2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

On Handling Variable Stiffness Parameters in Compliance Control via MPC

Nikolas Thelenberg and Christian Ott

***Abstract*— In variable impedance control the desired impedance parameters of a robot manipulator are adjusted in real-time. It has been observed that time varying impedance parameters can lead to a non-passive interaction and even destabilize the system. In this paper we consider a compliance controller, where the desired impedance is specified via a time-varying stiffness and damping. We propose to compute the controller gains from an additional online optimization instead of applying the desired time-varying stiffness and damping references directly. In this way the effect of the time-varying impedance parameters is evaluated over a finite time horizon. Together with a terminal constraint on the state this controller formulation aims to avoid a destabilization of the system due to the impedance variation. The proposed method is validated for free movement in two different simulations as well as for physical interaction on a Franka Research 3.**

I. INTRODUCTION

Since its first introduction by Hogan [1] in 1984, impedance control (IC) has been intensively studied and further developed. Due to its compliant behavior, this control approach is suitable for interaction with humans and, more generally, for interaction with unknown and changing environments. Loosely speaking, the error at the end-effector is controlled to imitate a virtual mass-spring-damper system with desired inertia, stiffness and damping properties.

The growing interest in human-robot interaction in recent years [2] has led to a change in requirements for stiffness. Focused aspects of the interaction have been safety [3] or save performance [4]. Another one is the ability to change the stiffness especially during a task. Therefore, IC has been extended to variable impedance control (VIC). One approach is to apply learning algorithms in order to learn the stiffness and damping matrices [5]. One problem inherent in learned stiffness is that stability statements are difficult to obtain. In [6] state-independent conditions on the time-varying stiffness and damping are presented, ensuring asymptotic stability of the system. Another approach to overcome this issue is by using energy tanks [7], [8]. The proposed controller shows high performance, if there is sufficient energy left, but its performance instantly drops, when the reservoir is depleted.

However, it is in general not straightforward how to pick a satisfactory time-varying target stiffness. One attempt depicts the use of quadratic optimization [9]. Model predictive control (MPC) has also been merged with both IC and VIC, respectively. One possible combination of MPC and IC is presented in [10]. There, the compliant impedance behavior has been successfully implemented into an MPC framework, whereby the stiffness and damping parameters have not been variable and were chosen before the usage. Whereas, [11] proposed a combination of MPC and VIC, where the impedance parameters are diagonal matrices and, thus, the decoupled one dimensional mass-spring-damper system with its scalar quantities is subject of the optimization.

Recently, the three fields MPC, VIC and learning algorithms have been combined in various ways. One approach is to learn the robot dynamics with a neural network and then use these dynamics in MPC to minimize only the external human-robot interaction force through various control parameters, mostly stiffness and damping. The optimal control parameters are then used to adapt a low-level IC [12][13]. A similar strategy is pursued in [14], where instead of the robot dynamics the Cartesian impedance model has been trained directly with a neural network.

In this paper, we present a MPC-based strategy to handle variable stiffness parameters. The end-effector follows a reference trajectory and, in addition, maintains its compliant behavior. The virtual mass-spring-damper system, however, obtains its stiffness and damping parameters from a MPC optimization, which takes a variable target stiffness into account and, further, aims for stability. To reduce computational effort, the optimization is performed over the eigenvalues of the parameters instead of their full matrix representation. This strategy does not limit the stiffness to be merely adjustable by the algorithm, but enables the possibility to plan explicit stiffness in advance for certain scenarios.

II. BACKGROUND AND PROBLEM DEFINITION

A. Robot Dynamics

Consider the joint space model of a serial robotic manipulator with $n \in \mathbb{N}$ degrees of freedom (DoF)

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau_c + J^{\mathrm{T}}(q)F_{\mathrm{ext}}, \qquad (1)$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ are the vectors of joint positions, velocities and accelerations, respectively. Further, $M(q) \in \mathbb{R}^{n,n}$ is the symmetric and positive definite inertia matrix, $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ contains the Centrifugal and Coriolis terms and $g(q) \in \mathbb{R}^n$ represents the gravity terms. Moreover, the controlled joint torques are denoted as $\tau_c \in \mathbb{R}^n$, while the torques caused by the time-varying generalized external force $F_{\mathrm{ext}}(t) \in \mathbb{R}^m$ at the end-effector appear in the term $J^{\mathrm{T}}(q)F_{\mathrm{ext}}$.

