

5.3.6 Feedback Optimization with Continuous Time Reference Tracking

This implementation of FO uses a PID controller for the reference tracking task. The optimizing controller is implemented similarly to the FO controller in Section 5.3.5. A block diagram showing the high-level system structure can be found in Figure 5.8. It depicts the controller consisting of two parts, the optimizing part and the reference tracking part.
The constraints used in this implementation are slightly different to the ones used for the FO implementation without CTRT. The reference tracking is handled by the PID controller and therefore a reference tracking constraint on the outputs is not necessary. However, it is necessary that the sum of the input deviations is zero. Thus, the narrow box constraint $\sum_{i=0}^{3} \overset{*}{m}_{\text{c},i}^{\text{ref}} \in [-0.0001, 0.0001]$ is used.

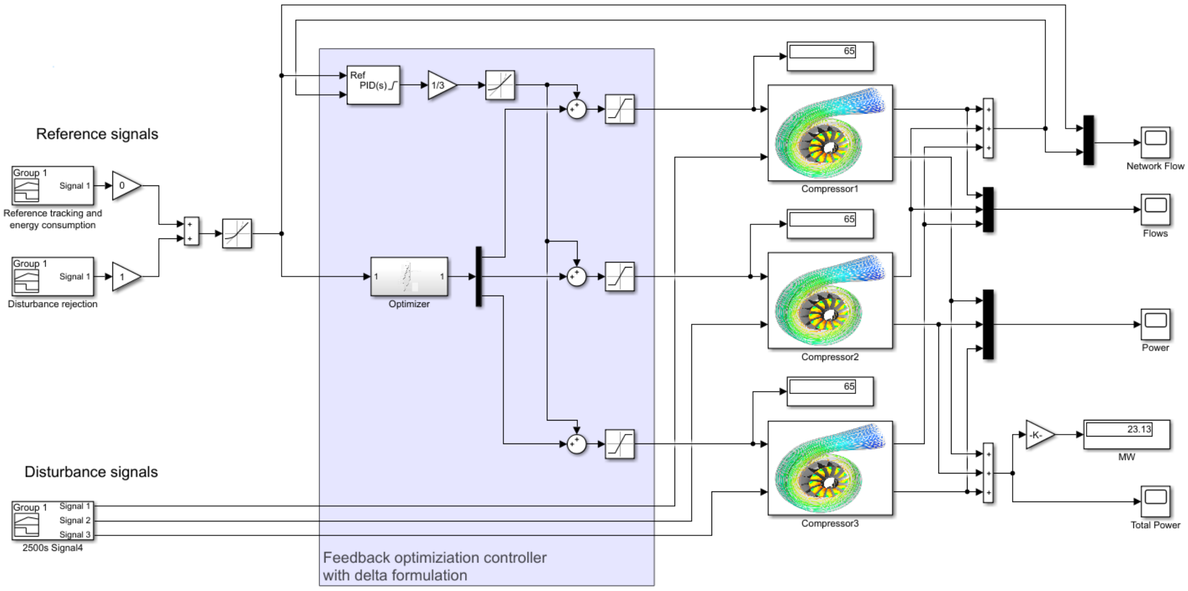
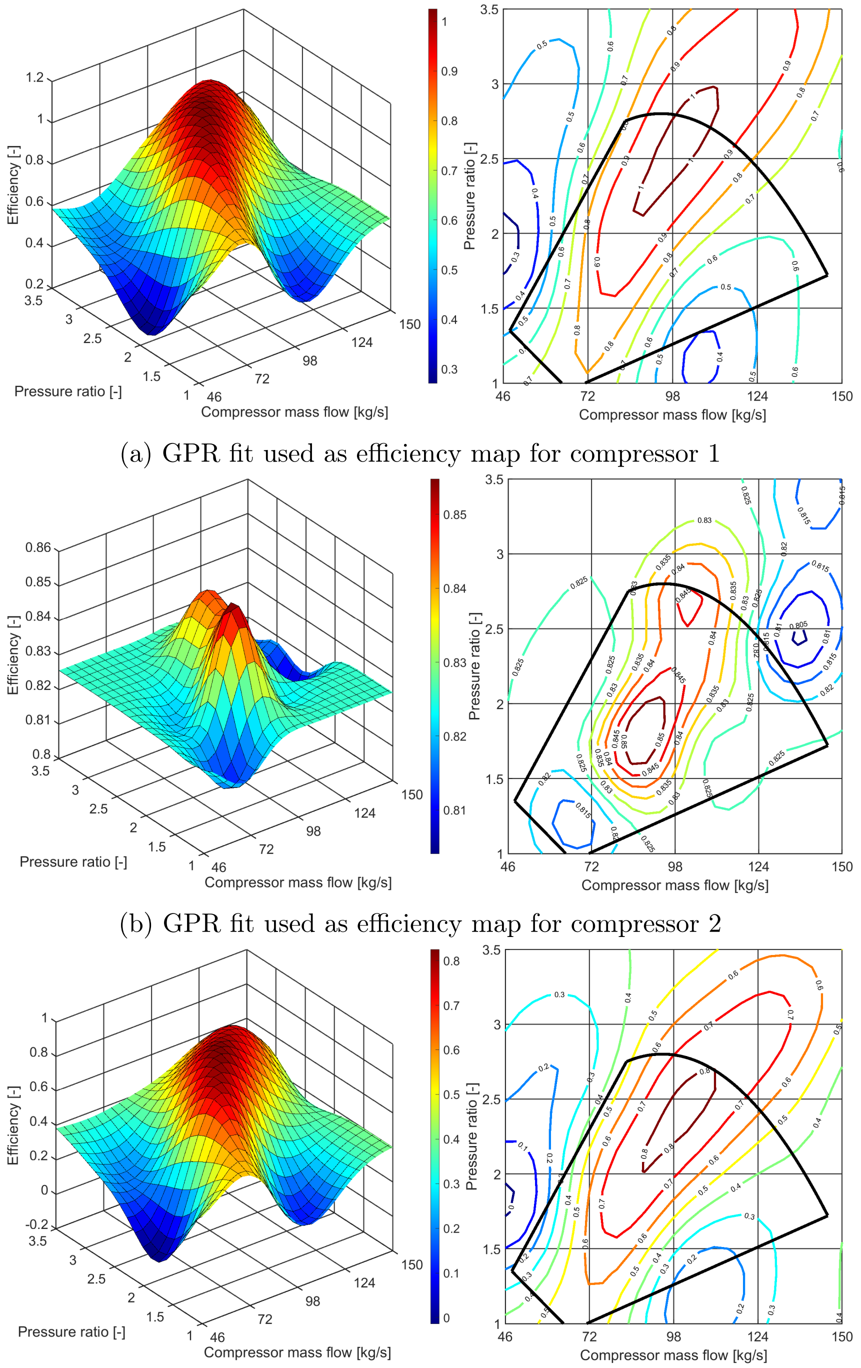


Figure 5.8: Block diagram of the three compressors and FO controller with CTRT

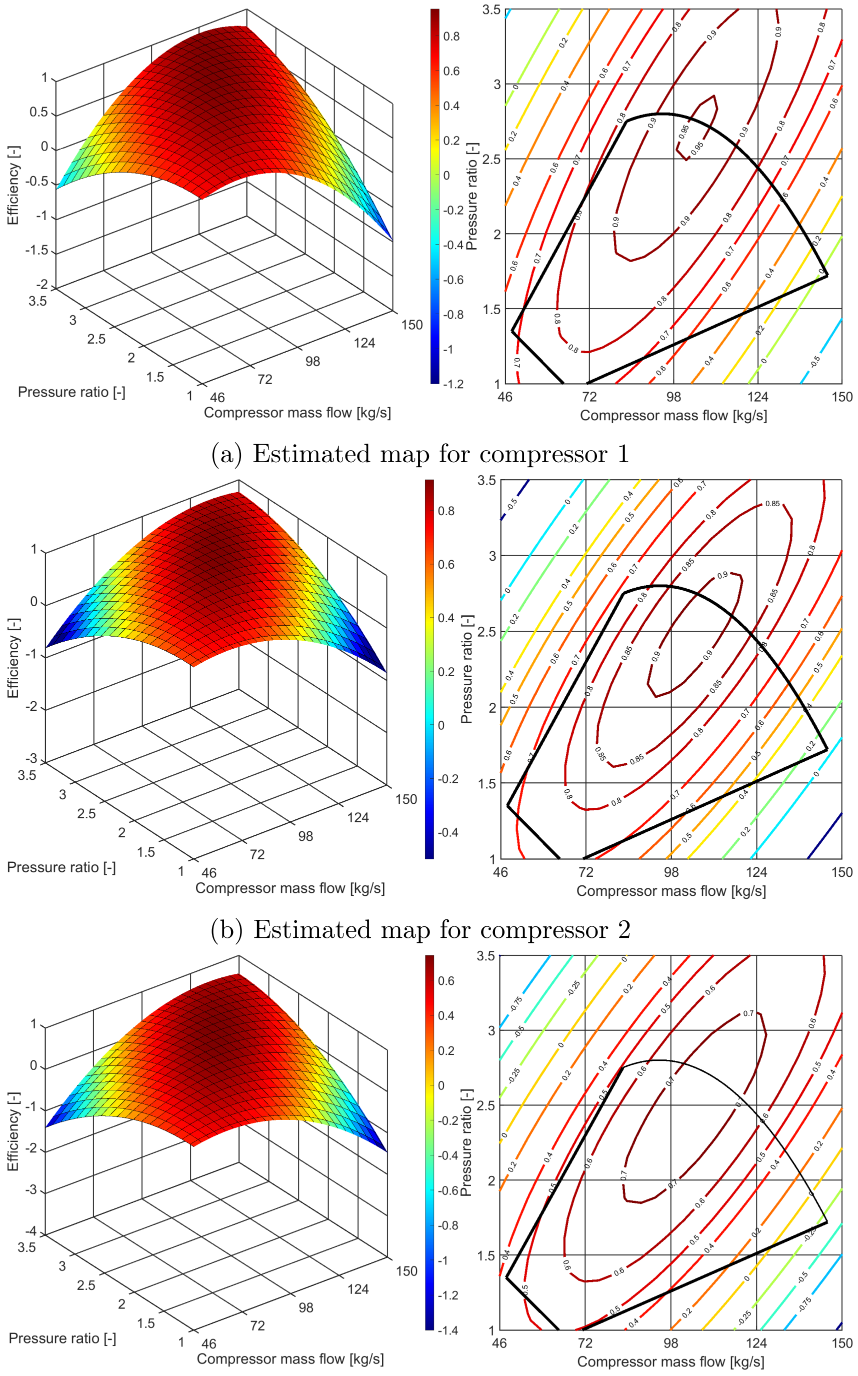


(a) GPR fit used as efficiency map for compressor 1

(b) GPR fit used as efficiency map for compressor 2

(c) GPR fit used as efficiency map for compressor 3

Figure 5.9: GPR fits used as true efficiency maps of the three compressors. The black solid lines represent the boundaries of the compressor's operating range. The line on the left is the surge line, the curved line on the top corresponds to the maximum speed line and the line on the right is the choke line.



(a) Estimated map for compressor 1

(b) Estimated map for compressor 2

(c) Estimated map for compressor 3

Figure 5.10: Polynomial functions used to estimate the true efficiency maps. The black solid lines represent the boundaries of the compressor's operating range. The line on the left is the surge line, the curved line on the top corresponds to the maximum speed line and the line on the right is the choke line.

5.4 Results and Data Analysis

5.4.1 Reference Tracking

To compare the reference tracking of the various controllers, the four parameters rise time, settling time, overshoot and peak time are reviewed. The same definitions as at the beginning of Section 4.3 are used. The term *fall time* is used for decreasing reference signals analogous to the rise time for increasing reference signals. Because of the ramp constraint on the reference signal ($\left|\frac{dr(t)}{dt}\right| \leq 1\frac{kg}{s^2}$), the definitions of these parameters are transferred to the ramp response. Two ramp responses are reviewed, an increase and a decrease. The Tables 5.1 and 5.2 contain the parameters corresponding to the system responses in Figures 5.11 and 5.12.

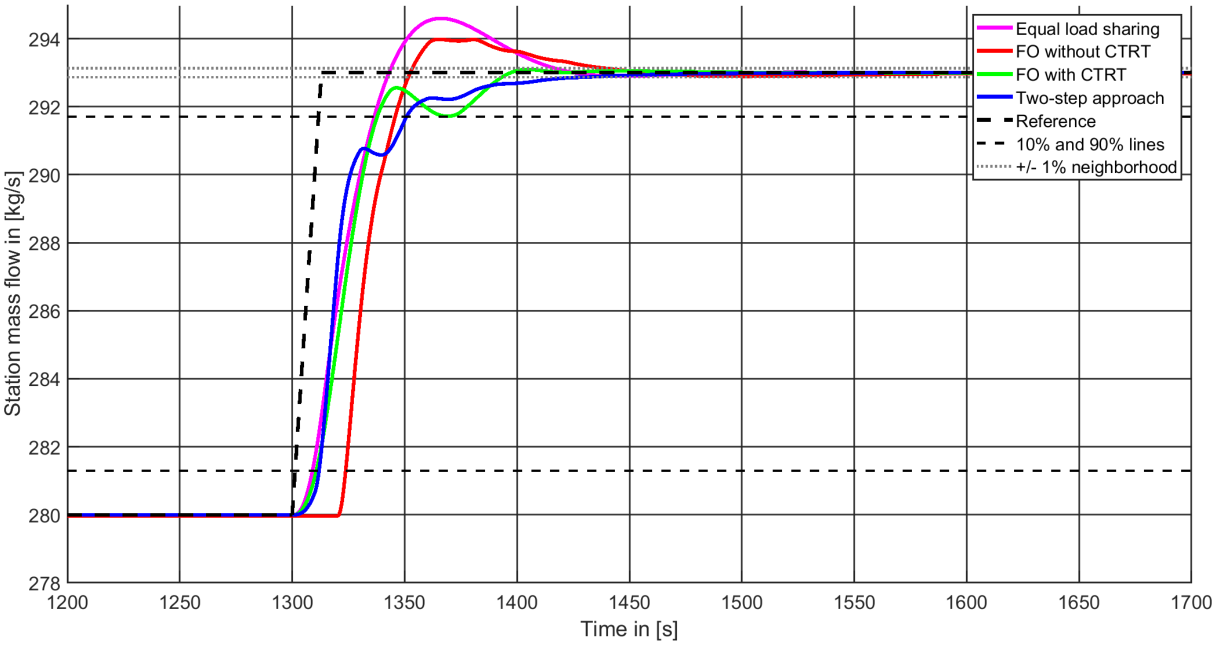


Figure 5.11: Increase in the reference signal and system responses

Controller	Rise time t_r [s]	Settling time t_s [s]	Overshoot M_p [kg/s]	Peak time t_p [s]
Equal load sharing	27.5	116.8	1.60	66.3
Two-step approach	38.9	130.8	0	0
Feedback optimization	22.0	139.1	0.98	65.25
Feedback optimization with CTRT	27.0	93.38	0.09	104.1

Table 5.1: Parameters characterizing the ramp response of the different systems for a ramp from $r(t = 1300\,s) = 280\,\frac{kg}{s}$ to $r(t = 1313\,s) = 293\,\frac{kg}{s}$.

