


Frequency-Aware Model Predictive Control

Journal Article**Author(s):**

Grandia, Ruben; Farshidian, Farbod; Dosovitskiy, Alexey; Ranftl, René; [Hutter, Marco](#) 

Publication date:

2019-04

Permanent link:

<https://doi.org/10.3929/ethz-b-000323965>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

IEEE Robotics and Automation Letters 4(2), <https://doi.org/10.1109/LRA.2019.2895882>

Frequency-Aware Model Predictive Control

Ruben Grandia¹, Farbod Farshidian¹, Alexey Dosovitskiy², René Ranftl², Marco Hutter¹

Abstract—Transferring solutions found by trajectory optimization to robotic hardware remains a challenging task. When the optimization fully exploits the provided model to perform dynamic tasks, the presence of unmodeled dynamics renders the motion infeasible on the real system. Model errors can be a result of model simplifications, but also naturally arise when deploying the robot in unstructured and nondeterministic environments. Predominantly, compliant contacts and actuator dynamics lead to bandwidth limitations. While classical control methods provide tools to synthesize controllers that are robust to a class of model errors, such a notion is missing in modern trajectory optimization, which is solved in the time domain. We propose frequency-shaped cost functions to achieve robust solutions in the context of optimal control for legged robots. Through simulation and hardware experiments we show that motion plans can be made compatible with bandwidth limits set by actuators and contact dynamics. The smoothness of the model predictive solutions can be continuously tuned without compromising the feasibility of the problem. Experiments with the quadrupedal robot ANYmal, which is driven by highly-compliant series elastic actuators, showed significantly improved tracking performance of the planned motion, torque, and force trajectories and enabled the machine to walk robustly on terrain with unmodeled compliance.

Index Terms—Legged Robots, Optimization and Optimal Control.

I. INTRODUCTION

TRAJECTORY optimization based on the full dynamics of a robotic system provides a flexible tool to generate complex motion plans. It enables the system to exploit the dynamic capabilities of the robot to achieve a task. State-of-the-art approaches are able to rapidly find solutions while incorporating increasingly complex model descriptions, which allows using trajectory optimization in a Model Predictive Control (MPC) fashion. However, relying on the specific structure of the model makes implementation of the synthesized motion plans prone to modeling errors. Executing motion plans on hardware has therefore proven to be nontrivial and often requires manual, task-dependent tuning of cost functions and constraints to achieve feasible motions.

Manuscript received: September, 10, 2018; Revised December, 10, 2018; Accepted January, 11, 2019.

This paper was recommended for publication by Editor Nikos Tsagarakis upon evaluation of the Associate Editor and Reviewers' comments. This research was supported by Intel Network on Intelligent Systems, the Swiss National Science Foundation through the National Centre of Competence in Research Robotics (NCCR Robotics), the European Unions Horizon 2020 research and innovation programme under grant agreement No 780883. This work has been conducted as part of ANYmal Research, a community to advance legged robotics.

¹ First, second, and last authors are with Robotic Systems Lab, ETH Zurich, Switzerland rgrandia@ethz.ch. ² Third and fourth authors are with Intel Labs, Munich, Germany.

Digital Object Identifier (DOI): see top of this page.

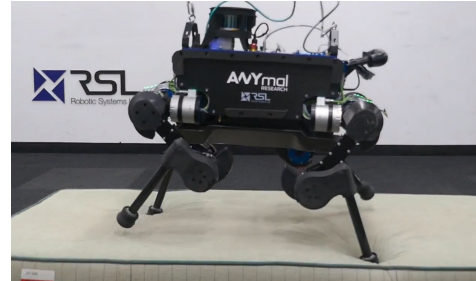


Fig. 1. The quadruped robot ANYmal [1] trotting in place on non-rigid terrain. This experimental setup is used to test the controller's robustness against unmodeled contact dynamics.

A major source of modeling error is the treatment of actuators as perfect torque sources. Any real system is subject to bandwidth limits and as such is not an ideal torque source. A similar modeling error occurs when assuming a rigid contact with the ground. The rigid contact essentially provides the optimizer with infinite bandwidth control over the contact forces. This assumption generally does not hold during locomotion in outdoor environments or on compliant surfaces as shown in Fig. 1. As a result, motion plans generated assuming idealized contact and actuator dynamics cannot be tracked by the hardware, leading to poor tracking performance or failure of the locomotion controller.

In this paper, we extend MPC methods for legged locomotion to situations where the assumptions of rigid ground and perfect actuators are invalid. The selected baseline model describes the 6 degrees of freedom Center of Mass (CoM) dynamics and motion of each leg. Simultaneous optimization of footstep location and contact interaction is achieved by having both contact forces and joint velocities as control inputs. We address the issues of inherent bandwidth limits in real robots by adapting the cost function to be frequency-dependent, making it possible to penalize high frequencies in the motion plans. The solver, therefore, does not have to reason about the exact details of terrain and actuator dynamics but will produce solutions that are achievable under the bandwidth limits. We show that motion plans generated with our frequency-aware trajectory optimization can be followed by the hardware more accurately than those generated with a standard baseline and enables locomotion on compliant terrain.

A. Related work

Local feedback stabilization around a planned motion is a well-known technique to mitigate modeling errors [2], [3], [4], [5] and has led to successful soft ground walking for a bipedal robot [6]. However, especially for dynamic motions, performance can be increased by providing a high-quality feedforward term. The effort in this work to improve the

feasibility of the feedforward term and state reference can be seen as complementary to local stabilization strategies. Moreover, disturbances can be rejected by adjusting the motion plan through fast replanning, reducing the burden on the feedback controller.

In the case of a series elastic actuator, the dynamics can be approximated and added to the model. The optimization algorithm is in those cases able to exploit the properties of the specific actuator and adding spring-damper elements to the joints is even known to result in motions that resemble those found in nature [7]. For series elastic actuators, methods have been proposed to incorporate bandwidth, torque, and joint limits in a computationally efficient way [8][9][10]. However, very often we do not have exact details of actuators and modeling them would lead to high engineering effort. Moreover, since parts of the underlying actuator dynamics have very different time constants, smaller timesteps are required. This leads to slower update rates, preventing such models from use in MPC with complex systems. Additionally, stability problems can arise when a limb with stiffly modeled actuators makes contact with the environment [11]. We avoid such issues by not explicitly modeling the actuators, but by incorporating well-known bandwidth limitations up to which the perfect tracking assumption is valid.

