Frequency-domain data-driven position-dependent controller synthesis for Cartesian Robots

Philippe Schuchert, Alireza Karimi, *Member, IEEE*

***Abstract*—Cartesian robots have position-dependent dynamics that must be taken into account for high-performance applications. Traditional methods design Linear Time-Invariant (LTI) controllers that are robustly stable with respect to position variations, but result in reduced performance. Advanced methods require Linear Parameter Varying (LPV) models and LPV controller design methods that are not well-established in the industry. On the other hand, the classical model-based gain-scheduled technique involves parametric identification, high-performance controller design for each position, interpolation of the controller parameters, and real-time controller validation, making it time-consuming and costly. Our approach uses frequency response at different operating points to design an LPV controller using a convex optimization algorithm based on second-order cone programming. The approach is applied to an industrial 3-axis Cartesian robot, showing significant improvements over state-of-the-art control design strategies. Data acquisition and controller design can be performed automatically, reducing significantly the engineering costs for controller synthesis.**

***Index Terms*—Positioning systems, LPV control, Frequency domain synthesis, data-driven control, $\mathcal{H}_2 - \mathcal{H}_\infty$ control**

I. Introduction

CARTESIAN robots have found a broad range of applications in the manufacturing process, e.g., pick-and-place, material handling, coordinate measuring, or CNC milling. Each axis of the robot can move in an orthogonal direction to the other axes and can be controlled using multiple single-input single-output (SISO) controllers. When the robot moves in the operating space, beam lengths and mass distributions are changing, resulting in a series of varying resonant modes. This makes the control design a challenging task, especially when a bandwidth above the first resonant mode is required [1]. The classical approach is to use a *simple* controller (e.g., PID with filters, or more recent approaches such as $\mathcal{H}_\infty$ or μ-synthesis, see [2], [3], [4]) such that the whole operating space is simultaneously stabilized. This approach will result in an inherent trade-off in performance [5]. A single linear control law can only achieve sup-optimal performance locally. The same controller must stabilize the dynamics at the different operating points, resulting in a robustness-performance trade-off.

In applications such as wafer inspection and coordinate measuring, high precision and large closed-loop bandwidth are required to obtain a competitive advantage. Ever-increasing demands for faster and more accurate robots are pushing traditional control strategies to their limits, sometimes resulting in impossible to meet design requirements. In many applications, control design schemes incorporating position-dependent gains [6] have been used to overcome this issue [7]. In *Divide-and-conquer* gain-scheduling, controllers are tuned around different operating points, and later blended into a single non-linear control law, e.g., in [8], [9], [10], [11]. Tuning and blending numerous controllers to achieve the desired performance is a time-consuming task and requires expertise from a skilled engineer. The effort scales with the number of operating points considered, as around each operating point, a model must first be identified, then validated, and finally an appropriate local controller designed. For Cartesian robots, the whole process must be repeated for each axis. This interpolated control law can have suboptimal performance when fitting a polynomial on the controller coefficients, and excessive computation time when interpolating directly on a real-time target.

LPV controller synthesis approach overcomes this issue by designing tractable, position-dependent controllers. This approach results in improved performance over traditional LTI controllers, with advantageous guarantees over gain scheduling. To achieve at the same time robustness and performance, an accurate LPV model is paramount to a good control design. Obtaining such a model is difficult and expensive [12]. If a machine is subject to any change during its life, e.g., wear and tear (aging of the machine), changing the end effector tool, or replacing parts of the mechanical assembly or power stage, the costly LPV modeling and identification phase must be done again.

Frequency response functions (FRFs) have been proven to be an effective way of representing mechanical systems with many resonant modes [13]. It can be obtained directly from input-output data, is well understood, and is extensively used in industry. Frequency-based approaches remove the costly parametric identification phases (structure identification, parameter estimation, and model validation). For position-dependent systems, obtaining the FRF at different operating

Manuscript submitted for review on March 10, 2023. This work has been financed under the grant INNOSUISSE 30595.1 IP-ENG. *(Corresponding author: Alireza Karimi.)*

Philippe Schuchert and Alireza Karimi are with the Institute of Mechanical Engineering, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: philippe.schuchert@epfl.ch; alireza.karimi@epfl.ch).

points can be used as an accurate representation of the system's local dynamics. Different FRF-based controller synthesis approaches have been extended to position-dependent or LPV systems [14], [15], [16], [17]. They offer a natural way of handling frequency-based design specifications but are limited in flexibility. Notably, none of the aforementioned approaches can handle simultaneously $\mathcal{H}_2$ and $\mathcal{H}_\infty$ synthesis requirements.