The reference tracking behavior of all controllers is similar and does not depend on the direction of the reference change. All controllers adapt the station mass flow quickly but need time to converge to the final value. However, some controllers create significant overshoots, others tend to oscillate more. The plain FO controller and the equal load sharing controller overshoot the final reference value to which they converge with little oscillations. The FO controller with CTRT

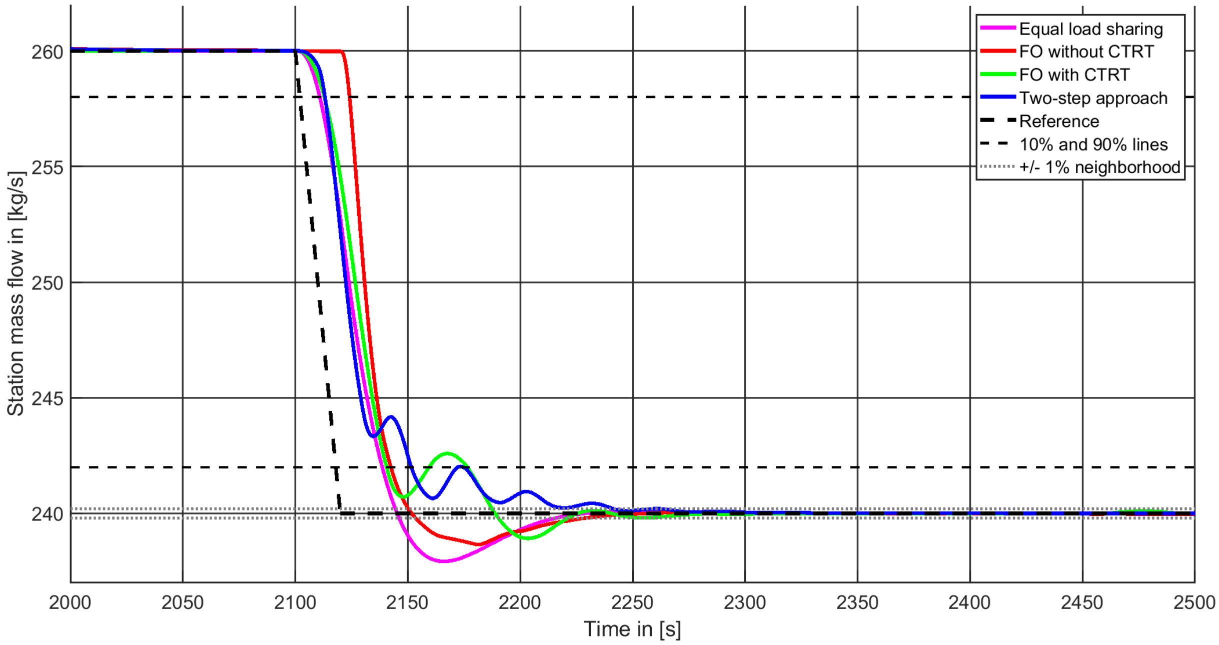


Figure 5.12: Decrease in the reference signal and system responses

Controller	Fall time t_r [s]	Settling time t_s [s]	Overshoot M_p [kg/s]	Peak time t_p [s]
Equal load sharing	27.8	115.8	-2.07	64.5
Two-step approach	38.4	162.1	0	–
Feedback optimization	18.4	121.2	-1.35	81.2
Feedback optimization with CTRT	27.63	121.2	-1.08	103.3

Table 5.2: Parameters characterizing the ramp response of the different systems for a ramp from $r(t = 2100\,s) = 260\,\frac{kg}{s}$ to $r(t = 2120\,s) = 240\,\frac{kg}{s}$.

achieves smaller overshoots and converges with some oscillations. Notable is the sampling delay of the plain FO implementation which is due to the controller running in discrete time. It cannot react to reference changes in between sampling instances.

The FO implementation without the PID and the equal load sharing controller create large overshoots in the range of 7 % to 12 % of the reference change. The peak times are lowest for the equal load sharing controller and highest for the FO implementation with the PID in parallel. The FO controller without CTRT yields the shortest rise and fall times. This is due to the sampling delay, as the FO controller without CTRT does not sample a ramp but rather steps at discrete times. For the considered reference changes, it samples at the beginning of the reference change and 20 s later again. The FO controller with CTRT does not achieve smooth outputs, but rather fast convergence with small oscillations and little overshoots. The other FO implementation produces higher overshoots but less oscillations, hinting that the inputs of the plants change steadily.

5.4.2 Disturbance Rejection

The disturbance rejection behavior of the different controllers is analyzed by using the disturbance signals from Figure 5.13. These disturbances are applied to the compressor model outputs $\overset{*}{m}_{c,i}$ and $\overset{*}{m}_{\text{out},i}$ before they are fed back to the optimizing controllers or used by PI respectively PID controllers. The signals are similar to the ones used in the previous chapter and contain steps and a ramp. The system outputs are shown in Figure 5.14.

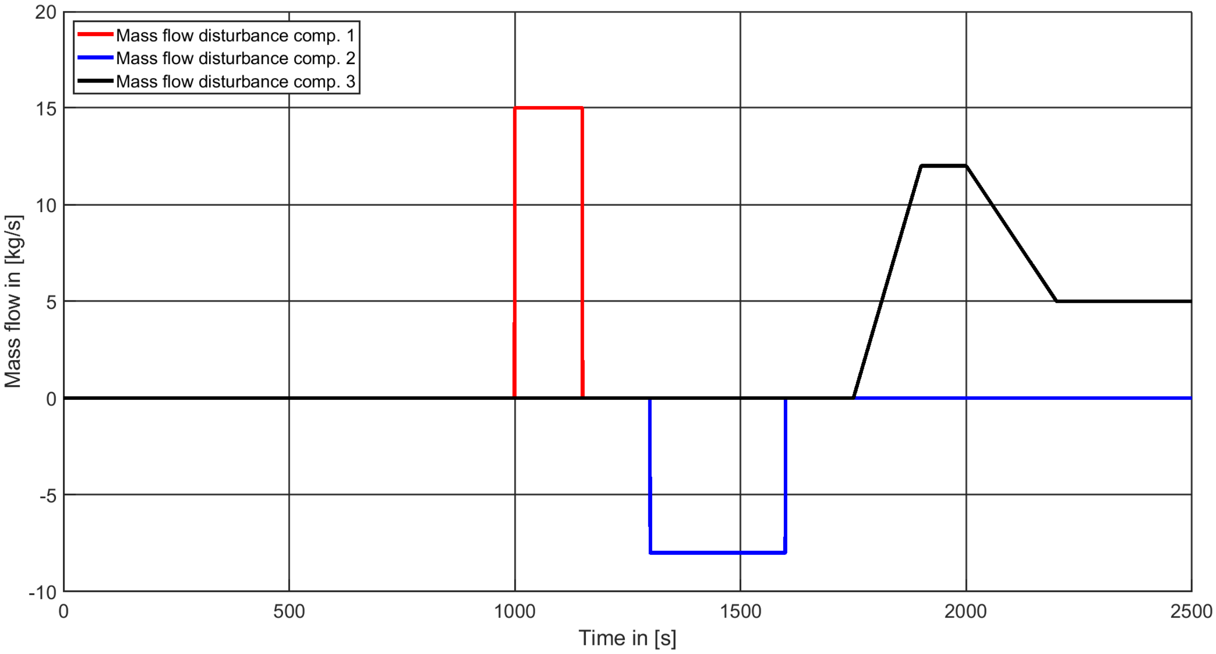


Figure 5.13: Disturbance signals which are applied to the compressors

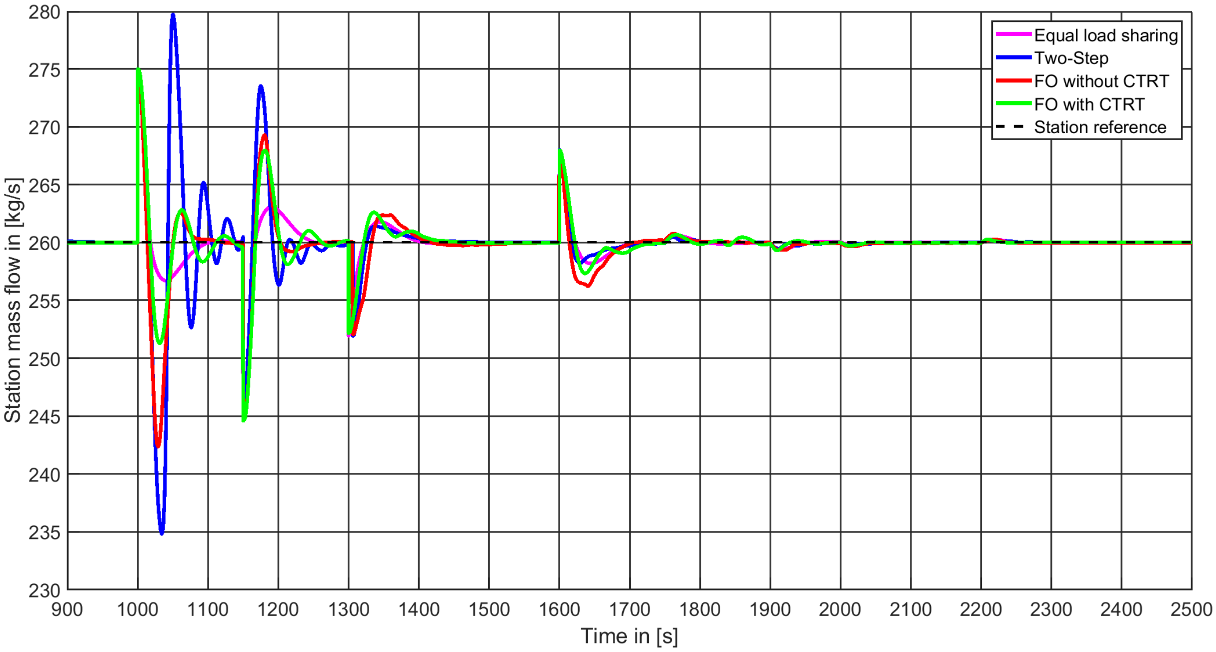


Figure 5.14: System outputs corresponding to the disturbance signal depicted in Figure 5.13

As in the generic load sharing case, the disturbance rejection behavior is only analyzed qualitatively. All controllers are able to react to the applied disturbances and to track the reference. However, the individual system responses are different.

The equal load sharing controller achieves the smallest overshoots and exhibits steady convergence to the reference. Oscillations are not present in the station mass flow. The largest overshoots and oscillations are created by the two-step based controller. The two FO-based controllers create overshoots, which are a little bit bigger for the implementation without CTRT. The FO controller with CTRT creates oscillations which are noticeable, but smaller than for the two-step approach. Similar to the equal load sharing controller, the other FO controller does not create significant oscillations. Here, no sampling delays are visible, which is due to the disturbance signal. The signal only contains reference changes that coincide with the sampling times of the controller. If this was not the case, a sampling delay would be visible in the system response of the FO controller without CTRT, like in the reference tracking plots (Figures 5.11 and 5.12). Analogous to the generic load sharing case, the ramp disturbance is handled best by all optimizing controllers. This type of disturbance signal has little effect on the station mass flow.

It is notable, that the considered disturbance signals did not cause the anti-surge controllers to react, even though the mass flow changes are abrupt. They are likely to become active in the case of larger disturbances, especially for decreases of the compressor mass flow.

5.4.3 Convergence and Energy Savings

The convergence behavior of the different optimizing controllers is important to assess their overall performance. For the qualitative and quantitative analysis, the power consumption of the systems will be analyzed for the reference depicted in Figure 5.15. The reference changes (except the first one at $t = 0\,s$) consist of ramps with the slope $\frac{dr(t)}{dt} = 1\frac{kg}{s^2}$ for increases and $\frac{dr(t)}{dt} = -1\frac{kg}{s^2}$ for decreases. The resulting station mass flows for the different controllers can be found in Figure 5.16, whereas the station power consumption is depicted in Figure 5.17.