While model parameters for actuators can be obtained from first principles or through repeated experiments, contact dynamics are considerably harder to model or predict. A combination of learning a terrain model and trajectory optimization has been proposed in [12]. However, such methods have not yet reached real-time capabilities.

In the context of contact invariant optimization, soft contact models [13] or contact smoothing [14] are used inside trajectory optimization. These models are selected and tuned for their numerical properties rather than physical accuracy. The models need to be smooth since the highly coupled interaction between stiff contact dynamics and slow dynamics of the robot lead to poor convergence of the algorithms. In the worst case, this numeric model tuning can lead to highly undesired effects when the optimizer starts exploiting dynamic properties of the terrain which are entirely wrong.

Reasoning over higher order terms results in solutions with a higher degree of continuity, which improves performance on hardware considerably. This can be achieved by selecting a smooth parameterization of the solution space as done in spline based optimization [15], [16], [17], collocation methods [18], [19], or when using dynamic motion primitives [20]. This, however, limits the motions that can be expressed and can often require a problem-specific, manual tuning procedure. Alternatively, higher derivatives can be selected as the control inputs in MPC [21]. In Sect. II-D we show that this formulation can be interpreted as a special case of the presented frequency weighting method.

As an alternative to higher order formulations or explicitly modeling actuator dynamics, we propose to encode bandwidth limits through the cost function. A trivial way to do so is to put extra costs on input signals, but this results in slower response overall and goes against the desire to perform highly dynamic motions at the limits of the system. Instead, we intend

to explore a different approach and propose to use a frequency-dependent cost function [22]. We penalize control actions only in the high frequency range and combine this idea with a modern optimal control framework [23], [24].

Frequency-based approaches have been proposed with several applications in both the MPC and legged robotic communities. In [25] constraints on the output spectrum are formed in the frequency domain. However, the proposed window-based constraints approach requires previous and future decision variables, which negatively affects the Riccati-sparsity pattern exploited in optimal control methods. In [26], a Fourier transform has been used to find a closed form solution for momentum compensation of the lower body with the upper body. In contrast to our setting where the unmodeled dynamics are unknown, the forces that are to be compensated for are assumed to be known based on a full rigid body dynamics model.

B. Contributions

We introduce frequency-dependent cost functions integrated into modern MPC strategies for legged locomotion. Through simulation experiments, we study the effect of such a cost function on the resulting solutions. The proposed method provides the user with an intuitive way to achieve robustness against unmodeled phenomena like actuator bandwidth limits and non-rigid contact dynamics. These findings were successfully validated in hardware experiments on different grounds. Using frequency-shaped cost functions, we could improve the robustness of ANYmal while locomoting under substantial external disturbances coming from external pushes or unmodeled soft ground.

II. METHOD

First, we discuss uncertainty in the dynamics from a robustness point of view and motivate the particular choice of cost functions. Afterward, the integration with a Sequential Linear Quadratic Model Predictive Control (SLQ-MPC) method [24] is presented. This method, based on Differential Dynamic Programming, relies on a linear approximation of the dynamics and a quadratic approximation of the cost function around the latest trajectory.

For brevity of notation in the current section, and without loss of generality, we consider the following quadratic cost function without linear and mixed state-input costs,

$$J = \frac{1}{2} \int_0^{\infty} (\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t)) dt, \quad (1)$$

where \mathbf{Q} is the positive semi-definite state cost Hessian and \mathbf{R} is the positive definite input cost Hessian.

A. High frequency robustness

Consider a linear plant $G(j\omega)$ with unstructured multiplicative uncertainty model $L(j\omega)$,

$$\tilde{G}(j\omega) = [I + L(j\omega)]G(j\omega), \quad (2)$$

$$\bar{\sigma}[L(j\omega)] < l_m(\omega), \quad \forall \omega \geq 0, \quad (3)$$

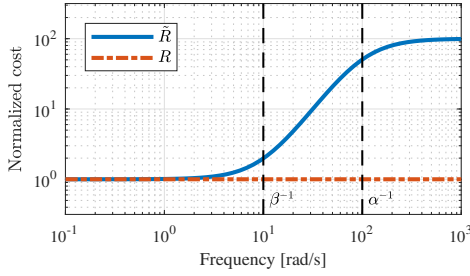


Fig. 2. Example of the frequency-shaped cost function for $\alpha = 0.01$, $\beta = 0.1$, and the standard input costs in the frequency domain. Costs are normalized by R .

where $\bar{\sigma}$ is the maximum singular value of the disturbance model and $l_m(\omega)$ is a frequency-dependent upper bound. The closed loop stability condition for these models is [27],

$$l_m(\omega) < \sigma [I + GK(j\omega)^{-1}], \quad (4)$$

where σ is the minimum singular value, and $GK(j\omega)$ is the transfer function of plant and controller together.

To be robust against large uncertainties at high frequencies, according to (4), the loop gain, $GK(j\omega)$, should be kept low. Intuitively, penalizing inputs at the high frequencies reduces the feedback gain at those frequencies, which allows for larger uncertainty magnitude, $l_m(\omega)$. We therefore propose to use the following frequency-dependent input weighting

$$\tilde{R}(\omega) = \left| \frac{1 + \beta j\omega}{1 + \alpha j\omega} \right|^2 R, \quad \text{with } \beta > \alpha, \quad (5)$$

where R is the original input cost, and $-\beta^{-1}$ and $-\alpha^{-1}$ are the zero and pole of the loopshaping transfer function. A visualization of such cost function is provided in Fig. 2.

Indeed, for Single-Input Single-Output (SISO) systems Anderson *et al.* [28] established that the open loop gain at high frequency under the frequency-shaped cost function (5) is reduced, *i.e.*, $|GK_{\tilde{R}}(j\omega)| < |GK_R(j\omega)|$ for large ω , where $K_{\tilde{R}}$ and K_R are the Linear Quadratic Regulator gains obtained under the frequency-shaped cost and baseline cost respectively. According to (4), the resulting increase in $\sigma [I + GK(j\omega)^{-1}]$ permits a higher model uncertainty in the stopband.