N. Thelenberg and C. Ott are with the Automation and Control Institute (ACIN), TU Wien, Austria thelenberg@acin.tuwien.ac.at christian.ott@tuwien.ac.at

Copyright ©2024 EUCA

B. Cartesian Impedance Control

The configuration of the end-effector in task space $x \in \mathbb{R}^m$ with $m \leq n$ is obtained by applying the forward kinematic map $f(q) = x$. By differentiation the relations $\dot{x} = \frac{\partial f}{\partial q}\dot{q} = J(q)\dot{q}$ as well as $\ddot{x} = \dot{J}(q,\dot{q})\dot{q} + J(q)\ddot{q}$ are obtained. Here, the Jacobian $J(q) \in \mathbb{R}^{m,n}$ is the same as in (1). For the ease of presentation, we assume that the Jacobian J has full row rank for each joint position q.

For a simpler representation in the task space, the symmetric Cartesian inertia matrix $\Lambda(q) \in \mathbb{R}^{m,m}$, the weighted right pseudoinverse $J^{\#}(q) \in \mathbb{R}^{n,m}$ and the Cartesian Coriolis matrix $\mu(q,\dot{q}) \in \mathbb{R}^{m,m}$ are defined via

$$\begin{aligned} \Lambda(q) &= \left(J(q)M^{-1}(q)J^{\mathrm{T}}(q)\right)^{-1} \\ J^{\#}(q) &= M^{-1}(q)J^{\mathrm{T}}(q)\Lambda(q) \\ \mu(q,\dot{q}) &= \Lambda(q)\left(J(q)M^{-1}(q)C(q,\dot{q}) - \dot{J}(q,\dot{q})\right)J^{\#}(q). \end{aligned} \tag{2}$$

Given are a time-varying reference trajectory $x_{\mathrm{ref}}(t) \in \mathbb{R}^m$ and its derivatives $\dot{x}_{\mathrm{ref}}, \ddot{x}_{\mathrm{ref}} \in \mathbb{R}^m$, we define the error between the actual state and the reference as $\tilde{x}(t) \coloneqq x - x_{\mathrm{ref}}$. Using the task space matrices (2) to define the IC input

$$\begin{aligned} \tau_c = g(q) + \tau_{\mathrm{null}} + J^{\mathrm{T}}\Lambda\Big(JM^{-1}C - \dot{J}\Big)\dot{q} \\ + J^{\mathrm{T}}\Big[\Lambda\ddot{x}_{\mathrm{ref}} - \mu\dot{\tilde{x}} - D_d\dot{\tilde{x}} - K_d\tilde{x}\Big], \end{aligned} \tag{3}$$

where $\tau_{\mathrm{null}} \in \ker(JM^{-1})$ denotes a torque for controlling the nullspace motion. Plugged into (1) and pre-multiplied with ΛJM^{-1} results in the closed loop Cartesian task space system dynamics

$$\Lambda(q)\ddot{\tilde{x}} + \mu(q,\dot{q})\dot{\tilde{x}} + D_d\dot{\tilde{x}} + K_d\tilde{x} = F_{\mathrm{ext}}. \tag{4}$$

This virtual mechanical system is characterized by the inertia matrix Λ, desired stiffness matrix $K_d = K_d^{\mathrm{T}} \succ 0$ and a desired damping matrix $D_d = D_d^{\mathrm{T}} \succcurlyeq 0$ [15]. The advantage of using the Cartesian inertia matrix Λ in (4) instead of an arbitrary desired inertia matrix is that the control input (3) does not rely on a measurement of the external force $F_{\mathrm{ext}}(t)$ [16].

C. Variable Impedance

In the classic compliance control system (4) the desired matrices D_d and K_d are constant with respect to time, but for variable stiffness time-dependent matrices are needed. Using time-dependent $K_d(t)$, however, might be problematic from a stability point of view. This becomes even clearer if one takes a closer look at the generalized energy function V of the closed loop system (4) and its derivative

$$\begin{aligned} V(t) &= \frac{1}{2}\dot{\tilde{x}}^{\mathrm{T}}\Lambda\dot{\tilde{x}} + \frac{1}{2}\tilde{x}^{\mathrm{T}}K_d(t)\tilde{x} \\ \dot{V}(t) &= -\dot{\tilde{x}}^{\mathrm{T}}D_d\dot{\tilde{x}} + \dot{\tilde{x}}^{\mathrm{T}}F_{\mathrm{ext}}(t) + \frac{1}{2}\tilde{x}^{\mathrm{T}}\dot{K}_d(t)\tilde{x}. \end{aligned}$$

Here, the term $\frac{1}{2}\tilde{x}^{\mathrm{T}}\dot{K}_d\tilde{x}$ has indefinite sign and, thus, violates the passivity/ stability. In previous work this issue has been handled for example with energy tanks [7].