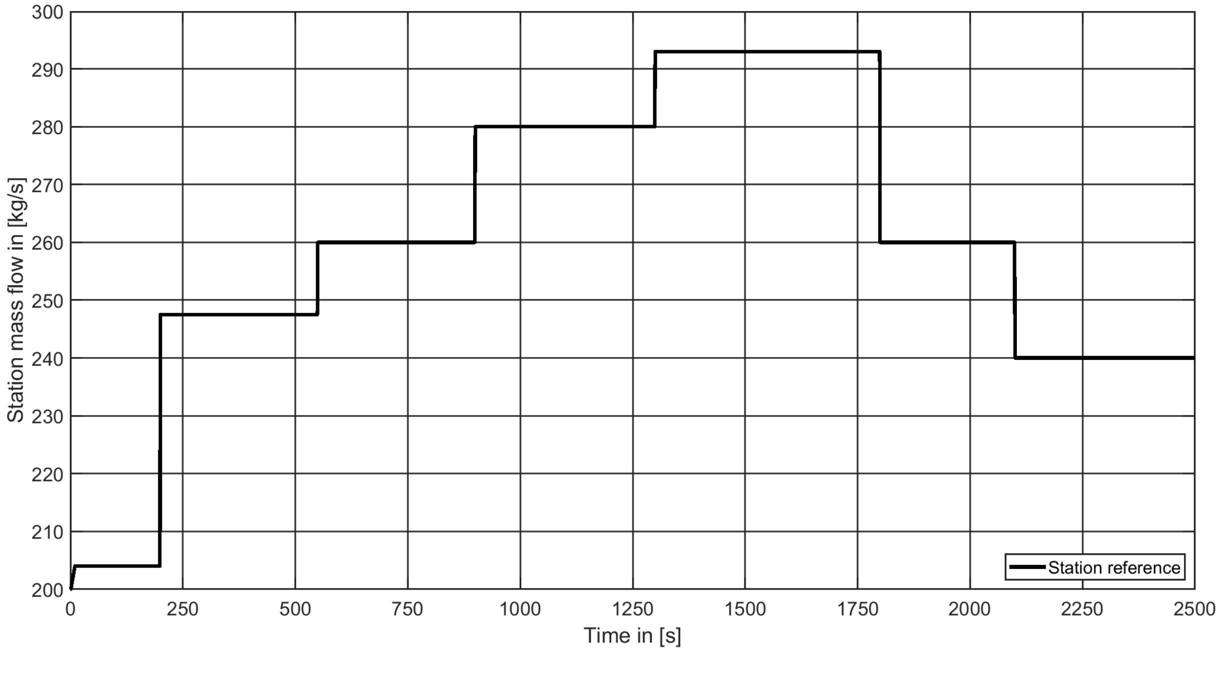


Figure 5.15: Reference signal which is used to analyze the convergence behavior and the energy consumption

The power consumption trajectories of the different systems clearly show that the two-step approach converges to different solutions than the FO controllers. It is also apparent that the FO controller without CTRT has some sampling delay for certain reference changes. This does not impact the convergence speed significantly when comparing the power consumption of the two FO implementations. However, the FO implementation without the PID in parallel tends to overshoot the final value. Some oscillations are also visible for the two-step approach.

The convergence to the final power consumption is characterized by its mean settling time per ramp height ($\frac{t_s}{\kappa}$, where κ is the ramp height) and the corresponding standard deviation. These quantities are calculated based on six reference changes (every reference change in Figure 5.15 but the first one at $t = 0\,s$). Here, the settling time denotes the time it takes the power consumption to stay within $\pm 1\,\%$ of its final value.

In Table 5.3, the mean settling times per ramp height and the corresponding standard deviations are listed. The implementation without CTRT achieves the best results, that is a low mean settling time and a low standard deviation. The settling time is 15 % lower than for the version with CTRT which achieves the second best results. From the standard deviation it is also apparent, that the convergence rate is spread further apart for the FO implementation with CTRT. The two-step approach performs worst with a mean settling time that is 1.75 times larger than the plain FO controller. The standard deviation is very large too. The reason for this bad performance and the large standard deviation can be seen in Figure 5.17. After the reference change at $550\,s$, the two-step approach seems to converge to a solution, until the power consumption abruptly decreases and increases. Because the spikes are large and the controller converges to a different power consumption afterwards, the settling time per ramp height is long for this reference change.

The total energy consumption of each systems is given in Table 5.4. It shows that the two-step approach can find solutions that lead to a decreased energy consumption, but the improvement of less than 0.1 % is negligible. The FO based implementations perform similarly, but FO without CTRT achieves a slightly lower energy consumption with an improvement of 1.7 % compared to the equal load sharing case.

Controller	$\frac{t_s}{\kappa}$ mean $[s/\frac{kg}{s}]$	$\frac{t_s}{\kappa}$ std. deviation $[s/\frac{kg}{s}]$
Equal Load Sharing	4.8	1.6
Two-step approach	6.5	7.0
Feedback optimization	3.7	0.9
Feedback optimization with CTRT	4.4	1.4

Table 5.3: Parameters characterizing the convergence behavior of different systems

Controller	Total energy consumption [MWh]	Improvement to ELS [%]
Equal load sharing (ELS)	26.18	–
Two-step approach	26.16	0.08
Feedback optimization	25.73	1.70
Feedback optimization with CTRT	25.79	1.47

Table 5.4: Energy consumption of the different systems

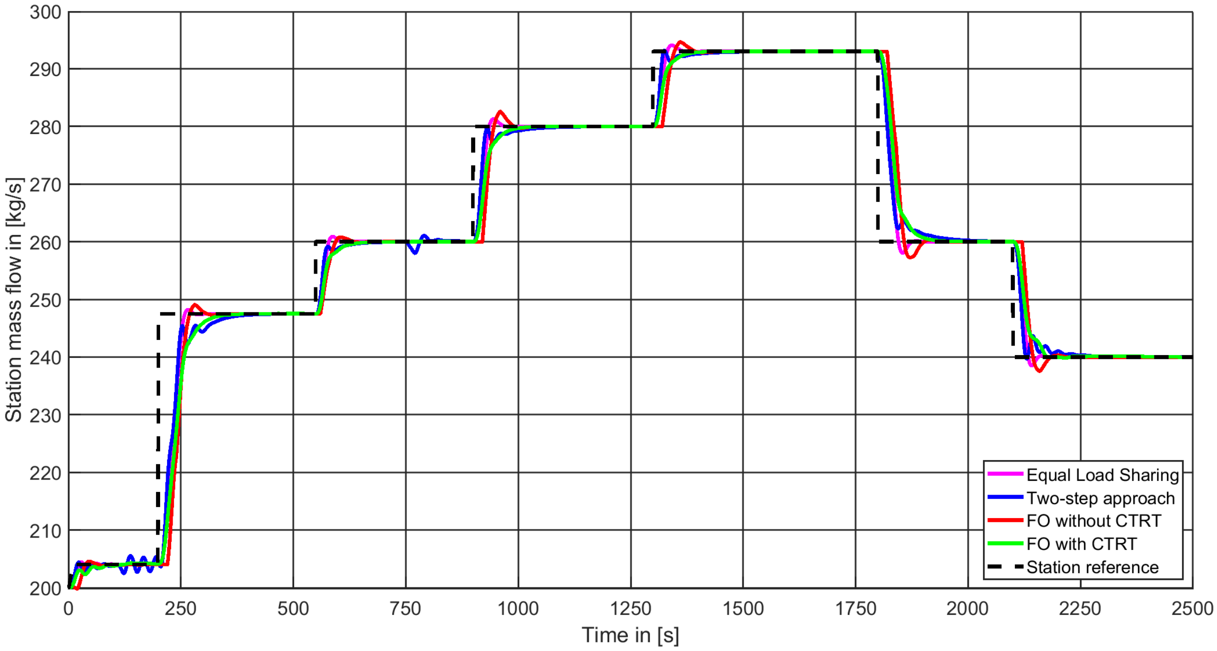


Figure 5.16: Station mass flow for different controllers

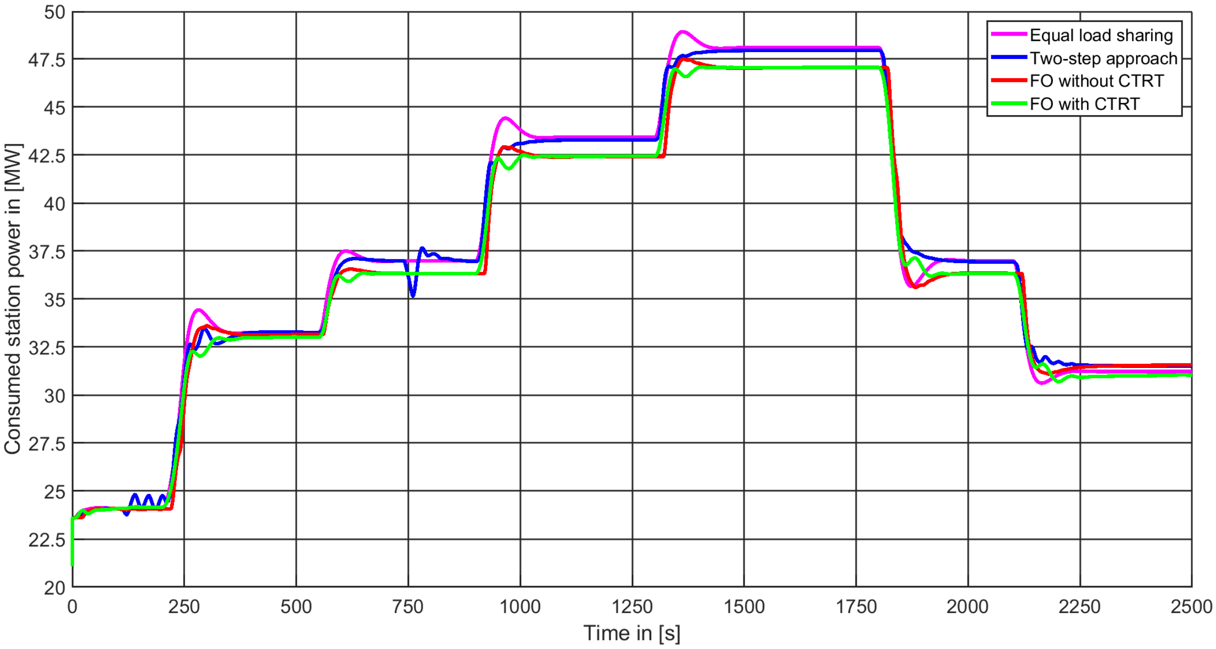


Figure 5.17: Power consumption of the station with different controllers

5.5 Discussion

The previously presented results show that it is possible to apply the FO algorithm presented in Chapter 3 to the compressor load sharing problem. It is possible to tune its parameters such that stable closed-loop operation is achieved.

The reference tracking behavior of the three optimizing controllers differs. The FO controller without CTRT achieves the lowest rise and fall times, whereas the two-step approach the largest. The latter oscillates the most, which also influences its rise and fall times. The settling times show which controller finds a steady-state solution quickly. Among the optimizing controllers, it is lowest for the FO implementation with CTRT. Even the settling times of the plain FO implementation can be lower than the settling times of two-step approach. This is especially remarkable as the plain FO controller has sampling delays. The sampling delays arise when the reference or disturbance changes in between sampling times because the controller does not include any continuous time controller blocks.

The disturbance rejection showed that FO can produce input signals that contain less oscillations than the two-step controller. The disturbance signal used steps and ramps with a height of up to 15 $\frac{kg}{s}$. For these disturbances, the anti-surge controllers do not have to become active and the optimizing controllers were able to steer the station mass flow back to the reference value. The mentioned sampling delays were not encountered but can appear if the disturbance does not coincide with the sampling time of the FO controller.

The power consumption plot shows that the FO controllers can deal much better with the considered structural plant-model mismatch than the two-step approach. The latter achieves an insignificantly lower energy consumption than the equal load sharing controller, whereas the FO controllers are able to save up to 1.7 % of energy (compared to the equal load sharing case) in approximately forty-one minutes of operation. The plain FO implementation also exhibits the fastest convergence to the final power consumption.

The quadratic term in the objective function enables the usage of power measurements in the gradient. This allows FO controllers to reduce the mass flow not only for compressors with a low estimated efficiency, but also for those which have a low true efficiency and a higher estimated efficiency. As FO uses the gradient of the objective function and the plant's sensitivities to approximate a gradient flow, it generates input signals without sudden changes (unless there is a reference change). The two-step approach does not use flows and therefore it is possible that the calculated inputs jump from an equal load sharing solution to one where the compressors get different inputs. This prolongs the settling time and can be observed in Figure 5.16 at 750 s.

The station references for real compressor stations typically change in longer intervals, i.e., in the order of hours and not seconds. Even if the dynamics are slower than the ones considered above, it is clear that no unwanted artifacts arise when using any of two FO-based controllers.

Chapter 6

Conclusion and Outlook

In this thesis, a recently introduced feedback optimization algorithm was applied to a generic load sharing example and to compressors in a parallel configuration. The generic load sharing problem shows that FO can be used as a reference tracking and optimizing controller, either by using CTRT or by including a reference tracking constraint in the optimization problem. It converges to optimal inputs and therefore minimizes the power consumption. Even with structural plant-model mismatch, FO achieves significantly better results than the two-step approach.

The convergence behavior is very similar for both FO implementations. By adapting the integration step size α to different operating ranges, it is possible to increase the convergence speed of FO such that the power consumption is reduced sufficiently fast for all operation points.

The main difference between the two implementations of feedback optimization lays in the reaction to disturbances. The implementation with CTRT reacts faster to disturbances because the change of the station output is noticed by the continuous-time PID controller. The other FO implementation can have a sampling delay between the beginning of the disturbance and the reaction because it runs in discrete time and does not have a continuous time reference tracking part.

The compressor load sharing problem emphasized that low-level flow controllers lead to simple input-output sensitivities that can be computed analytically. By using Gaussian process regression to represent the compressor efficiency maps, it is possible to differentiate between true efficiencies and estimated efficiencies. This leads to structural plant-model mismatch and shows a main disadvantage of the two-step approach: It is not able to find a solution to the constrained optimization problem that corresponds to a significantly lower power consumption.

FO showed that by using a discretized gradient flow, the convergence to a solution contains only little oscillations. It does not jump to a different solution just before convergence is reached and achieves better settling times (with regard to the station power consumption and station mass flow) than the two-step approach.

Overall, FO is superior to the two-step approach due to its steady convergence, better reference tracking and disturbance rejection, and because it achieves lower power consumptions. A comparison of FO with the RTO method "modifier adaptation" would be interesting to see whether a gradient flow with feedback is superior or equivalent to an adaption of the objective function and constraints. Modifier adaption is well-used in process and chemical industries and the comparison could reveal if FO is suitable for this field too. Further, the integration of anti-surge control into the FO controller or the extension of FO to mixed-integer load sharing problems (arise if the shutdown and startup of compressors or other machines is included in the problem formulation) would be challenging extensions of the implemented controllers.

Bibliography

[1] Marcello Colombino, Emiliano Dall'Anese, and Andrey Bernstein. "Online Optimization as a Feedback Controller: Stability and Tracking". In: *IEEE Transactions on Control of Network Systems* 7 (1 Mar. 2020), pp. 422–432.

[2] Marcello Colombino, John W. Simpson-Porco, and Andrey Bernstein. "Towards robustness guarantees for feedback-based optimization". In: *Proceedings of the IEEE Conference on Decision and Control* (Dec. 2019), pp. 6207–6214.