Unfortunately, to the best of our knowledge, a robustness proof for Multiple-Input Multiple-Output systems is not available. Despite that, the intuition that penalizing high frequency input increases compatibility with actuator bandwidth limits remains. In this paper, we aim to empirically validate the effect of using such a cost function.

B. Frequency-shaped cost functions

MPC plans over a receding horizon. The cost function in (5) therefore needs to be expressed in the time domain as well. This can be achieved by a state augmentation as described in [22].

The standard quadratic cost function for the time domain (1) can be converted to the frequency domain (6) according to Parseval's theorem:

$$J = \frac{1}{2\pi} \int_{-\infty}^{\infty} (\hat{\mathbf{x}}(\omega)^H \mathbf{Q} \hat{\mathbf{x}}(\omega) + \hat{\mathbf{u}}(\omega)^H \mathbf{R} \hat{\mathbf{u}}(\omega)) d\omega, \quad (6)$$

where $\hat{\mathbf{x}}(\omega)$, and $\hat{\mathbf{u}}(\omega)$ are the Fourier transform of $\mathbf{x}(t)$ and $\mathbf{u}(t)$, and $(\cdot)^H$ is the Hermitian transpose of the vector. Here, it becomes apparent that the standard costs over states and inputs are constant for all frequencies. To leave the possibility of having different loopshaping per input dimension, the frequency-dependent weight matrix in (5) is generalized to

$$\tilde{\mathbf{R}}(\omega) = \begin{bmatrix} r_1^*(\omega) & & \\ & \ddots & \\ & & r_m^*(\omega) \end{bmatrix} \mathbf{R} \begin{bmatrix} r_1(\omega) & & \\ & \ddots & \\ & & r_m(\omega) \end{bmatrix},$$

$$r_i(\omega) = \frac{1 + \beta_i j\omega}{1 + \alpha_i j\omega}, \quad \beta_i > \alpha_i, \quad (7)$$

where $r_i^*(\omega)$ is the complex conjugate of $r_i(\omega)$. Every input direction can now have its own shaping function r_i . In order to transfer this new cost function back into the time domain, a change of variables is required. The filtered inputs, $\hat{\nu}(\omega)$, are defined elementwise as

$$\hat{\nu}_i(\omega) = r_i(\omega) \hat{u}_i(\omega). \quad (8)$$

After substitution, we arrive back at a frequency-independent cost function over the filtered variables in (10). This cost is then converted back to the time domain in (11),

$$J = \frac{1}{2\pi} \int_{-\infty}^{\infty} (\hat{\mathbf{x}}(\omega)^H \mathbf{Q} \hat{\mathbf{x}}(\omega) + \hat{\mathbf{u}}(\omega)^H \tilde{\mathbf{R}}(\omega) \hat{\mathbf{u}}(\omega)) d\omega \quad (9)$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} (\hat{\mathbf{x}}(\omega)^H \mathbf{Q} \hat{\mathbf{x}}(\omega) + \hat{\nu}(\omega)^H \mathbf{R} \hat{\nu}(\omega)) d\omega \quad (10)$$

$$= \frac{1}{2} \int_0^{\infty} (\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) + \nu(t)^T \mathbf{R} \nu(t)) dt. \quad (11)$$

C. Implementation

The presence of the filtered inputs $\nu(t)$ in the cost function (11) requires the augmentation of the original problem. Considering the original system dynamics, $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$, and state input constraint, $\mathbf{g}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0}$, this can be achieved in several ways. If $\mathbf{r}(\omega)$ consists of proper rationals, its state space realization¹, $(\mathbf{A}_r, \mathbf{B}_r, \mathbf{C}_r, \mathbf{D}_r)$, can be used to substitute for ν in the cost function. The filter's internal dynamics are appended to the system.

$$J = \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + (\mathbf{C}_r \mathbf{x}_r + \mathbf{D}_r \mathbf{u})^T \mathbf{R} (\mathbf{C}_r \mathbf{x}_r + \mathbf{D}_r \mathbf{u})) dt \quad (12)$$

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_r \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{A}_r \mathbf{x}_r + \mathbf{B}_r \mathbf{u} \end{bmatrix}, \quad \mathbf{g}(\mathbf{x}, \mathbf{u}) \leq \mathbf{0}, \quad (13)$$

If $\mathbf{r}(\omega)$ consists of improper rationals, the transfer function $\mathbf{s}(\omega) = \mathbf{r}^{-1}(\omega)$ is defined such that $\hat{u}_i(\omega) = s_i(\omega) \hat{\nu}_i(\omega)$. The state space realization¹ of $\mathbf{s}(\omega)$, $(\mathbf{A}_s, \mathbf{B}_s, \mathbf{C}_s, \mathbf{D}_s)$, is used to substitute for \mathbf{u} in both system dynamics and constraints. Again, the filter's internal dynamics are added to the system.

$$J = \frac{1}{2} \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \nu^T \mathbf{R} \nu) dt \quad (14)$$

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{x}}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}, \mathbf{C}_s \mathbf{x}_s + \mathbf{D}_s \nu) \\ \mathbf{A}_s \mathbf{x}_s + \mathbf{B}_s \nu \end{bmatrix}, \quad \mathbf{g}(\mathbf{x}, \mathbf{C}_s \mathbf{x}_s + \mathbf{D}_s \nu) \leq \mathbf{0}, \quad (15)$$

¹State space realizations are computed according to the balanced realization described in [29]

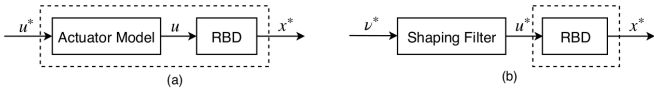


Fig. 3. Block diagrams for actuator modeling (a) and frequency shaping (b). RBD denotes the rigid body dynamics. \mathbf{u}^* and \mathbf{x}^* are the optimized system input and state references to be tracked by the robot.

The system inputs, $\mathbf{u} = \mathbf{C}_s \mathbf{x}_s + \mathbf{D}_s \nu$, are then retrieved after optimization.

Because the selected class of shaping functions are of relative degree zero, choosing between these equivalent options is a numerical consideration. Since the poles of each $r_i(\omega)$ are in our case higher than its zeros, the dynamics of \mathbf{A}_r are faster than those of the realization of its inverse, \mathbf{A}_s , we therefore use the latter formulation.