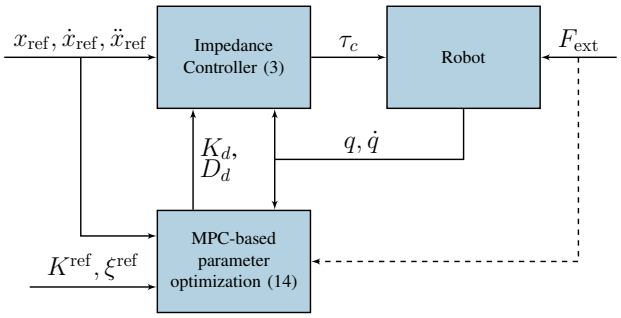


Fig. 1: Schematic structure of the control loop showing the low-level impedance controller and the high-level parameter optimization

III. MPC-based Variable Parameter Optimization

The general schematic structure of the proposed control system is pictured in Fig. 1. The low-level impedance controller ensures the compliant behavior of the end-effector by applying the control torque (3) to the robot and is executed in real time. The high-level MPC-based parameter optimization generates the symmetric and positive definite stiffness matrix $K_d(t)$ as well as the damping matrix $D_d(t)$. The optimization is executed based on the current errors $\tilde{x}, \dot{\tilde{x}}$, the inertia matrix Λ, the target reference stiffness K^{ref} and the external force F_{ext}, respectively. The core feature of the algorithm, is to perform the optimization on the eigenvalues of the stiffness and damping matrix rather than on their entries. This reduces the number of optimization variables drastically that even the simultaneous consideration of stiffness and damping is compensated and the computation time is shortened even further.

The reference stiffness $K^{\mathrm{ref}} = K^{\mathrm{ref}}(t) \in \mathbb{R}^{m,m}$ must be symmetric and positive definite at every time. Since the optimization algorithm is time-intensive, the high-level MPC-based parameter optimization is executed with a lower frequency, i.e. for $j = 1, 2, \ldots$ only at certain sample instances t_j.

A. Prediction Model

Beginning with t_j, we want to predict the system dynamics (4) for future $T > 0$ seconds. For ease of presentation the shifted time interval $[0, T]$ of $[t_j, t_j + T]$ is considered for the prediction and likewise the optimization. First, for the prediction on $[0, T]$, the Cartesian inertia matrix Λ is kept constant, i.e. $\Lambda = \Lambda(q(t_j)) = const.$ As a result, the Cartesian Coriolis matrix $\mu(q,\dot{q})$, which ensures passivity of (4), is not required in the prediction model. As well as the inertia matrix, the generalized external force $F_{\mathrm{ext}} = F_{\mathrm{ext}}(t_j) = const.$ is considered on $[0, T]$. Moreover, we abbreviate $K^{\mathrm{ref}}(t_j + t)$ with $K_j^{\mathrm{ref}}(t)$ for $t \in [0, T]$. Since $\tilde{x}$ represents the error of the end-effector, which is computed in real time and used in the control input (3), its prediction is denoted as $y(t) \in \mathbb{R}^m$ with $t \in [0, T]$.

With these definitions and assumptions, the complete

prediction model is stated as

$$\Lambda\ddot{y} + D(t)\dot{y} + K(t)y = F_{\text{ext}} \tag{5}$$

for $t \in (0,T)$ and the initial values $y(0) = \tilde{x}(t_j)$ and $\dot{y}(0) = \dot{\tilde{x}}(t_j)$. The symmetric and positive definite stiffness and damping matrices $K \in \mathbb{R}^{m,m}$ and $D \in \mathbb{R}^{m,m}$ will be optimized over the time horizon $[0,T]$ and afterwards applied as K_d and D_d in the control input (3).

B. Double Diagonalization and Optimization Variables

Our objective is that the stiffness K follows the given reference K^{ref} as closely as possible and ideally coincide with it. Therefore, the damping matrix D is designed such that it depends directly on the reference stiffness K_j^{ref} as well as on the inertia matrix Λ. The used method for the design of the damping matrix is double diagonalization [17]. First, A Cholesky decomposition of $\Lambda = LL^{\text{T}}$ is performed with the lower triangular matrix $L \in \mathbb{R}^{m,m}$. Then, two matrices $Q^{\text{ref}}, H \in \mathbb{R}^{m,m}$ as well as the vector $v^{\text{ref}} \in \mathbb{R}^m$ are computed as follows

$$\begin{aligned}\left[Q^{\text{ref}}, v^{\text{ref}}\right] &= \text{eig}\left(L^{-1}K_j^{\text{ref}}L^{-\text{T}}\right) \quad \text{and} \\ H &= LQ^{\text{ref}},\end{aligned} \tag{6}$$

where Q^{ref} contains the eigenvectors and v^{ref} the corresponding eigenvalues of $L^{-1}K_j^{\text{ref}}L^{-\text{T}}$. Using these matrices and the subsequent relation $\Lambda = HH^{\text{T}}$, the prediction model (5) can be rewritten into the form

$$HH^{\text{T}}\ddot{y} + D\dot{y} + H\,\text{diag}(v^{\text{ref}})H^{\text{T}}y = F_{\text{ext}}. \tag{7}$$