[3] Andrea Cortinovis et al. "Online performance tracking and load sharing optimization for parallel operation of gas compressors". In: *Computers and Chemical Engineering* 88 (May 2016), pp. 145–156.

[4] J. F. Forbes, T. E. Marlin, and J. F. MacGregor. "Model adequacy requirements for optimizing plant operations". In: *Computers and Chemical Engineering* 18 (6 June 1994), pp. 497–510.

[5] Jan Tommy Gravdahl and Olav Egeland. "A Moore-Greitzer axial compressor model with spool dynamics". In: *Proceedings of the 36th IEEE Conference on Decision and Control* 5 (1997), pp. 4714–4719.

[6] Verena Häberle et al. "Non-Convex Feedback Optimization with Input and Output Constraints". In: *IEEE Control Systems Letters* 5 (1 Jan. 2021), pp. 343–348.

[7] Alejandro Marchetti, Benoit Chachuat, and Dominique Bonvin. "Modifier-Adaptation Methodology for Real-Time Optimization". In: *Industrial & Engineering Chemistry Research* 48 (13 Mar. 2009), pp. 6022–6033.

[8] Alejandro G Marchetti et al. "Modifier Adaptation for Real-Time Optimization-Methods and Applications". In: *Processes* 4 (2016).

[9] Predrag Milosavljevic et al. "Optimal Load Sharing for Serial Compressors via Modifier Adaptation". In: *2018 European Control Conference, ECC 2018* (2018), pp. 2306–2311.

[10] Predrag Milosavljevic et al. "Optimal load sharing of parallel compressors via modifier adaptation". In: *2016 IEEE Conference on Control Applications, CCA 2016* (2016), pp. 1488–1493.

[11] Lukas Ortmann et al. "Experimental validation of feedback optimization in power distribution grids". In: *Electric Power Systems Research* 189 (2020).

Appendix A

Controller Parameters and Block Diagrams

A.1 Generic Load Sharing Problem

A.1.1 Generator Dynamics and Low-Level Controllers

The three different transfer functions that are used as simplified generator dynamics are presented in Table A.1. This table also contains information about the efficiency curves.

Machine	Transfer function	Efficiency curve
Generator set 1	$h_1(s) = \frac{1}{8s+1}$	$\eta_1(l_1) = -10^{-7} \cdot l_1^4 - \frac{3}{10000} l_1^3 + \frac{7}{250} l_1^2 - \frac{8}{25} l_1 + 54$
Generator set 2	$h_2(s) = \frac{1}{14s+1}$	$\eta_2(l_2) = -1.1 \cdot 10^{-7} \cdot l_2^4 - \frac{3}{10000} l_2^3 + \frac{3}{100} l_2^2 - \frac{3}{10} l_2 + 40$
Generator set 3	$h_3(s) = \frac{1}{10s+1}$	$\eta_3(l_3) = -10^{-7} \cdot l_3^4 - \frac{7}{25000} l_3^3 + \frac{4}{125} l_3^2 - \frac{7}{25} l_3 + 45$

Table A.1: Transfer functions and efficiency curves of the three generator sets

Concept	Controller	P	I	D	Initial value I_0	Filter coeff.	SPW b	SPW c	T_s $[s]$	Saturation
ELS / two-step	Load PIDs	100	0.1	0	0	100	1	–	CT	–
ELS / two-step	Ref. tracking PID	0.8	0.6	0	0	100	1	–	CT	–
FO	Load PIDs	150	0.2	0	0	100	1	–	CT	–
FO with CTRT	Load PIDs	100	1	0	0	100	1	–	CT	[0,100]
FO with CTRT	Ref. tracking PID	1.5	0.1	5	0	100	1	0	CT	[0,300]

Table A.2: Parameters of the reference tracking and low-level PID controllers

A.2 Compressor Load Sharing Problem

A.2.1 Low-Level controllers

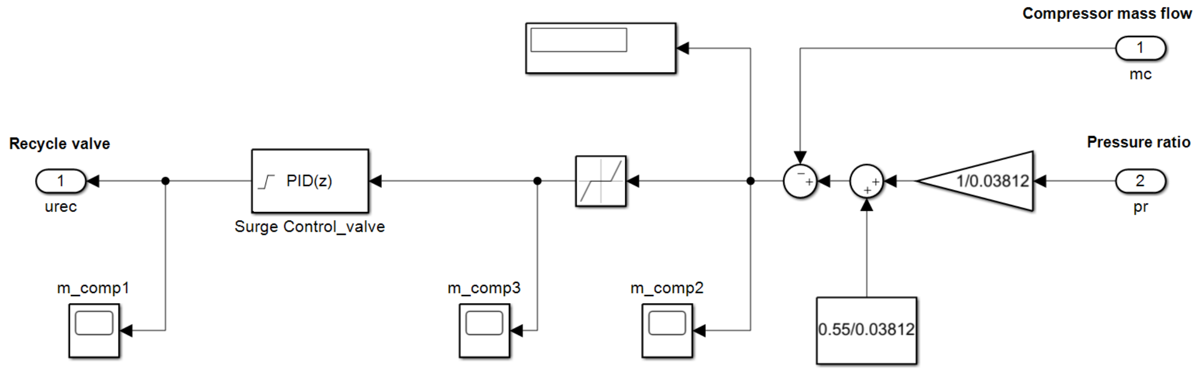


Figure A.1: Block diagram of the anti-surge controller of compressor 1

Figure A.1 shows the block diagram of an anti-surge controller. The structure is the same for all compressors. The gain $\frac{1}{0.03812}$, the constant $\frac{0.55}{0.03812}$, and the settings of the PID controllers are constant for all three anti-surge controllers (see Table A.4 for details). Only the dead zone has different values which are listed in Table A.3.

Controller of...	Start of dead zone	End of dead zone	Saturation on integer overflow
Compressor 1	-15	-1.8	enabled
Compressor 2	-15	-0.25	enabled
Compressor 3	-13	-1.5	enabled

Table A.3: Parameters of the dead zones used in the three anti-surge controllers

The low-level PID controllers that use the compressor mass flow reference and the current compressor mass flow to compute the external torques have different parameters for different compressors. The parameters are listed in Table A.4.

Controller	P	I	D	Initial value I_0	Filter coeff.	SPW b	SPW c	T_s $[s]$	Saturation
Anti-surge PID	0.3/30	1.75/30	0	0	–	1	–	0.1	[0,0.35]
Flow PID comp. 1	2.2	0.4	4	0	100	1	0	CT	–
Flow PID comp. 2	1.5	0.2	2.5	0	100	1	0	CT	–
Flow PID comp. 3	2.2	0.4	4	0	100	1	0	CT	–

Table A.4: Parameters of various PID controllers, where SPW corresponds to "set-point weight" and CT to "continuous time"

The higher level PID controllers that track the station mass flow reference for the controller implementations with CTRT have different parameters. These are listed in Table A.5.

Controller	P	I	D	Initial value I_0	Filter coeff.	SPW b	SPW c	T_s $[s]$	Saturation
Equal load sharing	1	0.1	0	200	–	1	–	CT	–
Two-step approach	0.2	0.05	0.1	200	100	1	0	CT	[195,600]
FO with CTRT	0.8	0.05	0.1	200	100	1	0	CT	[180,600]

Table A.5: Parameters of the PID controllers that track the station mass flow references. SPW corresponds to "set-point weight" and CT to "continuous time".

A.2.2 Model Layer

The model layer is depicted by Figure A.2. It shows the different quantities that are computed by the compressor model and which ones are fed back to the FO controllers. The two-step approach requires the compressor efficiency as additional feedback. The corresponding model layer is shown by Figure A.3.

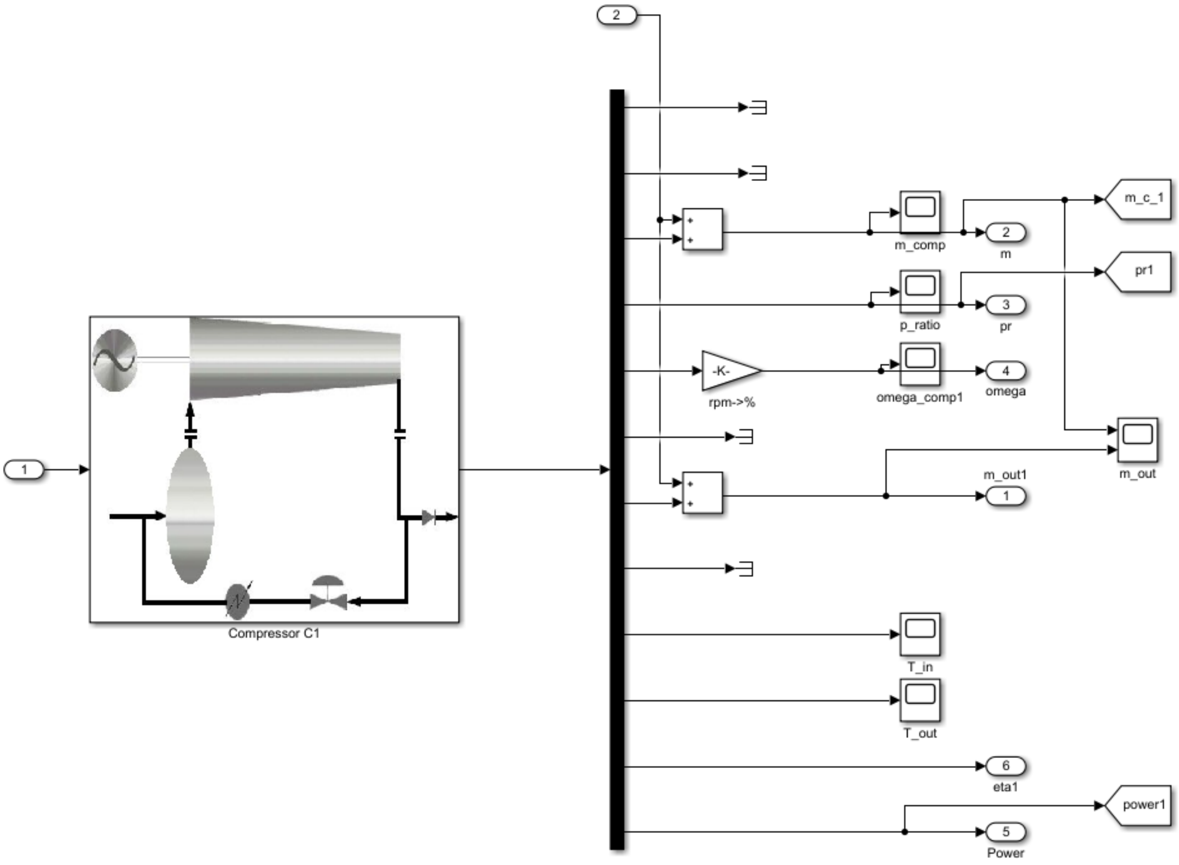


Figure A.2: Model layer showing the compressor model of compressor 1 and the feedback with GoTo-blocks for FO

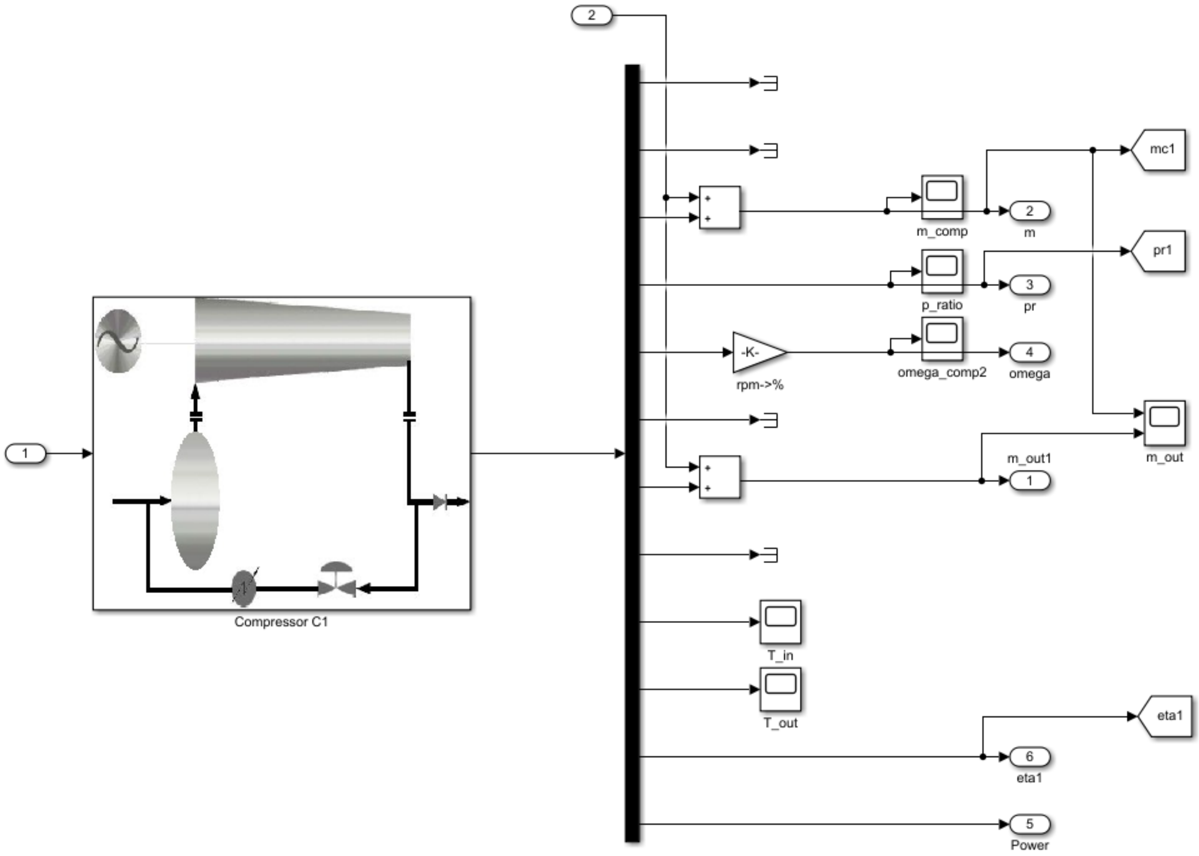


Figure A.3: Model layer showing the compressor model of compressor 1 and the feedback with GoTo-blocks for the two-step approach

Appendix B

Studies on Mechatronics Paper

The following paper was written in the scope of the course "151-0640-00 A Studies on Mechatronics" under the supervision of Dr. Saverio Bolognani and Lukas Ortmann. Prof. John Lygeros was the supervising professor.
The paper compares different optimization methods that can be used to solve the compressor load sharing problem qualitatively. The considered methods include modifier adaptation, real-time iterations, feedback optimization and the two-step approach as the industrial standard.