D. Related methods

The relation with higher order formulations can be understood by considering a shaping function of $r_i(\omega) = j\omega$ for each input in (7) instead of $\frac{1+\beta_i j\omega}{1+\alpha_i j\omega}$. The resulting state space realization is

$$\dot{\mathbf{x}}_s = \nu, \quad \mathbf{u} = \mathbf{x}_s, \quad (16)$$

which is equivalent to $\dot{\mathbf{u}} = \nu$, *i.e.*, the auxiliary input is the derivative of the original input. By selecting $r_i(\omega) = (j\omega)^n$, general n -th order formulations can be retrieved. Higher order methods can thus be seen as a special case of frequency shaping. The proposed method considers more general transfer functions, allowing the user with more flexibility to target a specific frequency range.

There is, however, a numerical consideration when using frequency shaping functions that go to infinity as ω goes to infinity. For such an transfer function $s(\omega)$ will be strictly proper and have a realization with $\mathbf{D}_s = 0$. As can be seen from the constraints in (15), this results in pure-state constraints. This is not a problem in theory, but such constraints are computationally more expensive to handle than state-inputs constraints, so we avoid them in practice to achieve fast replanning.

When using a higher order formulation, inequality constraints can be used to put hard limits on the smoothness of the trajectories. Similar constraints can be placed in the frequency based method as done in [25]. Such a discussion is thus rather a preference for designing behavior through costs or constraints.

The difference between the proposed method and embedding an actuator model can be understood from Fig. 3. When embedding an actuator model in the system dynamics, one would interpret the input to that model as the command to be sent to the robot. However, in the proposed method, the command sent to the robot is the original input, leaving the assumed relation between \mathbf{x} and \mathbf{u} unchanged. In the former, the optimized state input trajectories, $\{\mathbf{x}^*, \mathbf{u}^*\}$ relies on the accuracy of the actuator model, while in the latter, the filter is used to restrict input frequency content to a feasible range.

III. EXPERIMENTAL SETUP

A. Problem Formulation

The proposed method is applied to the kinodynamic model of a quadruped robot, which describes the dynamics of a single free-floating body along with the kinematics for each leg. The Equations of Motion (EoM) are given by

$$\begin{cases} \dot{\boldsymbol{\theta}} = \mathbf{T}(\boldsymbol{\theta})\boldsymbol{\omega} \\ \dot{\mathbf{p}} = {}_W\mathbf{R}_B(\boldsymbol{\theta})\mathbf{v} \\ \dot{\boldsymbol{\omega}} = \mathbf{I}^{-1} \left(-\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \sum_{i=1}^4 \mathbf{r}_{EEi}(\mathbf{q}) \times \boldsymbol{\lambda}_{EEi} \right) \\ \dot{\mathbf{v}} = \mathbf{g}(\boldsymbol{\theta}) + \frac{1}{m} \sum_{i=1}^4 \boldsymbol{\lambda}_{EEi} \\ \dot{\mathbf{q}} = \mathbf{u}_J, \end{cases}$$

where ${}_W\mathbf{R}_B$ and \mathbf{T} are the rotation matrix of the base with respect the global frame and the transformation matrix from angular velocities in the base frame to the Euler angles derivatives in the global frame. \mathbf{g} is the gravitational acceleration in body frame, \mathbf{I} and m are the moment of inertia about the CoM and the total mass respectively. The inertia is assumed to be constant and taken at the default configuration of the robot. \mathbf{r}_{EEi} is the position of the foot i with respect to CoM. $\boldsymbol{\theta}$ is the orientation of the base in Euler angles, \mathbf{p} is the position of the CoM in world frame, $\boldsymbol{\omega}$ is the angular rate, and \mathbf{v} is the linear velocity of the CoM. \mathbf{q} is the vector of twelve joint positions. The inputs of the model are the joint velocity commands \mathbf{u}_J and end effector contact forces $\boldsymbol{\lambda}_{EEi}$.

The constraints depending on the mode of a leg at that point in time are formulated as

$$\begin{cases} \mathbf{v}_{EEi} = \mathbf{0}, & \boldsymbol{\lambda}_{EEi} \in \mathcal{C}(\hat{\mathbf{n}}, \mu), & \text{if } i \text{ is a stance leg} \\ \mathbf{v}_{EEi} \cdot \hat{\mathbf{n}} = c(t), & \boldsymbol{\lambda}_{EEi} = \mathbf{0}, & \text{if } i \text{ is a swing leg} \end{cases}$$

where \mathbf{v}_{EEi} is the end effector velocity in world frame, which constrains a stance leg to remain on the ground and a swing leg to follow the predefined curve $c(t)$ in the direction of the local surface normal, $\hat{\mathbf{n}}$, to avoid foot scuffing with zero contact force inputs $\boldsymbol{\lambda}_{EEi}$. This curve ends with a negative velocity of 0.75 m/s, which is maintained until contact is detected. The friction cone, $\mathcal{C}(\hat{\mathbf{n}}, \mu)$, is defined by the surface normal and friction coefficient, $\mu = 0.7$. This constraint is enforced by projecting the inputs onto the feasible set in the forward rollout of the SLQ-MPC algorithm. Limitations of such a clamping strategy are discussed in [30]. In practice, we find that these constraints are rarely active and that the projection is sufficient for the motions in this work.

The baseline cost is formulated as a quadratic function

$$\begin{aligned} J &= \int_0^T L(\mathbf{x}(t), \mathbf{u}(t)) dt + \Phi(\mathbf{x}(T)) \\ L &= \frac{1}{2} (\mathbf{x} - \mathbf{x}_d)^T \mathbf{Q} (\mathbf{x} - \mathbf{x}_d) + \frac{1}{2} (\mathbf{u} - \mathbf{u}_0)^T \mathbf{R} (\mathbf{u} - \mathbf{u}_0) \\ \Phi &= \frac{1}{2} (\mathbf{x} - \mathbf{x}_d)^T \mathbf{Q}_T (\mathbf{x} - \mathbf{x}_d), \end{aligned} \quad (17)$$

where $\mathbf{x}_d = [\boldsymbol{\theta}_{des}^T, \mathbf{p}_{des}^T, \boldsymbol{\omega}_{des}^T, \mathbf{v}_{des}^T, \mathbf{q}_0^T]^T$ is a desired state consisting of commanded base pose and twist by the user and a default configuration for the joints. Inputs are defined as $\mathbf{u} = [\boldsymbol{\lambda}_{EE1}^T, \dots, \boldsymbol{\lambda}_{EE4}^T, \mathbf{u}_J^T]$, and \mathbf{u}_0 is the equilibrium input for standing in the default configuration. $\Phi(\cdot)$ is the final state cost, which is a heuristic to approximate the truncated