If we use an arbitrary vector $v \in \mathbb{R}^m$ instead of v^{ref} in (7), the stiffness matrix of the system becomes

$$K(v) = H\,\text{diag}(v)H^{\text{T}} \neq K_j^{\text{ref}}. \tag{8}$$

With this step the reference stiffness K_j^{ref} is not directly applied to the system any more, but it is still utilized to compute Q^{ref} and, thus, the transformation matrix H. Then, the chosen damping matrix has the form

$$D(\xi, v) = 2H\,\text{diag}(\xi \odot \sqrt{v})H^{\text{T}}, \tag{9}$$

with the vector of damping ratios $\xi \in \mathbb{R}^m$. In $\sqrt{v}$ the square root is taken element-wise and also $\odot$ denotes the element-wise multiplication of ξ and $\sqrt{v}$.

With the double diagonalization technique, the stiffness and damping matrix are adjustable in terms of their transformed eigenvalues v and damping ratios ξ, respectively. If the optimization were performed directly on K and D and all their entries were considered as optimization variables, there would be $m(m+1)$ variables due to the symmetry of the two matrices. Whereas, by choosing v and ξ as optimization variables, we can modify and, therefore, optimize $K(v)$ and $D(\xi, v)$ with the reduced amount of $2m$ variables.

C. State System Dynamics and Discretization

Up next, the state of the mass-spring-damper system (7) is defined as $z(t) \coloneqq [y^{\text{T}} \quad \dot{y}^{\text{T}}]^{\text{T}} \in \mathbb{R}^{2m}$ for $t \in [0,T]$. Using the state as well as the derived structure of the stiffness and damping matrix in (8) and (9), respectively, the prediction model (7) is rewritten in state-space representation as

$$\begin{aligned}\dot{z} &= \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} z + \begin{bmatrix} 0 \\ (HH^{\text{T}})^{-1}F_{\text{ext}} \end{bmatrix} \\ &=: \tilde{h}\left(z, \xi, v, H, F_{\text{ext}}\right)\end{aligned} \tag{10}$$

for $t \in (0,T)$ and the inital condition $z(0) = [y^{\text{T}}(0) \quad \dot{y}^{\text{T}}(0)]^{\text{T}} = [\tilde{x}^{\text{T}}(t_j) \quad \dot{\tilde{x}}^{\text{T}}(t_j)]^{\text{T}}$. Note that the matrix Λ is preserved in the expression HH^{T}. The entries of the matrix, which is used in (10), have the following values

$$\begin{aligned}A_{1,1} &= 0, \qquad A_{1,2} = I \\ A_{2,1} &= -H^{-\text{T}}\,\text{diag}(v)H^{\text{T}} \\ A_{2,2} &= -2H^{-\text{T}}\,\text{diag}(\xi \odot \sqrt{v})H^{\text{T}}.\end{aligned}$$

The interval $[0,T]$ is split equidistantly into $N \in \mathbb{N}$ subintervals, which are each $\Delta T \coloneqq T/N$ seconds long. The continuous time differential equation (10) together with its initial value can be approximated on the interval $[0,T]$ with an appropriate discretization method, e.g. Euler method or Runge–Kutta methods. This results in the discrete time approximation

$$z_{i+1} = h\left(z_i, \xi_i, v_i, H_i, F_{\text{ext}}\right) \tag{11}$$

for $i = 0, \ldots, N-1$ and with the initial condition $z_0 = z(0) = [\tilde{x}^{\text{T}}(t_j) \quad \dot{\tilde{x}}^{\text{T}}(t_j)]^{\text{T}} \in \mathbb{R}^{2m}$. To obtain the transformation matrix H_i, we can compute the double diagonalization (6) with $K_j^{\text{ref}}(i\Delta T)$ for $i = 0, \ldots, N-1$.

D. Optimization

The derived discrete time system (11) is optimized with respect to the transformed eigenvalues v and damping ratio ξ. The optimization problem is formulated as

$$\begin{aligned}&\underset{\substack{v_0,\ldots,v_{N-1} \\ \xi_0,\ldots,\xi_{N-1}}}{\text{minimize}} \sum_{i=0}^{N-1} \left\|w \odot (v_i - v_i^{\text{ref}})\right\|^2 + \eta\left\|\xi_i - \xi_i^{\text{ref}}\right\|^2 \\ &\text{s.t. } z_0 = \left[\tilde{x}^{\text{T}}(t_j) \quad \dot{\tilde{x}}^{\text{T}}(t_j)\right]^{\text{T}} \qquad (12) \\ &\quad z_{i+1} = h\left(z_i, \xi_i, v_i, H_i, F_{\text{ext}}\right) \text{ for } i = 0, \ldots, N-1 \\ &\quad \left| z_N - \begin{bmatrix} (K_j^{\text{ref}}(N\Delta T))^{-1}F_{\text{ext}} \\ 0 \end{bmatrix} \right| \leq \varepsilon. \qquad (13)\end{aligned}$$