The approach in [18] addresses mixed-objective design requirements using only the frequency response of an LTI system, resulting in a Semi-Infinite Programming (SIP) problem that is convex in the controller parameters and solved using a series of Linear Matrix Inequalities (LMIs). This paper extends the results of [18] to parameter-dependent controller design, improving the formulation for SISO controllers using Second-Order Conic Programming (SOCP) and addressing practical issues related to frequency gridding for a more reliable synthesis.

The main contributions of this paper are summarized as follows:

1) *Extension to position-dependent systems*. The original problem is reformulated using second-order conic constraints, making it tractable to consider for a larger set of models and controller order as well as denser frequency gridding. This formulation enables the design of high-order position-dependent controllers involving tens to hundreds of thousands of second-order conic constraints using conventional personal computers.
2) *Application to an industrial 3D positioning system*. The proposed approach has been applied to an industrial Cartesian robot with three orthogonal axes.

This paper presents only the discrete-time SISO controller synthesis, as the positioning system used is identified from input/output data obtained at discrete time steps. It is directly applicable to continuous-time systems by changing the Nyquist contour and the controller parametrization if a continuous-time controller is sought.

This paper is organized as follows. Section II addresses the notation used throughout this paper and gives a short overview of a related work. Section III deals with the problem statement and the description of the system. Section IV presents the development of a frequency-based position-dependent controller synthesis method and discusses mixed-sensitivity $\mathcal{H}_2/\mathcal{H}_\infty$ hard and soft requirements. Section V addresses some practical issues and proposes mitigation steps. Section VI shows the experimental results for an industrial positioning system.

II. Preliminaries

A. Notations

Throughout this paper, the imaginary unit is denoted by $j = \sqrt{-1}$. The principal argument of a complex-valued scalar z is represented by Arg $\{z\} \in (-\pi, \pi]$. The Hermitian transpose of a complex-valued vector F is denoted by F^* and its Euclidean norm by $|F| = \sqrt{F^*F}$. The $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm of an LTI system are expressed as $\|\cdot\|_2$ and $\|\cdot\|_\infty$, respectively.

B. Related work

In [18], the problem of mixed-sensitivity synthesis using only the frequency-domain data is considered. Given the frequency response of a plant model $G(e^{j\omega})$ and a controller with fixed structure $K = XY^{-1}$ for some polynomial matrices $X(z)$ and $Y(z)$, the design objective is to minimize the norm of the mixed-sensitivity problem. We will take as an illustrative example the $\mathcal{H}_\infty$ norm case:

$$\min_K \left\| \begin{bmatrix} W_1 \mathcal{S} \\ W_2 \mathcal{U} \end{bmatrix} \right\|_\infty \tag{1}$$

where W_1, W_2 are weighting filters, $\mathcal{S} = (I + GK)^{-1}$ the sensitivity, and $\mathcal{U} = K\mathcal{S}$ the input sensitivity function [19]. Unlike traditional methods, only the FRF of the plant G is required, along with a known initial stabilizing controller $K_c = X_c Y_c^{-1}$. The problem is reformulated as the minimization of an upper-bound of the mixed-sensitivity norm:

$$\min_{X,Y} \gamma$$

subject to

$$\begin{bmatrix} P^*P_c + P_c^*P - P_c^*P_c & (W_1Y)^* & (W_2X)^* \\ W_1Y & \gamma I & 0 \\ W_2X & 0 & \gamma I \end{bmatrix} \succ 0 \quad \forall \omega \in \Omega \tag{2}$$

where $\Omega = (-\pi/T_s\,, \pi/T_s]$ with T_s the sampling period and

$$P = Y + GX, \quad P_c = Y_c + GX_c.$$

Minimizing the upper-bound γ subject to (2) also ensures that the closed-loop system remains stable. The main stability theorem states:

Theorem 1: Given a plant model G, an initial stabilizing controller $K_c = X_c Y_c^{-1}$ with $\det(Y_c) \neq 0$, $\forall \omega \in \Omega$, and feasible solutions X and Y to the following LMI:

$$P^*P_c + P_c^*P \succ 0 \quad \forall \omega \in \Omega, \tag{3}$$

then the controller $K = XY^{-1}$ stabilizes the closed-loop system if

(1) $\det(Y) \neq 0$, $\forall \omega \in \Omega$.
(2) The initial controller K_c and the final controller K share the same poles on the stability boundary.
(3) The order of $\det(Y)$ is equal to the order of $\det(Y_c)$.