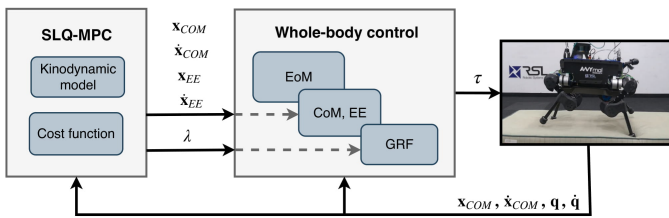


Fig. 4. MPC and whole-body control structure overview. The SLQ-MPC algorithm running on a separate desktop PC sends center of mass (CoM) and end-effector (EE) reference to the onboard whole-body control structure. This hierarchical controller computes torque commands based on the listed priorities.

infinite horizon, and is implemented as a diagonal cost on the base pose and velocities. $L(\cdot, \cdot)$ is the intermediate cost where we use a simple diagonal cost on all state variables and contact force inputs. For the costs on the joint velocities, a diagonal matrix is pre- and post-multiplied by the end-effector Jacobians to define costs over the task space.

B. System integration

In the model described in the previous section, the control inputs are end-effector forces and joint velocities. To translate the solution to torque commands, we extract a full position, velocity, and acceleration plan for CoM and end-effector trajectories, in addition to the planned contact forces. This plan is tracked by the hierarchical whole-body control (WBC) architecture described in [31]. The tasks in decreasing priority are (1) satisfying the equation of motions and zero acceleration for contact feet (2) tracking CoM and swing leg trajectories, and (3) tracking the planned contact forces. The desired contact forces from the MPC are thus communicated in two ways: The CoM trajectory dictates the net acting forces, and force tracking task regulates the internal forces. Without the latter, contact forces would be redistributed among the contacts, which would override the planned smoothness of the trajectories. Additionally, on all priorities, torque limits and friction cone constraints are imposed as inequality constraints.

The optimal control problem in (17) is solved for a user-defined gait with the continuous time SLQ-MPC algorithm described in [23]. We use a receding horizon length of 1.0s, which results in an MPC update rate of 70Hz for the baseline method and around 40Hz for the frequency-shaped method.

The MPC runs on a desktop PC with an Intel Core i7-8700K CPU@3.70GHz hexacore processor and continuously computes a motion plan from the latest known state through a real-time-iteration scheme. The WBC runs on the dedicated onboard PC and tracks the most recent plan. Here, the augmented filter state is propagated as well with the currently available augmented input plan $\nu(t)$. Both nodes communicate over a local network. An overview of the experimental setup is provided in Fig. 4.

IV. RESULTS

We study the effects of adopting the cost function in (11) for the previously described setup under various locomotion tasks. To see the results at different levels of model errors, we conduct perfect model simulations, rigid-body simulations,

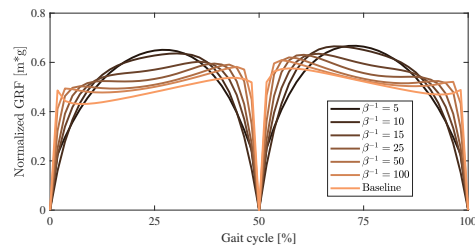


Fig. 5. Ground reaction forces in z-direction for a trotting gait at 0.5 m/s with a period of 0.7 s. The first half of the gait cycle corresponds to the left front foot and the second half to the left hind foot. As the frequency limit decreases from infinity (*i.e.*, baseline) to 5, the smoothness of the planned contact forces increases.

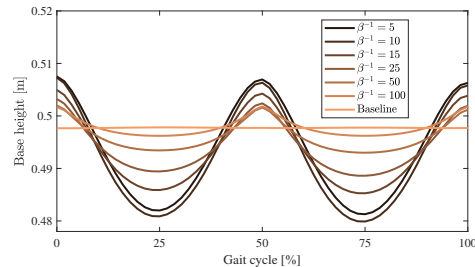


Fig. 6. Base height for a trotting gait at 0.5 m/s with a period of 0.7 s under different smoothing parameters. In general, the vertical displacement of the base increases as the controller becomes smoother since it cannot abruptly increase or decrease the ground reaction forces at the stance feet.

and hardware experiments. When selecting different values for β , α is selected such that the frequencies in the stopband incur a cost of 100 times the steady state cost, *i.e.*, $\alpha = 0.1\beta$.

A. Perfect model

First, we investigate the effect of the loopshaping on the contact forces in a simulation that uses the same model as the MPC. This shows how the resulting trajectories are different already in the case of no modeling errors. The analyzed gait is a trot with a duty factor of 0.5, *i.e.*, with no overlap in stance phases of the diagonal feet, while a forward velocity of 0.5 m/s is commanded. Fig. 5 shows the ground reaction forces when selecting different values for β . As seen from the plot, the baseline method instantaneously applies contact forces once a foot is in contact. As expected, lowering the frequency at which costs start to increase, *i.e.*, lowering β^{-1} , results in increasingly smooth trajectories. The frequency-shaped method approaches the baseline as β^{-1} goes to infinity. The corresponding base height trajectories are plotted in Fig. 6. Smoother contact force trajectories require more vertical displacement of the base, while the baseline produces the exact amount of force to keep the base level.