First, seeking for compensation between the different orders of magnitude within the transformed eigenvalues v, which due to experience reach from 15 up to 15.000, a weighting parameter $w \in \mathbb{R}^m$ is added. Furthermore, in the cost function the second weighing parameter $\eta > 0$ balances the different orders of magnitude of transformed stiffness eigenvalues and damping ratio. The damping ratio reference $\xi_i^{\text{ref}} \in \mathbb{R}^m$ is advised to be set to $\alpha[1 \ \ldots \ 1]^{\text{T}}$ with $\alpha \in (0,1]$ for all $i = 0, \ldots N-1$, which ensures a critically damped or underdamped oscillation, respectively.

The terminal constraint (13) is evaluated element-wise with the constraint vector $\varepsilon \in \mathbb{R}^{2m}$. The first m entries of (13) imply that the final position of z_N should be close to the equilibrium position $(K_j^{\text{ref}}(N\Delta T))^{-1}F_{\text{ext}}$ of the mass-spring-damper system. In other words, the initial value z_0

is the current error of the end-effector and the final value must be within a certain boundary. These two boundaries for the system on $[0, T]$ are fixed. In between deviation of the state is admitted and not penalized by the cost function, which retains the compliant behavior of the virtual mass-spring-damper system. The optimizer chooses eigenvalues and damping ratios to fulfill first these two boundaries on the state and second to be as close to the respective references as possible.

The solution of (12) are two minimizing sequences $(v_i^*)_{i=0,\dots,N-1}$ and $(\xi_i^*)_{i=0,\dots,N-1}$. The first element of the sequences is marked with $v^* \coloneqq v_0^* \in \mathbb{R}^m$ and $\xi^* \coloneqq \xi_0^* \in \mathbb{R}^m$.

E. Compute Control Parameter

The low-level impedance controller is executed much faster than the high-level parameter optimizer can find solutions. Therefore, v^* and ξ^* are updated only on the sample points t_j. But, there exists a whole interval $[t_j, t_{j+1}]$ between two sampling points, on which the high-level controller must provide a stiffness matrix K_d and a damping matrix D_d for the compliance control (3). In addition the inertia matrix Λ varies on the interval as well.

Consider the time point $t \in [t_j, t_{j+1}]$. Then we can repeat the double diagonalization (6) for each t. Perform a Cholesky decomposition of $\Lambda(q(t)) = L_t L_t^{\mathrm{T}}$ and compute $[Q_t, v_t] = \operatorname{eig}\left(L_t^{-1} K^{\mathrm{ref}}(t) L_t^{-\mathrm{T}}\right)$ as well as $H_t = L_t Q_t$. Then the stiffness and damping matrix are

$$\begin{aligned} K_d(t) &\coloneqq H_t \operatorname{diag}(v^*) H_t^{\mathrm{T}} \quad \text{and} \\ D_d(t) &\coloneqq 2 H_t \operatorname{diag}(\xi^* \odot \sqrt{v^*}) H_t^{\mathrm{T}}, \end{aligned} \tag{14}$$

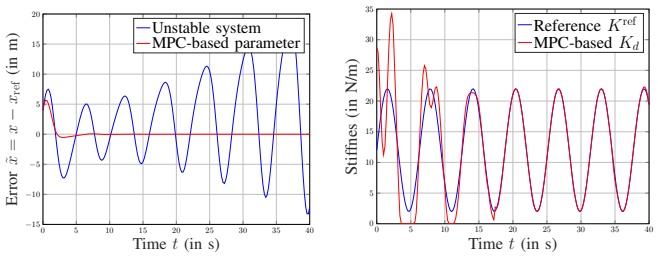
for $t \in [t_j, t_{j+1}]$, which are then applied in the compliance control input (3).

Remark 1. *The terminal constraint* (13) *is motivated by the common approach to ensure stability of MPC algorithms via equilibrium endpoint constraint [18]. According to the MPC literature, a MPC algorithm with a feasible equilibrium endpoint constraint $z_N = z^*$ is asymptotically stable. The mass-spring-damper system, however, is not controllable in finite time to the equilibrium point $z^* = 0$, i.e. the terminal constraint $z_N = 0$ is not feasible. Notice that in our case a constant impedance parameter setting would ensure stability. Therefore, we believe that the constraint from [18] is too conservative in our application. However, a detailed stability analysis is part of future work.*

Remark 2. *Due to the fact that the future course of the external force F_{ext} is unknown, we have assumed a constant disturbance model. Furthermore, we assume a constant equilibrium of the mass-spring-damper system. Since the system is time-variant, this assumption induces a small error.*

IV. Simulations and Experiment

The performance of our MPC-based parameter optimization algorithm has been validated in two simulations and one experiment. The nonlinear optimization problem (12) is solved using the open-source symbolic framework CasADi [19] in combination with the software packages for large-scale nonlinear optimization Ipopt [20].