Proof See [18]. ■

The stability constraint (3) is always enforced in (2) as the first minor of the matrix must also be positive. The LMI in (2) is a function of ω and must be satisfied for the whole set Ω but, in practice, is checked only on a finite set of sampled frequency points. In the case of multimodel uncertainty, i.e. $G \in \{G_1, ..., G_N\}$, then (2) must be satisfied for each model, which results in a large number of LMI constraints, making the solution potentially computationally expensive.

III. Description of the system

A schematic representation[1] of a 3D Cartesian robot is shown in Fig. 1. Each axis can move in an orthogonal direction

[1] Due to confidentiality agreements, pictures of the machine used in Sec. VI for the experimental results cannot be shown.

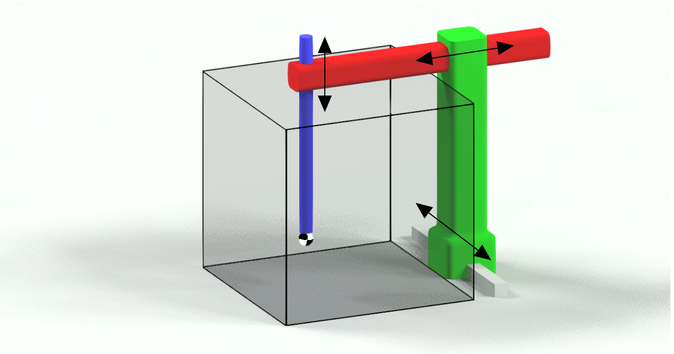


Fig. 1: Schematic representation of a 3D Cartesian robot. x-axis in green, y-axis in red, z-axis in blue. Each axis can move in an orthogonal direction to the other axes. The operating point, corresponding to the black and white sphere, is located at the bottom of the z-axis. The operating space is indicated using the transparent box.

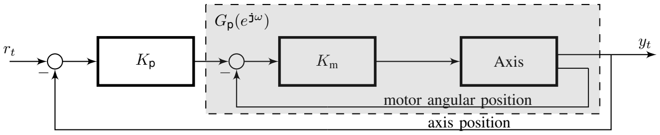


Fig. 2: Block diagram of individual axis loop.

to the other axes. The $(\mathsf{x}, \mathsf{y}, \mathsf{z})$ position of the end-effector is denoted p, and restricted to some operating space $\mathbb{P}$:

$$\mathsf{p} \in \mathbb{P} \subset \mathbb{R}^3. \tag{4}$$

Depending on the operating point, the mass distribution and beam lengths of the axes can vary significantly, leading to position-dependent dynamics. Such effects are often explicitly ignored but must be accounted for when a large closed-loop bandwidth is required.

For the 3D positioning robot considered in Sec. VI, each axis consists of a DC motor with a reduction stage and a belt pulley transmission, linking the motor to its corresponding axis. Encoders are available to precisely measure the axis position and motor angular position. The controlled inputs are the voltages applied to the DC motors. In industry, a common control scheme for such systems is cascaded control loops, where K_m, the inner controller, is regulating the motor angular position, and K_p, the outer-loop controller, the axis position. We will adopt the same control architecture and assume the inner loops have already been tuned. Satisfactory performance for the inner loops can often be obtained without the need for position-dependent controllers, and thus their tuning is not discussed here. The model of each axis with the closed inner-loop is denoted G_p. A block diagram representing this interconnection can be found in Fig. 2.

This paper focuses only on tuning K_p for the SISO model G_p, where the flexible-body and position-dependent dynamics are most prevalent. For brevity, only the x-axis tuning process is discussed in detail. The two other axes have less pronounced position-dependent dynamics and follow the same tuning approach, and thus only discussed briefly.