B. Physics simulation

The combination of tracking controller and MPC is evaluated in the Open Dynamics Engine (ODE) [32] rigid-body simulation, where we can vary ground properties in a controlled way. The model errors, in this case, come from the difference between rigid-body dynamics and the kinodynamic model, as well as the assumption of a rigid ground contact when the terrain is made compliant. ODE allows for simulation of soft

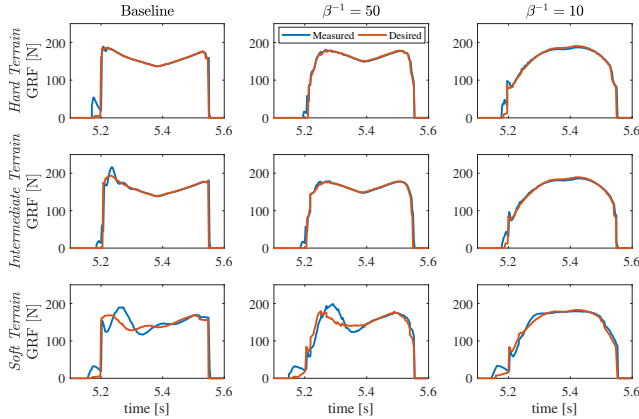


Fig. 7. Measured and desired contact forces in z-direction during a trot in-place for various ground properties and cost functions. The columns from left to right correspond to the baseline, $\beta^{-1} = 50$, and $\beta^{-1} = 10$ cost functions. The rows, from top to bottom corresponds to ground properties with k_p and k_d of $\{1e6, 100\}$, $\{1e5, 50\}$, and $\{1e4, 30\}$. As the compliance of the terrain increases, the difference between desired forces generated from the whole-body controller and resulting forces grows.

contacts by modeling² the ground contact forces as a spring-damper system. Three different sets of spring-damper parameters k_p and k_d are selected to simulate hard, intermediate, and soft ground, respectively. For each terrain, three different cost functions are evaluated: the baseline without frequency shaping as well as frequency-dependent cost functions with $\beta^{-1} = 50$ and $\beta^{-1} = 10$. These values were selected based on Fig. 5 to represent three levels of smoothness in the continuum of available cost functions. The resulting contact force profiles for all combinations are shown for a single stance phase in Fig. 7. Desired and measured contact forces are shown for a single leg during an in-place trotting motion with a stance duration of 0.35 seconds.

As the compliance of the terrain increases, the difference between desired forces generated from the WBC and resulting forces grows. The WBC uses rigid-body dynamics with a hard contact assumption to compute desired contact forces. The difference between desired and measured forces is therefore a measure of disturbances inserted by additional unmodeled dynamics, which in general includes the bandwidth limits of actuators and contact dynamics that we aim to avoid. Table I shows the force tracking performance averaged over six gait cycles in Mean Absolute Error (MAE) and Mean Squared Error (MSE) defined as

$$\text{MAE} = \frac{1}{T} \int_0^T |\lambda - \lambda_{des}| dt, \quad \text{MSE} = \frac{1}{T} \int_0^T (\lambda - \lambda_{des})^2 dt.$$

For all cost functions, tracking performance degrades as the model error increases. Qualitatively, the baseline controller suffers from a larger error at the beginning and end of the contact phase, due to its step-like change in the desired forces. Even on hard terrain, there is an apparent benefit of loopshaping. The smoother transition between contact phases mitigates the disturbance generated by contact timing mismatch. Differences become most apparent for the soft terrain, where the smoother trajectory has better performance especially in MSE.

²ODE relaxes the rigid-contact solver such that it implicitly resembles a spring-damper interaction.

TABLE I
FORCE TRACKING PERFORMANCE MAE (MSE) [N (N²)] FOR DIFFERENT TERRAIN AND COST FUNCTIONS IN SIMULATION.

Terrain $\{k_p, k_d\}$	Cost function		
	Baseline	$\beta^{-1} = 50$	$\beta^{-1} = 10$
Hard $\{1e6, 100\}$	4.8 (303.5)	3.6 (58.5)	5.0 (104.2)
Medium $\{1e5, 50\}$	5.5 (316.2)	4.6 (110.1)	5.0 (100.4)
Soft $\{1e4, 30\}$	13.5 (525.5)	11.6 (286.9)	7.4 (146.5)

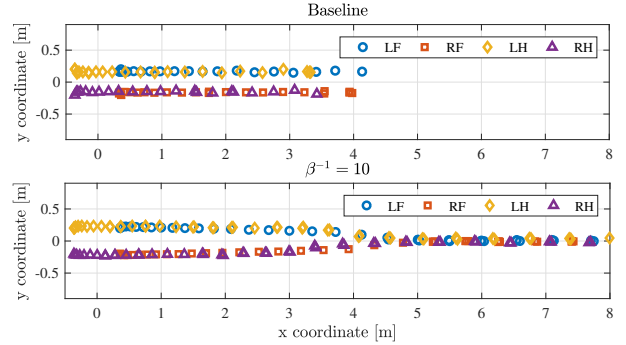


Fig. 8. Foot placement strategy for the baseline (top) and the $\beta^{-1} = 10$ (bottom) cost functions during a trot with increasing velocity commands in the positive x-direction. Commands start at $v = 0$ m/s and accelerate with 0.05 m/s². Failure occurs at 0.6 m/s and 0.9 m/s respectively. The footstep strategy of the baseline method is different from the frequency-aware solution. While the former minimizes the lateral motion of the center of pressure by aligning the support polygons (lines), the baseline method does not adapt its footstep planning based on the velocity.

Furthermore, we examine the locomotion strategy under two extrema of cost functions. The commanded forward velocity during a trotting motion is gradually increased until failure occurs. The foot placement strategies are visualized in Fig. 8. The plots show footstep locations from a top-view with the robot starting at the origin. As the velocity increases towards the right side of the plot, the footstep locations start to differ. Interestingly, with the smoother cost function of $\beta^{-1} = 10$, the foot placement strategy is significantly altered and becomes velocity dependent. Where under the baseline costs the controller chooses a fixed width foot placement, the frequency-shaped solution places the feet increasingly inwards for higher velocities. This can be explained by realizing that horizontal forces, like the normal forces in Fig. 5, smoothly start from and end at zero. During a switch from one contact pair to the next, a lateral force is required to make the CoM velocity change in the direction of the next support line. Under the frequency-dependent cost function, high lateral forces around the contact switch are expensive, and a solution where the supports are more aligned is thus preferred. As the forward velocity increases, so does the required change in lateral velocity for a given width. The alignment of support lines therefore gradually becomes more pronounced.

Remarkably, this results in a significantly higher maximum velocity of 0.9 m/s versus 0.6 m/s.