(a) The position error $\tilde{x}$

(b) The utilized stiffness for the mass-spring-damper system

Fig. 2: First Simulation - Comparison of the natural and optimized system parameters

A. Application on an Unstable System

As a first benchmark example, our proposed method has been tested on a single translational joint. Therefore, we have reproduced the example from [7]. Consider the given properties of the desired mass-spring-damper system

$$\begin{aligned} M &= 10, \\ D_d &= 1 \quad \text{and} \\ K_d(t) &= 12 + 10\sin(t), \end{aligned} \tag{15}$$

while the system has the initial values $x(0) = 4$, $\dot{x}(0) = 9$ and should follow the trajectory $x_{\mathrm{ref}}(t) = 10\sin(t/10)$. Further, no external force acts on the system, i.e. $F_{\mathrm{ext}} = 0$. First, the impedance control law (3) from section II-B is applied. The blue graph in Fig. 2(a) shows the course of the position error $\tilde{x}$ between x and the reference trajectory x_{ref}, which clearly diverges.

Next, we evaluate the performance of our algorithm. For this purpose, the stiffness in (15) is considered as reference stiffness $K^{\mathrm{ref}}(t)$, which is pictured as blue graph in Fig. 2(b). Further the prediction horizon is set to $T = 10\,\mathrm{s}$, the amount of discrete time system points equals $N = 100$ and the optimization is executed every $50\,\mathrm{ms}$. Since the system has one eigenvalue, w is not required and set to one. In addition, the weighting parameter η is chosen to be 20 because the orders of magnitude are close, but at the same time we want to penalize deviation in the damping ratio more severely. Its reference is based on the damping ratio in [7], i.e. $\xi_i^{\mathrm{ref}} = 0.5 D_d / \sqrt{M K_d(0)} \approx 0.045$ for all $i = 0, \dots, N-1$. At least, the terminal boundary is set to $\varepsilon = 10^{-3}[1\ 1]^{\mathrm{T}}$.

As expected, our proposed method can handle the varying target stiffness in combination with a small reference damping ratio and also causes the position error, visible in Fig. 2(a), to converge to zero, i.e. stabilizes the system. Moreover, the applied optimal stiffness K_d, generated by the algorithm, coincide with the reference stiffness after an initial deviation during transient conversion of the position error.

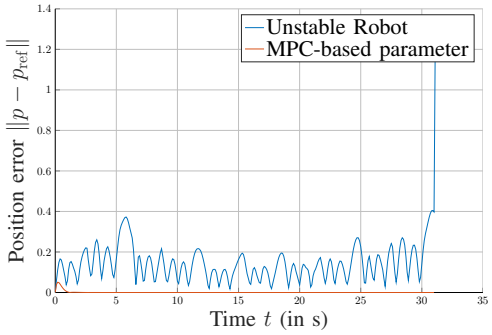


Fig. 3: Second Simulation - The absolute position error of the 7-Dof robot end-effector

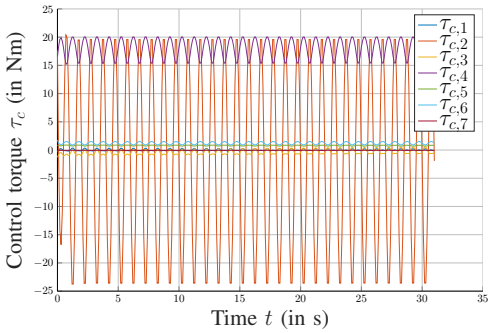


Fig. 4: Second Simulation - Produced control torque τ_c from the algorithm

B. Simulated Unstable Robot

We applied the controller in simulation to a $n = 7$-DoF Franka Research 3 robotic manipulator. In the $m = 6$ dimensional task space, the configuration of the end-effector is described by $x(t) = [p^{\mathrm{T}}(t) \quad \phi^{\mathrm{T}}(t)]^{\mathrm{T}}$, with the position $p(t) \in \mathbb{R}^3$ and the orientation $\phi(t) \in \mathbb{R}^3$ in roll-pitch-yaw angles. In the same structure, the reference trajectory $x_{\text{ref}}(t) \in \mathbb{R}^6$ is given. The reference has a constant Cartesian configuration except for the first coordinate, which oscillates according to $x_{\text{ref},1}(t) = 0.3 + 0.1\sin(2\pi t)$. The properties of the mass-spring-damper system are given as $D_d = I$ and $K_d(t) = \text{diag}(100 + 75\sin(2t), 200, 200, 50, 50, 50)$, while the system is in its initial configuration at rest. Similar to the one-Dof simulation, no external force $F_{\text{ext}} = 0$ is applied. With these quantities the impedance control law (3) is applied on the robot. The blue graph in Fig. 3 shows the absolute error in the position of the end-effector, which is strongly oscillating. After approximately 31 seconds, the simulation terminates, since the robot has reached its outstretched singular position.