Around every operating point, each axis can be modeled using an LTI model. This motivates the use of Linear Parameter Varying (LPV) models for position-dependent dynamics. A discrete-time LPV model is defined as:

$$\begin{aligned} \zeta_{t+1} &= A_\mathsf{p}\zeta_t + B_\mathsf{p}u_t \\ y_t &= C_\mathsf{p}\zeta_t + D_\mathsf{p}u_t \end{aligned} \tag{5}$$

where $A_\mathsf{p}, B_\mathsf{p}, C_\mathsf{p}, D_\mathsf{p}$ are the state-space matrices depending on the operating point, ζ_t the internal states, u_t the input, and y_t measurement at sample t. As the state-space matrices depend on the robot's states, this sort of model is called a quasi-LPV model [20]. Obtaining the state-space description is often a cumbersome and time-consuming task, especially with systems exhibiting many resonant modes.

Assumption 1: The frozen dynamics representation at each operating point remains a valid description of the system.

Remark: Assumption 1 may seem very restrictive, as the dynamics are a function of the operating points. For the positioning system considered, the dynamics vary significantly only over large distances in the operating space. Physical limitations will put a limit at which the operating space can be traversed, in turn limiting the rate of change of the dynamics. A closed-loop bandwidth much larger than this rate of change makes *Assumption 1* justifiable. For the system considered in Sec. VI, experimental results justify this assumption as well.

Under *Assumption 1*, at every operating point, a unique frequency response function G_p exists, which can be used to model the (frozen) dynamics [20]:

$$G_\mathsf{p}(e^{j\omega}) = C_\mathsf{p}\left(e^{j\omega}I - A_\mathsf{p}\right)^{-1} B_\mathsf{p} + D_\mathsf{p}, \tag{6}$$

with $\omega \in \Omega := (-\pi, \pi]$. To protect the commercial interests of our industrial partner, the frequency range is normalized such that the Nyquist frequency corresponds to $\omega = \pi$. The pragmatic approach to obtain (6) is to identify the FRF around different operating points using a local input/output data set, avoiding the need to explicitly find the state-space representation. The FRF can be obtained using a wealth of existing and well-understood techniques [21], [13], or more recent approaches focusing on LPV FRF identification [22], [23].

Multiple local data sets are collected for each axis. Each data set consists of $T = 8000$ input and output measurements. The input r is a *sum-of-sines* signal with random phase given by:

$$r_t = \sum_{k=1}^{T/2} \left(\alpha + \frac{\beta}{k}\right) \sin(2\pi k(t - \tau_k)/T)$$

where τ_k is a random integer in $\{0, \ldots, T-1\}$. The constant α is chosen sufficiently large to excite well all frequencies, but not too large to saturate the input of the DC motor. The constant β is chosen to add additional excitation in low frequencies. For every axis, the local FRF $G_\mathsf{p}(e^{j\omega})$ is computed using spectral analysis [21]. The local input-output data is first converted to the frequency domain, then averaged to obtain an estimate at 350 logarithmically spaced frequencies.

For each axis, a different number of models is required to describe well the change in dynamics w.r.t. the operating points. For the x-axis, $5\times5\times4 = 100$ models are obtained on a regular grid in the operating space: the x position is sampled at 5 equidistant points, the y position at 5 equidistant points, and the z position at 4 equidistant points. For the y-axis dynamics, there is only a negligible dependency on the x position: the y-axis is mounted on top of the x-axis as illustrated in Fig. 1, and the position of the x-axis will not have a significant impact on the y-axis dynamics. This has been confirmed empirically. Therefore, a reduced number of models for the y-axis can be chosen: a total of $1 \times 8 \times 8 = 64$ models on a regular grid in the y-z plane, obtained at a fixed x position. Similarly, the z-axis is mounted on top of the y-axis, and its dynamics do not depend on the two previous axes. Therefore, it has been noted to be sufficient to use $1 \times 1 \times 8 = 8$ different models, obtained at equidistant heights z, to describe well the dynamics in the whole operating space.

The identified spectra for the different axes are shown in Fig. 3–5. the level of green, red, or blue color depends on the $(\mathsf{x}, \mathsf{y}, \mathsf{z})$ coordinates of the operating point. The changing color highlights the position dependency in the dynamics: a change in the level of green indicates a change w.r.t the x position, a change in the level of red indicates a change w.r.t the y position, and a change in the level of blue indicates a change w.r.t the z position. For the x-axis (Fig. 3), the color of the first resonance mode shifts from cyan (blue and green) to blue, highlighting a strong dependency of this resonant mode with the x position. The color of the second resonance mode shifts from magenta (blue and red) to blue, indicating a strong dependency on the y position. For the y-axis (Fig. 4), the cyan-to-blue color transition indicates a strong dependency in the dynamics on the y position.