C. Hardware

On hardware, we aim to validate the simulation results for contact force tracking performance on different terrains; The

TABLE II
FORCE TRACKING PERFORMANCE MAE (MSE) [N (N²)] FOR DIFFERENT
TERRAIN AND COST FUNCTIONS ON HARDWARE.

Terrain	Cost function		
	Baseline	$\beta^{-1} = 50$	$\beta^{-1} = 10$
Hard	27.1 (1465.7)	14.7 (785.5)	12.7 (641.4)
Medium	21.4 (1040.1)	19.7 (741.3)	16.0 (555.4)
Soft	30.9 (2308.1)	26.0 (1237.5)	22.0 (824.3)

floor of the lab, a 3.5 cm foam block, and a mattress are selected to test a rigid, an intermediate and a very compliant terrain respectively. A force-torque sensor is mounted on the right front leg to obtain direct measurements of the ground reaction forces. The resulting measured and desired forces for different cost function and terrain combinations are shown in Fig. 9. The plots show the difference between measured and desired contact forces of the right-front leg during the first three steps. The MAE and MSE averaged over those first three gait cycles are given in Table II. On hard terrain, all methods perform well, and slight differences in tracking performance occur at the beginning and end of the contact phase. In this area the $\beta^{-1} = 10$ controller provides the smoothest transition, resulting in the best MAE and MSE on all terrains. Where in the simulation experiments we see that a medium amount of smoothing is best for hard and medium stiff terrain, we do not see this in the hardware experiments. The difference could be caused by the series elastic actuators of ANYmal, causing model errors and bandwidth limits even on hard terrain. For a step input to the torque level reached during trot, a 90% rise-time up to 35 ms, equivalent to a bandwidth of around 60 rad/s can be expected [1].

When further reducing the compliance by trotting on the mattress, the baseline and $\beta^{-1} = 50$ controllers suffer a substantial decrease in performance. The base height during the first part of the experiment is shown in Fig. 10. Due to the significant mismatch between the planned and resulting contact force with each footstep, the baseline controller loses base height in a few steps, causing it to fail. The $\beta^{-1} = 50$ cost function does achieve a trot, as shown in the video³, but strong oscillations are present between the terrain and the feet. The $\beta^{-1} = 10$ cost function, finally, results in both a stable trot and a smooth contact interaction.

In the accompanying video, we additionally show the behavior under disturbances. The robot trots in place and has costs on base deviation from the initial position. We disturb the robot in the horizontal plane. Qualitatively, the reactive stepping and push-back behavior differ. Under the baseline method, the robot reacts to a push by generating lateral forces and the user experiences instant resistance. The frequency-aware implementation with $\beta^{-1} = 10$ instead accepts deviation of the base trajectory and adapts future step location and force profile to smoothly return to the origin.

V. DISCUSSION

We have shown that a single parameter of a frequency-dependent cost function provides a handle on a rich variety

of solutions. While the smoothness at the beginning of the stance phase is not surprising due to the filter, the anticipatory decrease in force before lifting the foot, as seen in Fig. 5, shows that the filter and planning are tightly working together. The additional filter states allow the Riccati-type algorithm to reason about future state-input constraints, in this case, zero contact forces during the swing phase, and adapts the strategy to approach them smoothly. This is remarkable because the backward pass is projected on those constraints only at the point in time that they are active. Interestingly we also find that the foot placement strategy changes significantly. These observations highlight the fact that embedding frequency awareness of the MPC is richer than simply filtering the obtained inputs. As a future work, we will explore ways to change to cost function online and in this way adapt the locomotion strategy to the terrain.

For the legs in swing phase, optimizing over joint torque, which is effectively one derivative higher than optimizing over joint velocities, can also induce a smoother swing trajectory. However, obtaining real time performance with the extra nonlinear effects is challenging. We believe that keeping the model as simple as possible helps to get a robust solution working on hardware.

VI. CONCLUSION

We introduced frequency-aware MPC by combining frequency-dependent cost functions with modern MPC methods. With simulation experiments, we show that the resulting smoother force profiles improve tracking performance when the rigid terrain assumption is relaxed, without the need to explicitly model it. We validated these results on hardware and see a similar trend when comparing performance on various terrains. The method is shown to provide robustness against unmodeled dynamics of series elastic actuators and compliant terrains. We demonstrated that with this approach ANYmal is now able to execute dynamic motions even on highly compliant terrains.

REFERENCES

- [1] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, “ANYmal - a highly mobile and dynamic quadrupedal robot,” in *IROS*, 2016, pp. 38–44.
- [2] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, “Real time motion generation and control for biped robot -4th report: Integrated balance control-,” in *IROS*, 2009, pp. 1601–1608.
- [3] J. Engelsberger and C. Ott, “Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking,” in *Humanoids*, 2012, pp. 183–189.
- [4] C. Zhou, Z. Li, X. Wang, N. Tsagarakis, and D. Caldwell, “Stabilization of bipedal walking based on compliance control,” *Autonomous Robots*, vol. 40, no. 6, pp. 1041–1057, 2016.
- [5] F. Sygulla, R. Wittmann, P. Seiwald, A. Hildebrandt, D. Wahrmann, and D. Rixen, “Hybrid position/force control for biped robot stabilization with integrated center of mass dynamics,” in *Humanoids*, 2017, pp. 742–748.
- [6] K. Hashimoto, H. Kang, M. Nakamura, E. Falotico, H. Lim, A. Takahashi, C. Laschi, P. Dario, and A. Berthoz, “Realization of biped walking on soft ground with stabilization control based on gait analysis,” in *IROS*, 2012, pp. 2064–2069.
- [7] G. Schultz and K. Mombaur, “Modeling and optimal control of human-like running,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 783–792, Oct 2010.