Studies on Mechatronics: Optimization Methods for Gas Compressors

Maximilian Degner, ETH Zurich (mdegner@student.ethz.ch)

***Abstract*—This paper reviews several methods that can be used to optimize the load sharing of gas compressor arranged in parallel and serial configurations. The qualitative analysis and comparison are focused on online feedback-based optimization methods such as Modifier Adaptation, Real-Time Iterations and the recently proposed Feedback Optimization. The idea of Feedback Optimization is shown by presenting gradient and saddle flow as solvers of constrained optimization problems and by outlining how they can be used as controllers. With the help of a dynamic compressor model, the compressor load sharing problem is formulated and the the challenges of the optimization problem are highlighted. As most of the reviewed methods did not originate in the field of load sharing optimization, suitable methods are identified by looking at the demanding properties of the load sharing problem. The subsequent comparison is based on the convergence behavior, the computational effort, and the model requirements of the identified methods.**

***Index Terms*—feedback-based optimization, gas compressors, online optimization, optimal load sharing**

I. Introduction

Natural gas pipelines rely on compressor stations to ensure gas flow from the production facilities to the consumers. These stations must compensate the pressure loss due to friction and altitude changes (e.g. from sea level to Switzerland), and are therefore needed at regular intervals. The stations can consist of multiple compressors that are arranged in parallel and serial configurations. Compressors consume significant amounts of energy and are also needed in various industries to produce compressed air, to compress process gases, or for control and actuation. Thus, there is a need to minimize the energy consumption of gas compressors.

This work focuses on compressors in parallel and serial configurations, i.e. when multiple compressors are needed to reach a certain pressure or flow rate. In recent years, the operating ranges of modern compressor units used in gas pipelines widened due to the use of variable speed gas turbines or electrical variable frequency drives [1].

The modeling aspects are going to be discussed in a separate section. Nevertheless it is essential to understand the need for optimization in this context: Every compressor has slightly different performance characteristics and thus the energy consumption for a given set-point is not the same for each compressor. To meet the demand (i.e. to satisfy the set-point given by dispatch) and minimize the total energy consumption at the same time, it has to be decided which compressor units are active and at which speeds.

This task cannot be solved easily because of the properties of the efficiency map, the control structure, and the plant's dynamics. The efficiency map (expressing compressor performance) is time-varying, depends on compressor mass flow and pressure ratio, and a structurally correct representation is not available. Disturbances can arise from low-level controllers that are implemented to prevent instabilities. The compressor dynamics have time constants in the range of seconds, however fast convergence is wanted to maximize energy savings. Solving this challenging optimization problem is also called compressor load sharing optimization [1].

In the following sections, the most important methods that can be used to optimize the operation of compressor stations will be reviewed and their potential for solving the optimization problem with little computational effort will be compared. The methods also include very recent ideas and concepts that did not originate in compressor load sharing optimization. A qualitative analysis and comparison is presented, but numerical results are not included in this work. However, with the references provided, it is possible to set up simulations and to compare the controllers quantitatively.

Notation is introduced here. $||v||$ is the 2-norm of the vector $v \in \mathbb{R}^n$, whereas $||\cdot||_G = \sqrt{v^\top G v}$. The Jacobian matrix of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is signified by ∇f. Hence, the gradient of f for $m = 1$ is $\nabla f^\top$. For discrete time systems, $(a)^k$ represents the quantity a at time-step k. Quantities without a superscript k will refer to continuous-time systems. To emphasize the difference between physical quantities, their derivatives and the corresponding flows, e.g. m, $\frac{dm}{dt} = \dot{m}$, and $\overset{*}{m}$ will be used, respectively.

The paper is structured as follows: First, a few main ideas of optimization theory are recalled. Then the compressor dynamics are modeled, operational constraints, and a mathematical description of the load sharing problem in the context of optimization are introduced in section III. Various optimization methods that are used in control systems are covered in section IV. Based on the properties of the load sharing problem, suitable algorithms are identified in section V. These algorithms are compared in section VI.

II. Introduction to optimization problems and optimality conditions

A. Constrained optimization problem

The problem of finding the minimum of a smooth function $f : \mathcal{X} \to \mathbb{R}$ with constraints can be written as

$$\overset{\diamond}{x} = \underset{x \in \mathcal{X}}{\arg\min} \;\; f(x) \tag{1a}$$

$$\text{subject to} \quad x \in \mathcal{C}, \tag{1b}$$

where $\mathcal{C}$ is a closed subset of $\mathcal{X}$. Here, $\overset{\diamond}{x}$ is a *(local) minimizer* if there exists a neighborhood $\mathcal{U} \subseteq \mathcal{C}$ surrounding $\overset{\diamond}{x}$ such that $f(\overset{\diamond}{x}) \le f(x)\ \forall x \in \mathcal{U} \setminus \{\overset{\diamond}{x}\}$ [2, Ch. 2.1].

Regarding the application in control systems, we can rewrite the general constraint $x \in \mathcal{C}$ as equality and inequaliy constraints and limit $\mathcal{X}$ to $\mathcal{X} \subseteq \mathbb{R}^n$. This leads to

$$\overset{\diamond}{x} = \underset{x \in \mathbb{R}^n}{\arg\min}\ \ f(x) \tag{2a}$$

$$\text{subject to} \quad g_i(x) \le 0 \tag{2b}$$

$$h_j(x) = 0 \tag{2c}$$

where $i \in [1, k]$ and $j \in [1, l]$.

B. Optimality in the constrained case

To verify that a candidate point is actually the minimizer of an optimization problem defined in (2), optimality conditions are needed. Thus, the Karush–Kuhn–Tucker (KKT) conditions and the linear independence constraint qualification (LICQ) are going to be recalled.

LICQ [3, Ch. 11.8]: The LICQ holds for a point $\overset{\diamond}{x} \in \mathbb{R}^n$, that satisfies the constraints of (2), if the gradients $\nabla g_i(\overset{\diamond}{x})$ and $\nabla h_j(\overset{\diamond}{x})$ are linearly independent.

KKT [3, Ch. 11.8]: If the point $\overset{\diamond}{x}$ is a local minimum for the problem defined in (2) and the LICQ holds at $\overset{\diamond}{x}$, then there exist a vector $\overset{\diamond}{\lambda} \in \mathbb{R}^p$ and a vector $\overset{\diamond}{\mu} \in \mathbb{R}^q_{\ge 0}$ such that

$$\begin{aligned} \nabla f(\overset{\diamond}{x}) + \overset{\diamond}{\lambda}{}^\top \nabla h(\overset{\diamond}{x}) + \overset{\diamond}{\mu}{}^\top \nabla g(\overset{\diamond}{x}) &= 0 \\ \overset{\diamond}{\mu}{}^\top g(\overset{\diamond}{x}) &= 0. \end{aligned} \tag{3}$$

We consider a slightly altered formulation of (1), where a parameter $\varepsilon \in \mathbb{R}^p$ is added, i.e.

$$\begin{aligned} \overset{\diamond}{x} &= \underset{x}{\arg\min}\, \Phi(x, \varepsilon) \\ &\text{subject to} \quad g(x, \varepsilon) \le 0. \end{aligned}$$

Additional restrictions are

- Φ and g are twice continuously differentiable in x,
- Φ and g and their first and second derivatives in x (∇_x and ∇^2_{xx}) are continuous in ε,
- Φ is strongly convex in $x\ \forall \varepsilon$,
- the polytope defined by the constraint $g(x, \varepsilon)$ is non-empty and convex in x for all ε, and
- the LICQ hold for all ε and all feasible x.

Then, it can be shown that the KKT conditions are necessary and sufficient conditions for optimality [4, Theorem 2]. In general, the KKT conditions are only first-order necessary optimality conditions. This means that the KKT conditions can be used to find candidate points that minimize a function. These candidates are not necessarily minimizers but can also be saddle points of the function.

C. Gradient and saddle flows

Many algorithms were developed, that find points where the conditions for optimality are satisfied. Notable for the scope of this work are gradient and saddle flows.

Gradient schemes exist in various forms, they can be used in discrete or continuous time. A standard form for continuous time, the *gradient flow*, can be defined as

$$\dot{x} = -Q(x) \nabla \Phi(x)^\top, \tag{4}$$

where Q is a metric on the Euclidean space (i.e. a square symmetric positive definite matrix that may depend on the position x). With the help of this metric, other concepts can be recovered such as the *Newton gradient flow*, which has isotropic convergence in continuous time.

To incorporate constraints into gradient flows, barrier and penalty functions can be used. Both types are simply added to the objective function and increase the cost of points that are outside of the feasible set (penalty functions) or close to its boundary (barrier functions). Thus, they are an approximation of inequality constraints. Barrier functions apply constraints strictly (they approach infinity as x approaches a boundary of the feasible set) whereas penalty functions punish the constraint violation (thus usually only acting when x is outside of the feasible set).

Projections are another possibility to include inequality constraints in optimization algorithms. A differentiation between continuous and discrete time is needed here: In continuous time, the gradient is projected, yielding the *projected gradient flow*

$$\dot{x} = \Pi_{\mathcal{X}} \left[-\nabla \Phi(x)^\top \right](x), \tag{5}$$

where $\Pi_{\mathcal{X}}$ represents the projection onto the tangent cone of $\mathcal{X} \subset \mathbb{R}^n$ at the point x. In discrete time, the next calculated point $x^{k+1} = x^k - \alpha^k \nabla \Phi(x^k)^\top$ has be projected onto $\mathcal{X}$. This is equivalent to searching the point $x_p^{k+1} \in \mathcal{X}$ which is closest to x^{k+1}, i.e. minimizing $||x_p^{k+1} - x^{k+1}||^2$. Projected gradient flows are discontinuous systems, because they follow the steepest direction when inside the feasible set and abruptly follow the steepest *feasible* direction when at a boundary. Important properties are similar to gradient flows. Both converge to the KKT points, respectively to a single point if the objective function Φ is convex. The point is a minimizer, if the Hessian matrix is positive definite.

Optimization problems with constraints can also be solved via the associated Lagrangian $L(x, \mu) = \Phi(x) + \mu^\top g(x)$, following the *saddle flow* approach. Here, the constraints are included with one additional variable μ_i per constraint $g_i(x)$ ($\mu = (\mu_1, ..., \mu_r)^\top; g(x) = (g_1, ..., g_r)^\top$). These additional variables are the *dual* variables, whereas the variables of the objective function are the *primal* variables. A gradient descent in the primal and a gradient ascent in the dual variables is performed simultaneously to find the saddle point of $L(x, \mu)$ respectively the tuple $\left(\overset{\diamond}{x}, \overset{\diamond}{\mu}\right)$. This is equivalent to finding the minimizer of the constrained problem [5].

Analogously to gradient flows, saddle flows can be projected onto the tangent cone of $\mu \ge 0$ and the tangent cone of $\mathcal{X}$. In the continuous time case, this leads to

$$\dot{x} = \Pi_{\mathcal{X}} \left[-\nabla_x L(x, \mu)^\top \right] \tag{6a}$$

$$\dot{\mu} = \Pi_{\mathbb{R}^r_{\ge 0}} \left[\nabla_\mu L(x, \mu)^\top \right]. \tag{6b}$$

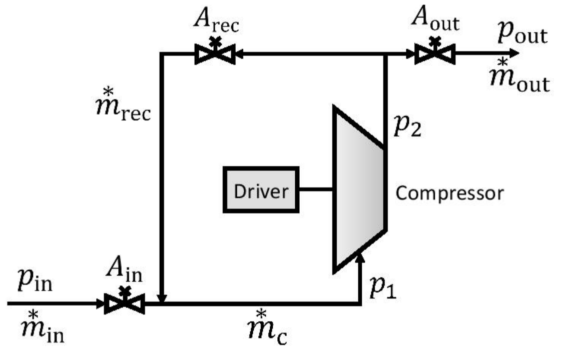

A_{rec}
A_{out}
p_{out}
$\overset{*}{m}_{\text{out}}$
$\overset{*}{m}_{\text{rec}}$
p_2
Driver
Compressor
p_{in}
A_{in}
p_1
$\overset{*}{m}_{\text{in}}$
$\overset{*}{m}_{\text{c}}$

Fig. 1. Diagram of one compressor with surrounding tubing and valves

The previously presented approaches to finding the optimum of an objective function can be combined. Some examples of combinations are the *augmented saddle flow* (saddle flow with additional penalty function for each constraint), or the *mixed saddle flow* which combines dualization of some constraints with the projection onto $\mathcal{X}$ (defined by other constraints). As with gradient flows, the projection is favorable when constraints (e.g. mechanical limits) have to be enforced strictly.

An illustration of the convergence behavior of the different optimization algorithms can be found in [5, Figure 8].

III. Modeling of compressor configurations

A. Model of a single compressor

Models for parallel and serial configurations of gas compressors are based on the model of one simplified compressor. This simplified compressor can be seen in Fig. 1 and incorporates the most important input-output dynamics. As done in [6], [7], a model can be derived by using physical first principles.

Inspired by [6] and the notation in Fig. 1, 'in', 'out', and 'rec' denote the inlet, outlet and recycle parts of the compressor; $\overset{*}{m}_{\text{in}}$, $\overset{*}{m}_{\text{out}}$, $\overset{*}{m}_{\text{rec}}$, and $\overset{*}{m}_{\text{c}}$ are the inlet, outlet, recycle and compressor mass flows. Similarly, p_{in}, p_{out}, p_1, and p_2 are the inlet, outlet, and compressor in- and outlet pressures.