For our proposed algorithm, K_d is again considered as K^{ref}. The prediction horizon is set to $T = 1\,\text{s}$, the amount of discrete time system points equals $N = 26$ and the optimization is executed every 80 ms. In order to cope with the computation cost for the optimization of the 7-DoF robot, the prediction horizon as well as the amount of discrete time points are reduced. The first weighting parameter is $w = [0.01,\ 0.1,\ 0.1,\ 1,\ 1,\ 1]^{\mathrm{T}}$ and the second is set to $\eta = 1400$,

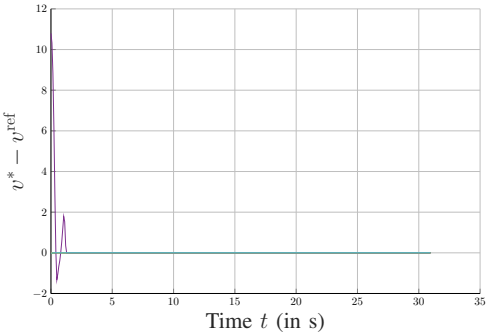


Fig. 5: Second Simulation - Difference between optimized eigenvalues v^* and computed reference v^{ref}

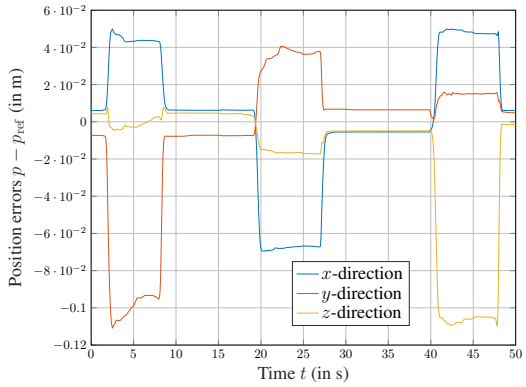
in order to normalize the effect of stiffness and damping. Further, the damping ratio is $\xi_i^{\text{ref}} = \sqrt{2}/2\,[1 \ \ldots \ 1]^{\mathrm{T}}$ for all $i = 0, \ldots, N-1$. The terminal boundary is set to $\varepsilon = 10^{-3}[1 \ \ldots \ 1]^{\mathrm{T}}$.

The red graph in Fig. 3 illustrates the systems behavior with the MPC-based parameter optimization. Like in the previous simulation, our algorithm can handle the time-varying stiffness and drains the energy out of the system, which causes the error to converge to zero. With Fig. 4 we can verify that all control torques, which are generated by our algorithm, are all smooth and bounded. In Fig. 5 the deviation of the optimized eigenvalues v^* from the reference eigenvalues v^{ref} is shown. This figure illustrates that adjustments of the eigenvalues are only necessary at the beginning and they coincide after a short time.

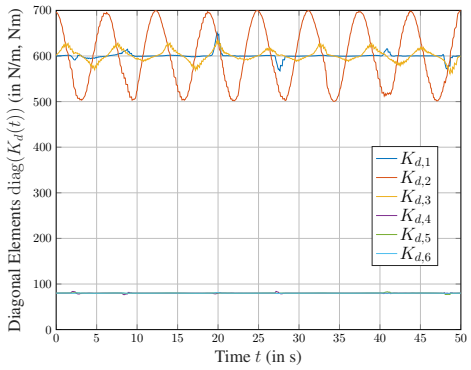
C. Experimental Validation

In the two previous scenarios, our approach has only been validated in simulations. Finally, it has been be used for physical interaction on a real 7-DoF Franka Research 3 robotic manipulator. The end-effector is described by $x(t) = [p^{\mathrm{T}}(t) \quad \phi^{\mathrm{T}}(t)]^{\mathrm{T}}$, exactly as in the last case. The Cartesian reference $x_{\text{ref}} \in \mathbb{R}^6$ is a constant configuration. The reference stiffness varies over time with $K^{\text{ref}}(t) = \text{diag}(600, 600{+}100\sin(t), 600, 80, 80, 80)$. The optimization parameters retain their values from the previous simulation, except the amount of discrete time points, which is increased to $N = 30$. Tests have shown that the dynamics can be better reflected with a few more time points. After 1, 19 and 40 seconds forces have been applied on the end-effector in several directions and have been hold for approximately 9 seconds.