These varying resonant modes depend on the mechanical structure of the machine, the inner-loop controllers, and many other difficult-to-model interactions. Such varying modes can drastically limit the performance of traditional control approaches and are especially difficult to model using parametric approaches.

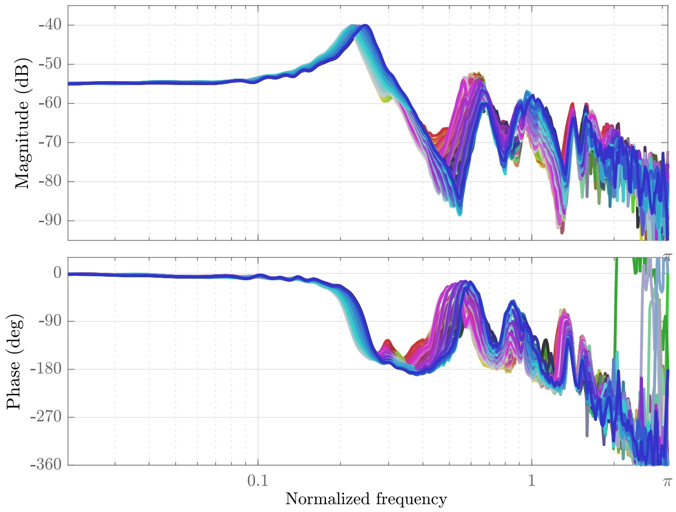


Fig. 3: x-axis. Identified FRF of the position-dependent dynamics around different operating points $(\mathsf{x}, \mathsf{y}, \mathsf{z})$. The FRF is color-coded at each frozen operating point as green, red, or blue based on the position of the operating point. A change in resonance mode can be observed near frequencies 0.2 and 0.6.

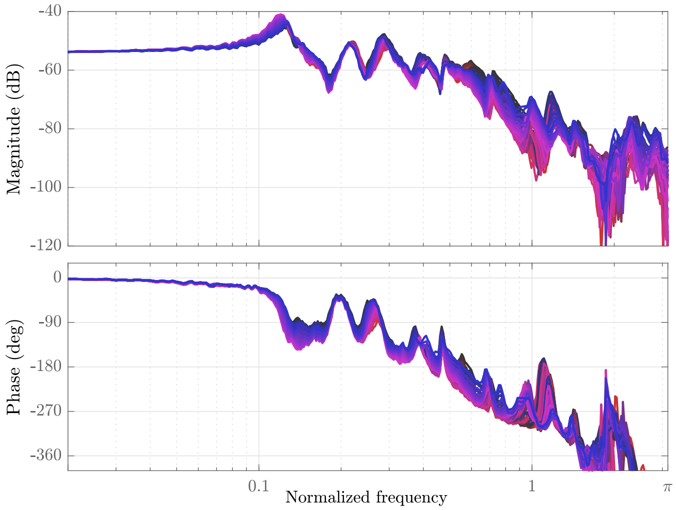


Fig. 4: y-axis. Identified FRF of the position-dependent model around different frozen operating points. A change in resonance mode can be observed shortly after frequency 0.1.

IV. Position-dependent controller synthesis

A. Control structure and performance

To achieve the desired performance, the axis controller K_{p} will be dependent on the operating point:

$$K_{\mathsf{p}} = \frac{X_{\mathsf{p}}}{Y_{\mathsf{p}}} = \frac{X_{\text{var}}(z, \mathsf{p})}{Y_{\text{var}}(z, \mathsf{p})} \frac{F_X(z, \mathsf{p})}{F_Y(z, \mathsf{p})}, \tag{7}$$

where z is the z-transform variable, and F_X and F_Y are fixed parts in the controller. $X_{\text{var}}(z, \mathsf{p})$ and $Y_{\text{var}}(z, \mathsf{p})$ are the variable parts in the controller, defined as:

$$\begin{aligned} X_{\text{var}}(z, \mathsf{p}) &= X_a(\mathsf{p}) z^a + X_{a-1}(\mathsf{p}) z^{a-1} + \ldots + X_0(\mathsf{p}), \\ Y_{\text{var}}(z, \mathsf{p}) &= \qquad\quad z^b + Y_{b-1}(\mathsf{p}) z^{b-1} + \ldots + Y_0(\mathsf{p}). \end{aligned} \tag{8}$$