The mass flows depend on the pressures

$$\overset{*}{m}_{\text{in}} = k_{\text{in}} A_{\text{in}} \sqrt{|p_{\text{in}} - p_1|} \tag{7a}$$

$$\overset{*}{m}_{\text{out}} = k_{\text{out}} A_{\text{out}} \sqrt{|p_2 - p_{\text{out}}|} \tag{7b}$$

$$\overset{*}{m}_{\text{rec}} = k_{\text{rec}} A_{\text{rec}} \sqrt{|p_2 - p_1|}, \tag{7c}$$

where $A_{\text{in}}, A_{\text{out}}, A_{\text{rec}}$ represent the inlet, outlet and recycle valve orifice areas and $k_{\text{in}}, k_{\text{out}}, k_{\text{rec}}$ the respective valve gains, which take values between 0 and 1.

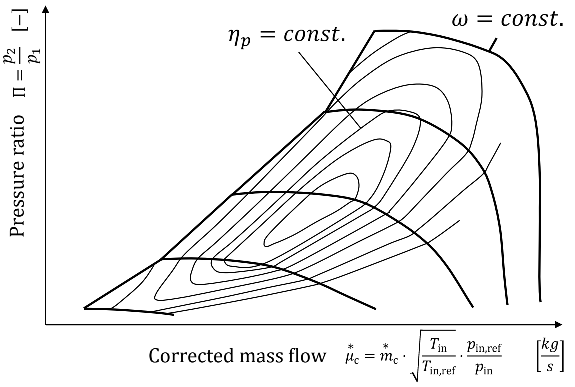


Fig. 2. Schematic compressor map, showing the dependence of the efficiency η_p on the pressure ratio Π, the compressor speed ω, and the mass flow $\overset{*}{m}_c$. Here, the corrected mass flow $\overset{*}{\mu}_c$ is used. It adapts the mass flow $\overset{*}{m}_c$ if the inlet conditions differ from the reference conditions. The surge line is the line that connects the left ends of the ω lines.

The dynamical effects are captured by a system of coupled differential equations

$$\frac{dp_1}{dt} = K_1 \left(\overset{*}{m}_{\text{rec}} + \overset{*}{m}_{\text{in}} - \overset{*}{m}_{\text{c}} \right) \tag{8a}$$

$$\frac{dp_2}{dt} = K_2 \left(\overset{*}{m}_{\text{c}} - \overset{*}{m}_{\text{rec}} - \overset{*}{m}_{\text{out}} \right) \tag{8b}$$

$$\frac{d\overset{*}{m}_c}{dt} = K_3 \left(p_1 \Pi - p_2 \right) \tag{8c}$$

$$\frac{d\overset{*}{m}_{\text{rec}}}{dt} = K_4 \left(\overset{*}{m}_{\text{rec,ss}} - \overset{*}{m}_{\text{rec}} \right) \tag{8d}$$

$$\frac{d\omega}{dt} = K_5 \left(\tau_{\text{ext}} - \tau_{\text{comp}} \right) \tag{8e}$$

where $\tau_{\text{comp}} = \sigma r \omega \overset{*}{m}_c$, σ is the slip factor, r is the radius of the shaft, ω is the angular velocity of the compressor shaft, τ_{ext} is the external torque applied by the driver to the compressor, $\overset{*}{m}_{\text{rec,ss}}$ is the steady-state recycle flow, and K_i $(i = 1, ..., 5)$ are constants. The machine internal pressure ratio can be modeled as a polynomial function $\Pi = f_\Pi(\omega, \overset{*}{m}_{\text{c}})$. This function is the *compressor map* which couples the mass flow, the pressure, and the rotational speed.

The polytropic efficiency $\eta_p = f_\eta(\omega, \Pi, \overset{*}{m}_c)$ depends on the compressor speed, the pressure ratio and the mass flow. It is defined on top of the compressor map, is called *efficiency map* in this paper, and often displayed as a Π vs. $\overset{*}{m}_c$ plot with ω as a parameter and η_p as contour lines. A correction of the mass flow might be necessary to adapt the map to inlet temperature and pressure differing from the reference conditions. A schematic compressor and efficiency map can be found in Fig. 2.

The shaft power depends on the efficiency, the compressor mass flow and the polytropic head y_p. It can be calculated as

$$P = \frac{y_p}{\eta_p} \overset{*}{m}_{\text{c}} \tag{9}$$

$$\text{with} \quad y_p = \frac{Z_{\text{in}} R T_{\text{in}}}{M_w} \frac{n_\nu}{n_\nu - 1} \left[\Pi^{\frac{n_\nu - 1}{n_\nu}} - 1 \right]. \tag{10}$$

The polytropic head is the work needed to increase the enthalpy of the gas by compression when assuming a polytropic change. It depends on the properties of the gas (molecular weight M_w and inlet compressibility factor Z_{in}), the polytropic index n_ν (which describes the thermodynamic process), the temperature on the inlet side T_{in}, and the pressure ratio Π. R is the universal gas constant. Therefore, y_p is a non-constant quantity that mainly changes with Π. The other variables are slow-changing or only change when a different gas is used and can be considered parameters of the function $y_p = f_y(\Pi)$.

B. Constraints on compressor operation

The following mathematical formulation of the constraints on compressor operation are adopted from [6].

$$s_{0,i} - s_{1,i}\overset{*}{m}_{\text{c},i} + \Pi_i \leq 0 \tag{11a}$$

$$c_{0,i} + c_{1,i}\overset{*}{m}_{\text{c},i} - \Pi_i \leq 0 \tag{11b}$$

$$\overset{*}{m}_{\text{c},i} \in \left[\overset{*}{m}{}^{\text{lower}}_{\text{c},i}, \overset{*}{m}{}^{\text{upper}}_{\text{c},i}\right] \tag{11c}$$

$$\tau_{\text{ext},i}\omega_i = P_i \in \left[0, P_i^{\text{upper}}\right] \tag{11d}$$

$$A_{\text{rec},i} \in \left[0, A_{\text{rec},i}^{\text{upper}}\right] \tag{11e}$$

where $i \in \mathcal{N} = \{1, ..., n\}$ with n as the number of compressors; $s_{0,i}$, $s_{1,i}$, $c_{0,i}$ and $c_{1,i}$ are positive constants. The constraints are relevant in the application because if they are violated, serious damages can result [6]:

- Inequality (11a) represents the *surge constraint*. A surge happens, if the flow generated for some pressure ratio is too low. This causes reverse flow in the compressor, heavy vibrations, and mechanical damage . Eventually, a compressor surge leads to machine failure.
- The *choke constraint* is formulated in (11b). It is equivalent to the maximum flow through the compressor and depends on the aerodynamic properties of the compressor and the piping on the outlet side.
- *Mechanical limits* can be found in (11c)–(11e) and apply to flow, consumed power and valve orifice area respectively.

C. Parallel and serial configurations

The interconnection between the compressors of serial and parallel configurations will be presented and afterwards inputs and outputs of the individual compressors will be defined.

When compressors are arranged in a serial configuration, the discharge line of compressor i feeds the suction line of compressor $i+1$. Thus, the interconnection can be modeled as $p_{\text{out,i}} = p_{\text{in,i+1}}$ and $\overset{*}{m}_{\text{out,i}} = \overset{*}{m}_{\text{in,i+1}}$ [7].

Here, the inlet pressure $p_{\text{in},i}$, the recycle valve orifice area $A_{\text{rec},i}$, the applied torque $\tau_{\text{ext},i}$ (and ω_i due to (8e)), and $p_{\text{in},i}$ can be controlled for each individual machine and serve as *inputs*. $\overset{*}{m}_{\text{c},i}$ depends on Π_i (and therefore ω_i); the absorbed torque $\tau_{\text{comp},i}$ is a function of ω_i, $\overset{*}{m}_{\text{c},i}$, $\eta_{\text{p},i}$, and Π_i. Thus, $\overset{*}{m}_{\text{c},i}$, $\overset{*}{m}_{\text{out},n}$, and $\tau_{\text{comp},i}$ are *outputs* of the plant.

A similar notation is possible for parallel compressor systems. One-way valves prohibit unwanted reverse flows, such that constraints on the in- and outflow pressures are not necessary.

Analogous to the considerations for serial configurations, $\tau_{\text{ext},i}$ and $A_{\text{rec},i}$ serve as *inputs*. p_{in} and p_{out} cannot be controlled, as they are external parameters that are fixed by dispatch. $\overset{*}{m}_{\text{c},i}$, $\overset{*}{m}_{\text{out},i}$, and τ_{comp} are the *outputs* of the parallel configuration.

D. Notable properties for optimization problems

When analyzing the constraints stated in (11) and considering characteristics of turbo-machines and controllers, a few aspects are notable. The constraints specified in (11) can be viewed as polytopes and thus as a subset of $\mathbb{R}^n$. If equality constraints must be enforced, they can either be given to the solver as explicit equalities, or if necessary they can be formulated as a narrow box constraint consisting of two inequality constraints.

The compressor dynamics are also subject to time-dependent changes. $f_{\eta,i}$ changes e.g. when fouling and small damages occur and only during bigger maintenance sessions it is possible to obtain extensive measurements to update the efficiency map. The gas related parameters of $f_{y,i}$ change when different gases or gas mixtures are used.

A gas compressor station must follow the reference $r(t)$ (time-series of the set-points given by dispatch). This means that the total output of the system has to be approximately the reference, i.e. for a total mass flow reference $\overset{*}{m}{}^{\text{ref}}_{\text{total}}(t)$:

$$y(t) = \sum_i \overset{*}{m}_{\text{out},i}(t) \approx \overset{*}{m}{}^{\text{ref}}_{\text{total}}(t) = r(t) \quad \forall t \tag{12}$$

The optimizing controllers have to be able to track the reference. If they are not able to track a reference and reject disturbances by adjusting the set-points of the individual machines, a slightly different control structure is needed: The controller has to consist of an optimizing part and a reference-tracking part, e.g. a PID controller. As the optimizing controller acts on top of the PID controller, this structure will be referred to as *delta formulation*. Depending on the reference-tracking controller used (feed-forward, PID, etc.), the closed-loop response to a change in the reference will vary.

Certain operating points are only reachable when using the recycle valve and the recycle valve needs to be opened to prevent a compressor to surge if it is too close to its surge line. The valve is operated by a low-level anti-surge controller, thus the surge constraint does not have to be handled directly by an optimizing controller. For all operating points that do not require the recycle valve to be open, the individual power consumption is minimized if the recycle valve is closed and the input and output valves are fully open. Within the scope of this paper, it is assumed that all operating points are far away from surge.

In the industry, PID controllers are responsible for tracking the flow target specified for the individual compressor [8], [9]. Thus, the optimizing controller do not have to ensure that the individual machines follow their respective targets. With such low-level controllers it is also possible to use the compressor mass flow $\overset{*}{m}_{\text{c},i}$ as input instead of the torque $\tau_{\text{ext},i}$.

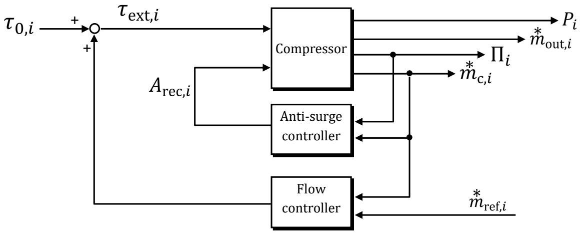


Fig. 3. Block diagram on the level of the ith compressor, consisting of the dynamic compressor model itself, the anti-surge controller and a flow controller. The individual compressor reference is the mass flow target $\overset{*}{m}_{\text{ref},i}$, which comes from the optimizing controller. The anti-surge controller passes the recycle valve orifice area $A_{\text{rec},i}$ to the compressor which opens the valve accordingly. The initial torque $\tau_{0,i}$ is added to the diagram explicitly to emphasize that it might be needed for simulations.

A block diagram showing the different low-level controllers can be found in Fig. 3.

E. Formulation of the load sharing problem

Based on the model and properties presented above, the optimization problem for n interconnected compressors can be formulated similar to the general problem from (2). The objective function includes the individual compressor power consumption, e.g.

$$\Phi(\overset{*}{m}_{\text{c},i}, \Pi_i) = \sum_{i \in \mathcal{N}} P_i = \sum_{i \in \mathcal{N}} \left(\frac{y_{\text{p},i}(\Pi_i)}{\eta_{\text{p},i}(\overset{*}{m}_{\text{c},i}, \Pi_i)} \overset{*}{m}_{\text{c},i} \right), \tag{13}$$

where $i \in \mathcal{N} = \{1, \ ..., \ n\}$.

For parallel configuration the optimization problem can be expressed as

$$\overset{\diamond}{u} = \underset{\overset{*}{m}_{\text{c},i}}{\arg\min} \;\; \Phi(\overset{*}{m}_{\text{c},i}) \tag{14a}$$

$$\text{subject to} \quad \text{steady-state equations} \tag{14b}$$

$$\overset{*}{m}_{\text{c},i} \in \left[\overset{*}{m}_{\text{c},i}^{\text{lower}}, \overset{*}{m}_{\text{c},i}^{\text{upper}} \right] \tag{14c}$$

$$p_{1,i} \in \left[p_{1,i}^{\text{lower}}, p_{1,i}^{\text{upper}} \right] \tag{14d}$$

$$p_{2,i} \in \left[p_{2,i}^{\text{lower}}, p_{2,i}^{\text{upper}} \right] \tag{14e}$$

$$\omega_i \in \left[\omega_i^{\text{lower}}, \omega_i^{\text{upper}} \right] \tag{14f}$$

$$P_i(\overset{*}{m}_{\text{c},i}) \in [0, P_i^{\text{upper}}] \tag{14g}$$

$$\{p_{\text{in}}, p_{\text{out}}, \overset{*}{m}_{\text{total}}\} = \{p_{\text{low}}^{\text{ref}}, p_{\text{high}}^{\text{ref}}, \overset{*}{m}_{\text{total}}^{\text{ref}}\}, \tag{14h}$$

where the quantities with the superscript 'ref' are the reference values that are specified by dispatch. The pressure ratio Π depends on the external parameters p_{in} and p_{out}. Therefore, it cannot be used as decision variable here.