³<https://youtu.be/RSJgqkk2VRI>

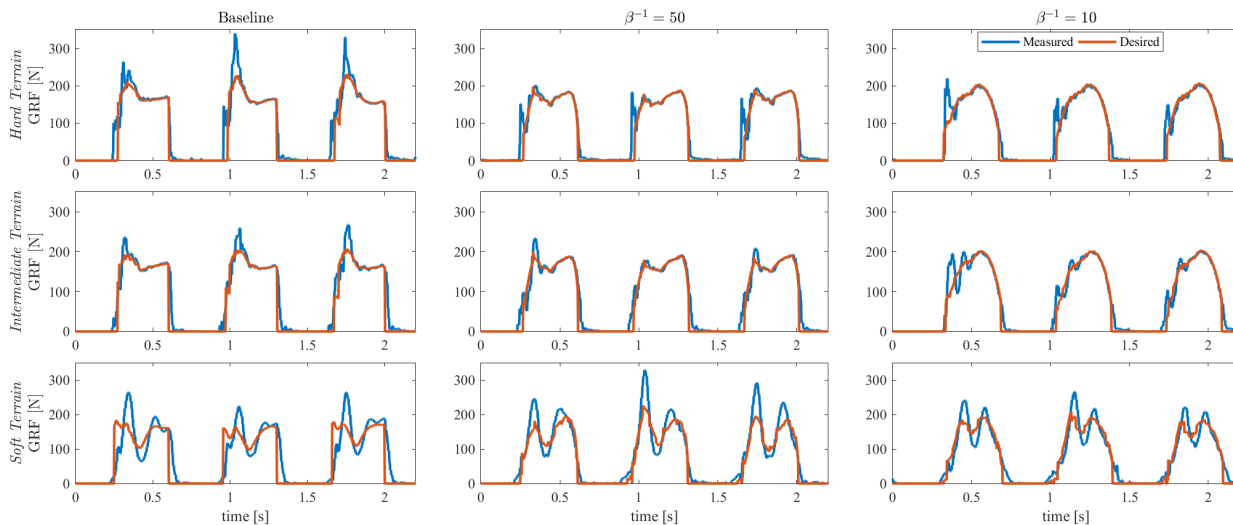


Fig. 9. Measured and desired contact forces for the right front foot during the first three trotting gait cycles on the lab floor (top row), foam (middle row) and a mattress (bottom row) with the baseline (left column), $\beta^{-1} = 50$ (middle column), and $\beta^{-1} = 10$ (right column) cost functions. A smoother transition between swing and stance phase results in better tracking performance.

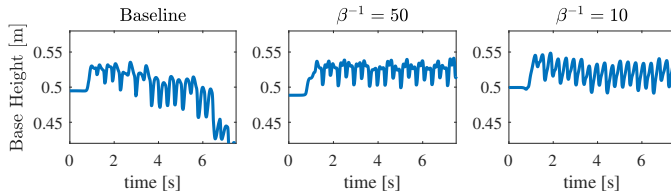


Fig. 10. Measured base height during the first seconds of trotting motion on the mattress with the baseline (left column), $\beta^{-1} = 50$ (middle column), and $\beta^{-1} = 10$ (right column) cost functions. Where the baseline method fails to maintain its height, the frequency-aware controllers remain stable.

- [8] J. Nakanishi and S. Vijayakumar, "Exploiting passive dynamics with variable stiffness actuation in robot brachiation," in *Robotics: Science and Systems*, 2012.
- [9] D. J. Braun, F. Petit, F. Huber, S. Haddadin, P. van der Smagt, A. Albu-Schffer, and S. Vijayakumar, "Robots driven by compliant actuators: Optimal control under actuation constraints," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1085–1101, Oct 2013.
- [10] R. Schlossman, G. C. Thomas, and L. Sentis, "Exploiting the natural dynamics of series elastic robots by actuator-centered sequential linear programming," *arXiv preprint arXiv:1802.10190*, 2018.
- [11] J. Koenemann, A. D. Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *IROS*, 2015, pp. 3346–3351.
- [12] A. H. Chang, C. M. Hubicki, J. J. Aguilar, D. I. Goldman, A. D. Ames, and P. A. Vela, "Learning to jump in granular media: Unifying optimal control synthesis with gaussian process-based regression," in *ICRA*, 2017, pp. 2154–2160.
- [13] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.
- [14] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, p. 43, 2012.
- [15] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Fast, robust quadruped locomotion over challenging terrain," in *ICRA*, 2010, pp. 2665–2670.
- [16] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, July 2018.
- [17] A. Werner, W. Turliej, and C. Ott, "Generation of locomotion trajectories for series elastic and viscoelastic bipedal robots," in *IROS*, 2017, pp. 5853–5860.
- [18] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3d dynamic walking with underactuated humanoid robots: A direct collocation

framework for optimizing hybrid zero dynamics," in *ICRA*, 2016, pp. 1447–1454.

- [19] D. Pardo, M. Neunert, A. Winkler, R. Grandia, and J. Buchli, "Hybrid direct collocation and control in the constraint-consistent subspace for dynamic legged robot locomotion," in *Robotics: Science and Systems*, 2017.
- [20] A. Werner, B. Henze, F. Loeffl, S. Leyendecker, and C. Ott, "Optimal and robust walking using intrinsic properties of a series-elastic robot," in *Humanoids*, 2017, pp. 143–150.
- [21] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *ICRA*, 2003, pp. 1620–1626.
- [22] N. K. Gupta, "Frequency-shaped cost functionals-extension of linear-quadratic-gaussian design methods," *Journal of Guidance, Control, and Dynamics*, vol. 3, no. 6, pp. 529–535, 1980.
- [23] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *ICRA*. IEEE, 2017, pp. 93–100.
- [24] F. Farshidian, E. Jelavic, A. Satapathy, M. Gifthalder, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *Humanoids*, 2017, pp. 577–584.
- [25] J.-H. Hours, M. N. Zeilinger, R. Gondhalekar, and C. N. Jones, "Constrained spectrum control," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1969–1974, 2015.
- [26] K. Hashimoto, H. Kondo, H.-O. Lim, and A. Takanishi, *Online Walking Pattern Generation Using FFT for Humanoid Robots*. Cham: Springer International Publishing, 2015, pp. 417–438.
- [27] J. Doyle and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," *IEEE transactions on Automatic Control*, vol. 26, no. 1, pp. 4–16, 1981.
- [28] B. Anderson and D. Mingori, "Use of frequency dependence in linear quadratic control problems to frequency-shape robustness," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 3, pp. 397–401, 1985.
- [29] B. Moore, "Principal component analysis in linear systems: Control-lability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, 1981.
- [30] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1168–1175.
- [31] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, "Perception-less terrain adaptation through whole body control and hierarchical optimization," in *Humanoids*, 2016, pp. 558–564.
- [32] R. Smith *et al.*, "Open dynamics engine," <http://www.ode.org>, 2005.