Since each optimization usually took 80ms to be fully solved, we have sampled at a frequency of 100Hz to leave a buffer for any potential outliers. The Cartesian position error of the end-effector is depict in Fig. 6(a). Here, the effects of the applied forces are clearly visible as distinct displacements. The slight deviation in the position when no force is applied is a natural phenomenon in impedance control, caused by friction. Our proposed algorithm generates an optimal stiffness matrix K_d. We observed that its values are close to the given reference stiffness matrix K^{ref} and



(a) Error in the position of the end-effector due to multiple applied forces



(b) Obtained diagonal elements of the optimized stiffness matrix K_d

Fig. 6: Results of the Experiment - Characteristic course of parameters of the Franka Research 3 robot due to an external applied force

it has negligible off-diagonal elements. Therefore, Fig. 6(b) shows only the diagonal elements of the optimal stiffness matrix K_d. The elements associated with rotation, i.e. $K_{d,4}$, $K_{d,5}$ and $K_{d,6}$, maintain their desired values of 80 over time. While $K_{d,2}$ almost approximately follows its sinusoidal reference and does not deviate from its shape despite the presence of a force, $K_{d,3}$ shows an oscillating behavior.

V. Conclusion and Outlook

In this paper we have presented a MPC-based strategy to handle variable impedance parameters. These are obtained as the result of an optimization of a virtual mass-spring-damper system. Our proposed algorithm has been validated in several simulations as well as a hands-on experiment and has met our expectations.

In future work, the algorithm will be examined in a detailed proof of stability. Moreover, we will work on further increasing the calculation speed of the optimization. One option is to exploit the linearity of the virtual mass-spring-damper system and incorporate it into the optimization procedure.

Acknowledgment

We thank Andreas Kugi for his productive discussions and his insights into the design of nonlinear model predictive controllers.

References

[1] N. Hogan, "Impedance Control: An Approach to Manipulation," in *1984 American Control Conference*, Jun. 1984, pp. 304–313.

[2] M. Goodrich and A. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, Jan. 2007.

[3] S. Haddadin, "Physical Safety in Robotics," in *Formal Modeling and Verification of Cyber-Physical Systems*, R. Drechsler and U. Kühne, Eds. Wiesbaden: Springer Fachmedien, 2015, pp. 249–271.

[4] A. Bicchi and G. Tonietti, "Fast and "soft-arm" tactics: dealing with the safety-performance tradeoff in robot arms design and control," *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 22–33, Jun. 2004.

[5] J. Buchli, E. Theodorou, F. Stulp, and S. Schaal, "Variable Impedance Control A Reinforcement Learning Approach," Jul. 2010.

[6] K. Kronander and A. Billard, "Stability Considerations for Variable Impedance Control," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1298–1305, Oct. 2016.

[7] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4948–4953, iSSN: 1050-4729.

[8] Y. Michel, C. Ott, and D. Lee, "Safety-aware hierarchical passivity-based variable compliance control for redundant manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3899–3916, 2022.

[9] R. Johansson and M. Spong, "Quadratic optimization of impedance control," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 616–621 vol.1.

[10] M. Bednarczyk, H. Omran, and B. Bayle, "Model Predictive Impedance Control," May 2020, pp. 4702–4708.

[11] Z. Jin, D. Qin, A. Liu, W.-a. Zhang, and L. Yu, "Model Predictive Variable Impedance Control of Manipulators for Adaptive Precision-Compliance Tradeoff," *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 1174–1186, Apr. 2023.

[12] L. Roveda, A. Testa, A. A. Shahid, F. Braghin, and D. Piga, "Q-Learning-based model predictive variable impedance control for physical human-robot collaboration," *Artificial Intelligence*, vol. 312, p. 103771, Nov. 2022.

[13] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. Molinari Tosatti, and N. Pedrocchi, "Model-Based Reinforcement Learning Variable Impedance Control for Human-Robot Collaboration," *Journal of Intelligent and Robotic Systems*, Nov. 2020.

[14] A. S. Anand, F. J. Abu-Dakka, and J. T. Gravdahl, "Deep Model Predictive Variable Impedance Control," Sep. 2022.

[15] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics : modelling, planning and contro*, ser. Advanced Textbooks in Control and Signal Processing. London: Springer, 2009.

[16] A. Dietrich, X. Wu, K. Bussmann, M. Harder, M. Iskandar, J. Englsberger, C. Ott, and A. Albu-Schäffer, "Practical consequences of inertia shaping for interaction and tracking in robot control," *Control Engineering Practice*, vol. 114, p. 104875, 2021.

[17] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms," in *2003 IEEE International Conference on Robotics and Automation*, vol. 3, Sep. 2003, pp. 3704–3709 vol.3.

[18] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*, ser. Communications and Control Engineering. Springer International Publishing, 2017.

[19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, mar 2019.

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.