An analogous problem can be formulated for serial configurations, which leads to

$$\overset{\diamond}{u} = \underset{p_{\text{in},i}}{\arg\min} \;\; \Phi(\Pi_i) \tag{15a}$$

$$\text{subject to} \quad \text{steady-state equations} \tag{15b}$$

$$\overset{*}{m}_{\text{c},i} \in \left[\overset{*}{m}_{\text{c},i}^{\text{lower}}, \overset{*}{m}_{\text{c},i}^{\text{upper}} \right] \tag{15c}$$

$$p_{1,i} \in \left[p_{1,i}^{\text{lower}}, p_{1,i}^{\text{upper}} \right] \tag{15d}$$

$$p_{2,i} \in \left[p_{2,i}^{\text{lower}}, p_{2,i}^{\text{upper}} \right] \tag{15e}$$

$$p_{\text{out},i} = p_{\text{in},i+1}, \quad \overset{*}{m}_{\text{out},i} = \overset{*}{m}_{\text{out},i+1} \tag{15f}$$

$$\omega_i \in \left[\omega_i^{\text{lower}}, \omega_i^{\text{upper}} \right] \tag{15g}$$

$$P_i(\Pi_i) \in [0, P_i^{\text{upper}}] \tag{15h}$$

$$\{p_{\text{in},1}, p_{\text{out},n}, \overset{*}{m}_{\text{total}}\} = \{p_{\text{low}}^{\text{ref}}, p_{\text{high}}^{\text{ref}}, \overset{*}{m}_{\text{total}}^{\text{ref}}\}. \tag{15i}$$

$p_{\text{out},n}$ is the outlet pressure of the compressor that feeds the station outlet. Π_i depends on $p_{\text{in},i}$ and $p_{\text{out},i}$. The mass flow through all compressors is (for closed recycle valves) the same and equal to the station reference. Therefore, $\overset{*}{m}_{\text{c},i}$ cannot be used as decision variable.

Constraints (15b) and (14b) refer to the steady-state equations derived from the dynamic compressor model presented in (8) and the compressor map. Note, that not all of the constraints of (14) and (15) are needed, if some assumptions are made (e.g. inflow and outflow pressure are constant due to dynamics with large time constants). However, for completeness, they are all presented here.

IV. Optimizing controllers

This section will give an overview of optimization methods used in control systems which can be implemented in closed-loop with a physical system. The presented methods are capable of solving constrained optimization problems and use measurements to increase robustness against time-varying disturbances. Various established methods and the industrial standard for compressor load sharing will be presented.

A. Optimization algorithms for feedback optimization

Gradient flows and saddle flows, as they are presented in section II-C, can track a slowly changing time-varying minimum of a function, because they simply follow the altered gradient towards a (local) minimum [10].

To make use of flows as controllers that steer a system to an optimal steady-state operating point, they have to be adapted and the flow has to happen in the inputs (i.e. instead of the x used e.g. in (5), we now calculate inputs u). The objective function is defined as a function of inputs u and outputs y and a measurement of the system output $y = h(u)$ must be available. The steady-state map is used to ensure that the calculated inputs produce feasible outputs. We can then calculate the gradient of the cost function for a specific operating point and use it in a gradient descent. The integrative behavior can be transferred to discrete time by using an iterative sum. If a projection is used, it is also necessary to adapt it to discrete time as mentioned in section II-C.

Online feedback optimization controllers are proposed for saddle flows in [10] and for gradient flows in [4]. The latter is a discrete time algorithm and will be introduced in this section. It can handle constraints on inputs and outputs that are formulated like $A \cdot u \leq b$ and $C \cdot y \leq d$, where $A, C \in \mathbb{R}^{m \times n}$ and $b, d \in \mathbb{R}^m$.

It consists of the integral feedback controller with step-size $\alpha > 0$

$$u^{k+1} = u^k + \alpha \widehat{\sigma}_\alpha(u^k, y^k) \quad \text{with } y^k = h(u^k), \tag{16}$$

where $y^k = h(u^k)$ is the measured system output. $\widehat{\sigma}_\alpha(u^k, y^k)$ is the solution of the constrained optimization problem

$$\widehat{\sigma}_\alpha(u, y) = \underset{w \in \mathbb{R}^p}{\arg\min} \left|\left| w + G^{-1}(u) H(u)^\top \nabla \Phi(u, y)^\top \right|\right|^2_{G(u)} \tag{17a}$$

$$\text{subject to} \quad A\left(u^k + \alpha w\right) \leq b \tag{17b}$$

$$C\left(y^k + \alpha \nabla h(u^k) w\right) \leq d. \tag{17c}$$

Here, $G(u^k)$ is a continuous metric on the space of all feasible inputs $\mathcal{U} = \{u \in \mathbb{R}^p | Au \leq b\}$, and $H(u^k)^\top = \left[\mathbb{I}_p \ \nabla h(u^k)^\top\right]$. For space reasons, the superscript $(\cdot)^k$ is dropped in (17a). Due to the feedback part and the optimization algorithm that are connected here, it is called a *feedback optimization* (FO) algorithm.

As $\alpha \to 0^+$, this algorithm actually represents a projected gradient flow and has a global convergence guarantee [4]. It also converges to local minima for non-convex objective functions and nonlinear algebraic plant models [5, Table 1]. Solving the optimization problem in real-time is not a problem, because (17a) can be rewritten as a quadratic program (QP) and solved by using established QP solvers.

B. *Modifier Adaptation*

Real-time optimization approaches (RTOs) comprise multiple adaptive optimization methods, which use measurements to compensate for the effects of unknown disturbances. These methods produce the inputs based on an optimization problem and a model of the plant.

Modifier adaptation (MA) is such a method, it uses the measurements to modify the optimization problem and therefore reduce the effects of disturbances. The goal is to diminish or at least to reduce the effects of plant-model mismatch, but it does not aim at reducing the computational effort or the amount of model information that is needed [5, Section 2.2]. MA was developed in the field of process control and is used for the control of chemical processes where structural plant-model mismatch and inaccurate models are present [11].

MA does not find the optimal inputs through parameter estimation but adds correction terms to the objective function and to the constraints. Consequently, it does not identify a better model over the course of time, but corrects the optimization problem [5, Section 2.2]. The correction terms are updated at every iteration k and consist of the so called modifiers $\epsilon_k^{(\cdot)}$ and $\lambda_k^{(\cdot)}$ and possibly past and present inputs. The updated functions at the kth iteration are

$$\Phi_{m,k}(u) = \Phi(u) + \epsilon_k^\Phi + (\lambda_k^\Phi)^\top (u - u_k) \tag{18a}$$

$$g_{m,i,k}(u) = g_i(u) + \epsilon_k^{g_i} + (\lambda_k^{g_i})^\top (u - u_k) \leq 0. \tag{18b}$$

$\Phi_{m,k}$ and $g_{m,i,k}$ represent the modified objective functions and constraints [11]; the constraints g_i have to be inequality constraints, like in (2b).

Multiple orders of modifiers can be introduced. The zeroth order modifiers ($\epsilon_k^{(\cdot)}$ in (18)) represent the differences between the predicted and actual values, and the first order terms (here $\lambda_k^{(\cdot)}$) serve to diminish the differences in the respective gradients. Higher order modifiers act on the higher order derivatives accordingly. If there are no differences between plant and model at a certain operating point, the modifiers are zero.

The cost and constraint gradients are used to calculate the modifiers and can be inferred from the plant's outputs and the gradients of the outputs with respect to the inputs. Further information on how to approximate the outputs gradients (e.g. with finite difference approximation) and how the modifiers are calculated can be found in [12, Section 3.1] and [13].

MA has the advantage, that (in the absence of noise) upon convergence KKT matching is guaranteed [12, Theorem 1], i.e. the first order conditions for optimality are satisfied. Thus, the solution of the corrected problem corresponds to (feasible) optimal plant inputs which could be theoretically applied directly. Nevertheless, applying these directly to the plant can have unwanted side-effects: The resulting implementation can be sensitive to process noise and lead to excessive corrections, which can compromise the convergence of MA. First order filters can limit the unwanted effects. The filters can be implemented in the concept of MA directly [11].

C. *Real-time iteration schemes*

Real-time iteration (RTI) schemes were developed based on Model Predictive Control (MPC). An introduction can be found in [5, Section 2.3], the references therein, and notably for nonlinear models in [14].

In the past, MPC was limited to plants with a low number of states and relatively long time constants because of the needed computational effort. Therefore, it was mainly used in process industries. Today, MPC methods are widely used. These methods have the goal to stabilize the plant and track a given input while adhering to constraints on states, inputs and outputs. An optimal control problem results and it is solved for a finite receding horizon. In order to do this, a full model is required. The computational effort increases with an increase of the planning horizon.

A variant of the classical MPC approach is *nonlinear MPC*, which can use RTIs [14]. The RTIs approximate the solution of the optimal control problem by carrying out only one iteration of the optimization algorithm (e.g. sequential quadratic programming). This aims at significantly lowering the computational effort and thus ”solving” more optimal control problems with updated parameters in the same amount of time. The change in parameters between the optimization problems becomes smaller and the stability and convergence of the approach can be shown [5].

RTIs can handle constraints on inputs, states, and outputs, because they follow the same problem formulation as the typical MPC methods. These constraints can be inequality constraints for inputs, outputs and states [15].

RTI and MPC methods generally require an exogenous set-point to which they drive the plant while minimizing the cost of the trajectory. However, they do not search for an operating point where the cost is minimal [5].

Economic MPC (EMPC) [16, Ch. 2.8], [17] is a variant, that includes an objective function in its formulation, which reflects the process' economics. As a side-effect, EMPC cannot drive the plant to an exogenous set-point and operate it there. A main aspect of MPC, the stabilization of the plant, has to be reconsidered. Proofs for the closed-loop stability and performance of EMPC can be found in the references in [18]. These proofs are necessary, because the ideas regarding stability cannot be directly transferred from conventional MPC.

The optimization problem solved by EMPC is given in the review paper [18, Equation (9)] as

$$\overset{\diamond}{u} = \underset{u \in \mathcal{S}(\Delta)}{\arg\min} \int_0^{\tau_N} l_e\left(\tilde{x}(t), u(t)\right)\, dx \tag{19a}$$

$$\text{subject to} \quad \dot{\tilde{x}}(t) = f(\tilde{x}(t), u(t), 0) \tag{19b}$$

$$\tilde{x}(0) = x(\tau_k) \tag{19c}$$

$$g(\tilde{x}(t), u(t)) \leq 0, \quad \forall\, t \in [0, \tau_N)\,, \tag{19d}$$

where l_e is the economic cost function, $\mathcal{S}(\Delta)$ is the family of piece-wise constant functions with period Δ, τ_k the kth sampling time instance, τ_N is the time instance when the prediction horizon of length N is reached, $\tilde{x}$ is the open-loop predicted state trajectory which can be obtained from the vector field $\dot{x}(t) = f(x(t), u(t), w(t))$. This vector field results from the standard linear, time-invariant, continuous time state-space form. x is the plant's state vector, u the input vector, and w a disturbance vector.

Usually, the nominal dynamic model of the plant is used as constraint in (19b). The initial value for the trajectory $\tilde{x}(t)$ is obtained from a measurement. Process constraints on states and inputs can be formulated like in (19d). If economic constraints exist, they are often added as inequality constraints. The problem is solved in the same way as with conventional MPC. An optimal piece-wise constant input trajectory is computed and the first control action is sent to the actuators, which implement it until the next sampling instance. The EMPC optimization problem is solved at every sampling time instance.

D. *Industrial practice for gas compressors*

In the industry, a two-step procedure is widely used which falls into the category of adaptive RTO approaches but acts on a different layer than MA. It is probably the most intuitive approach: If the model of our plant changes over time, we regularly update and improve our model. With the improved model, the optimization problem yields the minimizer of the energy consumption. This approach has the advantage, that a predefined model can be used and hypersensitivity to local efficiency map peculiarities can be mitigated. The parameter estimation of such a model is relatively straight forward and can be setup without expert knowledge.

The efficiency maps are updated first by using measurements. This model improvement can be made e.g. by using static nonlinear regression or a least squares approach. In the second step, the optimization of the load-sharing problem is run [7]. The solution of this problem is applied as input to the plant. These two steps are repeated continuously, and eventually convergence is reached [11].

A few assumptions must be fulfilled to ensure good results (i.e. optimality upon convergence): The structural plant-model mismatch must not be significant, the excitations need to be sufficient for the estimation of uncertain parameters [6], and the model must be *adequate*. These assumptions are not satisfied by default for the two-step approach. The quality of the model can be evaluated with the help of the *model-adequacy condition* (MAC) and the analysis tools presented in [19].

MAC [19, Definition 2.1]: If a process model can produce a fixed point which is a local minimum of the optimization problem at the (generally unknown) plant optimum $\overset{\diamond}{u}_p$, it is adequate for the use in a RTO scheme.

This is equivalent to demanding that the RTO optimization problem satisfies first and second order conditions for optimality, i.e. that it matches the KKT points at the plant optimum $\overset{\diamond}{u}_p$ and has a positive definite Hessian matrix at $\overset{\diamond}{u}_p$ [11]. If the model-adequacy conditions are not fulfilled, set-points will be calculated that do not correspond to local minima. Thus, optimality guarantees cannot be given in general [11].

V. Algorithms suitable for the load sharing problem

The selection of optimization techniques presented in section IV shows that there are many different possibilities to steer a system to the solution of an optimization problem. Not every method that was mentioned above can be used directly or without additions in the context of gas compressors, because there are a few properties that must be fulfilled by the methods. As can be seen in (14) and (15), the constraints mainly concern the inputs. Another important aspect is, that the controllers (or at least with some extension) must be able to follow a reference, because the references and mass flows are usually specified by dispatch or a higher level controller. The objective function in (13) changes over time as the efficiency map f_η is changing due to external factors. Therefore, controllers must work in the presence of plant-model mismatch.

To summarize, the controllers have to use feedback, be able to track references, incorporate inequality constraints on the inputs, and they should compute optimal operating points.

Regarding the optimization algorithms, these three qualities are fulfilled by the FO algorithm presented in [4]. It uses feedback when computing $\widehat{\sigma}_\alpha$ and by choosing the inequality constraints carefully, reference tracking is possible. In [10], a feedback optimization controller is presented, that tracks the solution of a constrained optimization problem. Tracking a reference can also be achieved by using a delta formulation.

MA is also able to handle all three requirements as the optimization problem is adjusted at every iteration and it uses a PI or PID controller for the reference tracking.

RTI approaches use the measured state as a starting point for the trajectory optimization and therefore incorporate feedback. They further need an exogenous set-point in general and they can include constraints. To steer a physical system autonomously to an optimal operating point, an additional optimization step might be needed.

The two-step approach presented in section IV-D fulfills all three requirements. It updates the model with measurements and tracks the set-point as it applies a solution of a constrained optimization problem to the plants.

VI. Comparison of the presented algorithms

To compare the methods identified as suitable for the load sharing problem, some topics have to be defined that will be covered by the subsequent section. Thus, the convergence to the (optimal) solution, the computational effort required, the amount of model or gradients required, the possibility of infeasible points, and special requirements or features will be analyzed.

A. Convergence

The FO algorithm converges to the local minimizer, if some technical assumptions on the set of inputs are fulfilled, and the gradients of the plant's steady-state map and objective function are globally Lipschitz continuous [4]. If the objective function is convex, this minimizer is also a global one. If the objective function is not globally convex, the gradient flow based algorithm can get stuck in local minima.

Model adequacy for MA can be guaranteed if convex models are used together with strictly convex constraints (or strictly convex models with convex constraints) [11]. The second order necessary conditions for optimality have to be checked (i.e. whether the Hessian matrix is positive definite). Only if the MAC is satisfied and the second order conditions are satisfied, MA converges to the local optimum [11], [19].

In general, RTIs have the goal to drive the plant to steady-state at the origin. They do this in a stable manner and the trajectory actually converges to the origin eliminating disturbances that might act on the system. However, fulfilling their goal does not lead to minimizing the plant's operational cost. The only MPC variant that can find a point of optimal operation is EMPC, but it has the disadvantage, that the optimal solution is not necessarily a steady-state solution [5].

As already mentioned when presenting the current industrial two-step practice, this method can converge to an optimal solution. The MAC has to be fulfilled for this to happen. A general optimality upon convergence guarantee is therefore not possible and in practice optimality upon convergence rarely happens [12].

B. Computational effort

The optimization problem from (17), i.e. the computation of $\widehat{\sigma}_\alpha$, can be rewritten as a QP for the implementation of FO. This QP can be solved efficiently and fast.

In [12], the computational effort of finding the optimal inputs for the plant are compared for modifier adaptation and the industrial two-step approach. The computational complexity of both methods is linked to the complexity of the optimization problem. In both cases non-linear programming is needed and therefore the same complexity results for both methods. However, the conventional two-step approach needs a parameter estimation step before solving the optimization problem. This step is not needed with MA, and according to the authors of [12], this leads to MA being less computationally demanding.

RTI methods require a full model by design and economic MPC has to solve an optimal control problem at every iteration. This leads to RTI and MPC methods being computationally very demanding and expensive. Furthermore, the computational cost scales with the horizon size and the dimension of the system [5].

C. Model requirements

The presented methods require different amounts of model data. On the one hand, MPC and RTI require a full dynamic model of the plant [5]. On the other hand, feedback optimization only requires the gradient of the objective function with respect to in- and outputs and the sensitivities of the steady-state input-output map of the plant. Usually, the latter can be derived from a plant model, but estimations also lead to sufficient results, because FO is robust against model uncertainty [20].

The industrial two-step approach only converges to the optimum if the MAC is fulfilled, i.e when little structural plant-model mismatch is present. Thus, it needs a structurally correct model, the coefficients will be estimated in real-time on the basis of measurements.

MA lies somewhere in between the other methods regarding the model requirements. MA needs a nominal model but only adapts the cost and constraint function to incorporate plant-model mismatch [12].

D. Constraints and infeasible points

Constraints on the inputs of a plant are easier and more immediate to enforce than constraints on the outputs, because disturbances can violate the output constraints. This might lead to infeasible points and to optimization methods not finding a solution that satisfies the constraints. For the use with compressors, usually only constraints on the inputs are needed. The more challenging surge and choke constraints will be handled by low-level controllers. The input constraints can be formulated as inequality constraints, like in (11).

However, it is still possible, that for certain problem formulations constraints on the output have to be considered. Constraints on the outputs are more challenging because they can be violated by disturbances. MPC-based methods are able to include constraints on outputs, for classical MA this is not possible. If difficulties are encountered during the implementation of these constraints in MA, a problem formulation can be used where output constraints are added to the objective function as penalty or barrier terms.

FO is a method that can work with constraints on the outputs by design, but the considerations regarding disturbances apply here too. Consider the special case, where the inputs are at the boundary of the feasible set and the output is in the feasible set. If a disturbance causes the output to lie outside of the feasible set and to correct the output, inputs are necessary that also lay out of the feasible (input) set, the optimizing algorithm cannot find a feasible solution. This causes either problems with the disturbance rejection, or numerical problems if not handled properly.

E. Special features or requirements

According to [11], it is still unclear whether MA is robust to uncertainties in the gradients of the plant output with respect to the input. This was not covered mathematically when their paper was published, but first simulations suggest that MA is robust to these uncertainties. The same problem applies to all methods that need gradients to compute the solution of the optimization problem, i.e. the FO methods are affected by this robustness question too. The results in [20] show that FO is sufficiently robust to uncertainties in the gradients.

The assumptions for the state cost functions of RTIs, which are typically quadratic, incorporate the requirement for an exogenous set-point [5]. Depending on the setup of the optimization problem, the use of quadratic functions could cause some issues as the power consumption of a single compressor is dictated by the compressor and efficiency map that depends on the compressor speed ω, the pressure ration Π, and the flow $\overset{*}{m}_c$. These maps cannot be described correctly by using quadratic functions.

For all controllers, it is essential that the separation of the plant's and controller's timescales is sufficient to ensure stability. This is especially the case when considering plants that do not have fast-decaying dynamics. Stability bounds for continuous time systems that guarantee close-loop stability are presented in [21]. They can be used to determine how fast a FO controller can be without compromising stability.

VII. Conclusion and outlook

The properties of some methods analyzed in the previous section make their usage as optimizing controller in practice hard and call for other methods. The industrial two-step approach follows an intuitive approach and comes with a predefined model which is not sensitive to local differences in the efficiency map. However, it comes with a relatively high computational effort and it is almost impossible to fulfill the MAC in practice and therefore reach optimality. RTI methods focus on computing optimal state trajectories and all RTI and MPC methods require a full dynamic model. This makes the implementation more difficult and requires expert knowledge, especially if the computational effort should be tuned to a minimum. Further, economic MPC has the problem, that the computed inputs might not lead to steady-state solutions. However, RTIs come with the advantage, that they are able to incorporate measurements very often and can therefore react quickly to disturbances.

The two methods that are best suited to minimize the energy consumption of compressor stations are modifier adaptation and feedback optimization. One the one hand, MA is well-tested and showed good results in simulations (e.g. in [6], [7]). In these papers, it also shown that it is capable of tracking references closely. In comparison to the two-step approach, MA has the advantage, that the MAC are partially fulfilled by design. On the other hand, feedback optimization requires less computational effort (as only QP problems have to be solved) and does not need a full model, but only the steady-state input-output sensitivities and the objective function. It is not yet clear, whether special stability bounds are required for discrete time implementations corrupted by noise [21].

Based on theoretical results, FO has advantages over MA. However, this was not verified with simulations or an implementation on physical systems yet. For a quantitative analysis of the behavior of the different methods, one could use the compressor model provided in section III, implement an anti-surge controller by using the distance of the current operating point to a control line, and use low-level flow controllers for the tracking of flow set-points of the individual machines.

References

[1] A. Cortinovis, M. Mercangöz, M. Zovadelli, D. Pareschi, A. D. Marco, and S. Bittanti, "Online performance tracking and load sharing optimization for parallel operation of gas compressors," Computers and Chemical Engineering, vol. 88, pp. 145–156, 5 2016.

[2] J. Nocedal and S. Wright, Numerical Optimization. Springer, New York, NY, 2 ed., 2006.

[3] D. G. Luenberger and Y. Ye, Linear and Nonlinear Programming International Series in Operations Research & Management Science. Springer, Cham, 4 ed., 2016.

[4] V. Häberle, A. Hauswirth, L. Ortmann, S. Bolognani, and F. Dörfler, "Non-convex feedback optimization with input and output constraints," IEEE Control Systems Letters, vol. 5, pp. 343–348, 1 2021.

[5] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers." https://arxiv.org/pdf/2103.11329.pdf, March 2021.

[6] P. Milosavljevic, A. Cortinovis, A. G. Marchetti, T. Faulwasser, M. Mercangöz, and D. Bonvin, "Optimal load sharing of parallel compressors via modifier adaptation," 2016 IEEE Conference on Control Applications, CCA 2016, pp. 1488–1493, 2016.

[7] P. Milosavljevic, A. Cortinovis, R. Schneider, T. Faulwasser, M. Mercangöz, and D. Bonvin, "Optimal load sharing for serial compressors via modifier adaptation," 2018 European Control Conference, ECC 2018, pp. 2306–2311, 2018.

[8] M. Zagorowska, N. Thornhill, T. Haugen, and C. Skourup, "Load-sharing strategy taking account of compressor degradation," in 2018 IEEE Conference on Control Technology and Applications, CCTA 2018, pp. 489–495, Institute of Electrical and Electronics Engineers Inc., 10 2018.

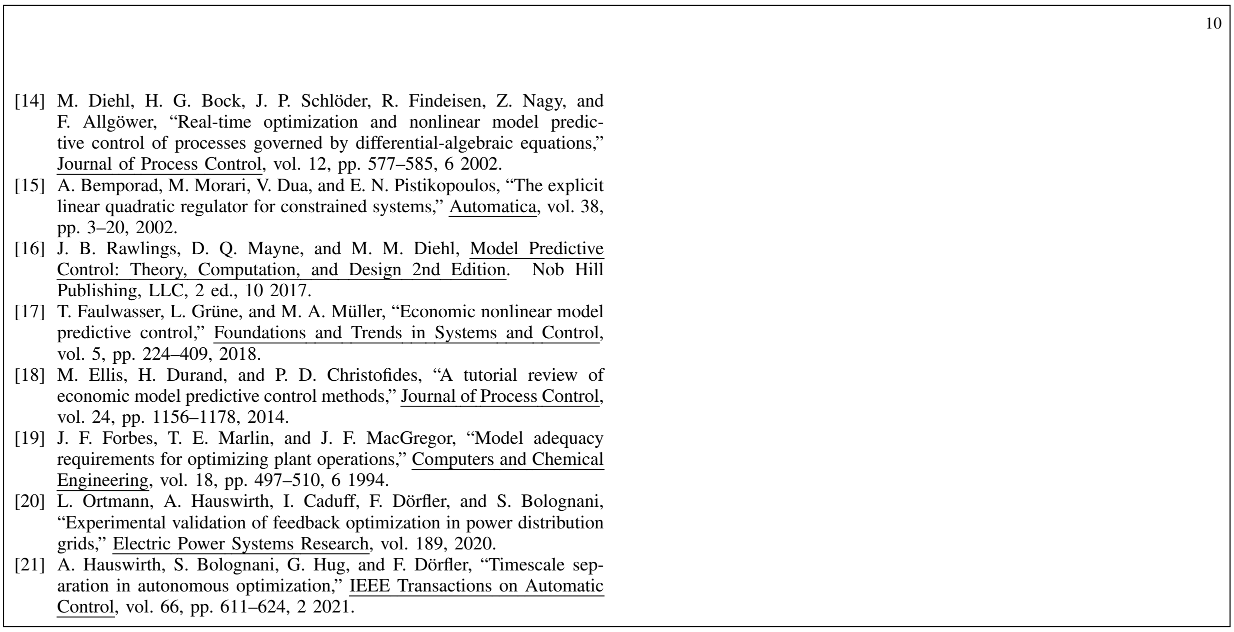
[9] W. Jacobson, M. Zaghloul, A. Dhabi, M. Tolmatsky, S. Staroselsky, and J. Mcwhirter, "Compressor load sharing control and surge detection techniques," tech. rep., Turbomachinery Laboratory, Texas A&M Engineering Experiment Station, 2016. https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/159801/12_Jacobson.pdf.

[10] M. Colombino, E. Dall'Anese, and A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," IEEE Transactions on Control of Network Systems, vol. 7, pp. 422–432, 3 2020.

[11] A. G. Marchetti, G. François, T. Faulwasser, and D. Bonvin, "Modifier adaptation for real-time optimization-methods and applications," Processes, vol. 4, 2016.

[12] A. Marchetti, B. Chachuat, and D. Bonvin, "Modifier-adaptation methodology for real-time optimization," Industrial & Engineering Chemistry Research, vol. 48, no. 13, pp. 6022–6033, 2009.

[13] M. Mansour and J. E. Ellis, "Comparison of methods for estimating real process derivatives in on-line optimization," Applied Mathematical Modelling, vol. 27, pp. 275–291, 4 2003.



[14] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations,” Journal of Process Control, vol. 12, pp. 577–585, 6 2002.

[15] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, “The explicit linear quadratic regulator for constrained systems,” Automatica, vol. 38, pp. 3–20, 2002.

[16] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, Model Predictive Control: Theory, Computation, and Design 2nd Edition. Nob Hill Publishing, LLC, 2 ed., 10 2017.

[17] T. Faulwasser, L. Grüne, and M. A. Müller, “Economic nonlinear model predictive control,” Foundations and Trends in Systems and Control, vol. 5, pp. 224–409, 2018.

[18] M. Ellis, H. Durand, and P. D. Christofides, “A tutorial review of economic model predictive control methods,” Journal of Process Control, vol. 24, pp. 1156–1178, 2014.

[19] J. F. Forbes, T. E. Marlin, and J. F. MacGregor, “Model adequacy requirements for optimizing plant operations,” Computers and Chemical Engineering, vol. 18, pp. 497–510, 6 1994.

[20] L. Ortmann, A. Hauswirth, I. Caduff, F. Dörfler, and S. Bolognani, “Experimental validation of feedback optimization in power distribution grids,” Electric Power Systems Research, vol. 189, 2020.

[21] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, “Timescale separation in autonomous optimization,” IEEE Transactions on Automatic Control, vol. 66, pp. 611–624, 2 2021.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Online Feedback Optimization for Gas Compressors

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Degner	Maximilian

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Zurich, 19 July 2021

Signature(s)
Maximilian Degner

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.