The variable parts have order a and b respectively, such that K_{p} is proper. Y_{var} should not have any zeros on the unit circle, and therefore F_Y must at least include all the controller's poles on the unit circle. $X_i(\mathsf{p})$ and $Y_i(\mathsf{p})$ are affine combinations of a pre-defined scheduling vector $\theta(\mathsf{p})$ of length n_θ:

$$X_i(\mathsf{p}) = \sum_{k=0}^{n_\theta} x_{ik} \theta_k(\mathsf{p}), \quad Y_i(\mathsf{p}) = \sum_{k=0}^{n_\theta} y_{ik} \theta_k(\mathsf{p}). \tag{9}$$

x_{ik}, y_{ik} are scalar optimizations variables, and the scheduling vector $\theta(\mathsf{p}) = [\theta_0(\mathsf{p}), \ldots, \theta_{n_\theta}(\mathsf{p})]$ is a (possibly non-linear) function of the operating point p.

The control synthesis problem is formulated as a mixed-sensitivity problem with soft and hard requirements:

$$\begin{aligned} &\min_{K_{\mathsf{p}}} \gamma \\ &\text{subject to} \\ &\|R_{\text{soft}}(G_{\mathsf{p}}, K_{\mathsf{p}})\|_{2,\infty} < \gamma, \quad \|R_{\text{hard}}(G_{\mathsf{p}}, K_{\mathsf{p}})\|_{2,\infty} < 1. \end{aligned} \tag{10}$$

R_{soft} corresponds to soft requirements, i.e., the objective(s) to be minimized. R_{hard} corresponds to hard requirements, i.e.,

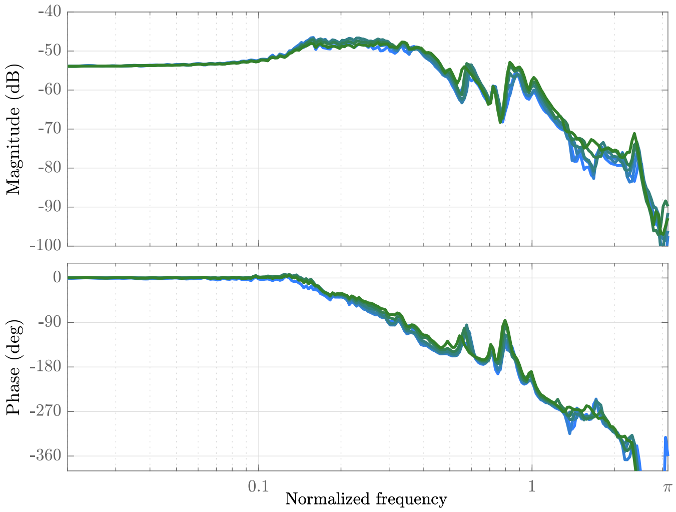


Fig. 5: z-axis. Identified FRFs show the dynamics at different frozen operating points. Unlike the other two axes, the dynamics for this axis have little position dependency and can be controlled effectively with conventional control techniques.

the constraints that must be satisfied irrespectively of the norm achieved by the soft requirements, and given by design specifications (e.g., modulus margin, attenuation of the control input in a certain frequency range, etc.).

B. Second-order cone constraints

We consider the synthesis approach for the control performance as described in (10), where every requirement can be formulated as the weighted norm of the closed-loop sensitivity functions. It is shown in this section that the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ mixed-sensitivity framework can be reformulated as constraints in the frequency-domain with the following structure:

$$|F|^2 < \gamma P^* P, \ \gamma > 0, \tag{11}$$

where $F \in \mathbb{C}^n$ and $P \in \mathbb{C}$ are linear in the optimization variables. This type of constraint is not convex but can be convexified around an arbitrary $P_c \in \mathbb{C}$ using the following inequality:

$$P^* P \geq P^* P_c + P_c^* P - P_c^* P_c, \tag{12}$$

which can be shown by developing the inequality $(P - P_c)^* (P - P_c) \geq 0$. An inner approximation of (11) can be obtained using (12):

$$|F|^2 < \gamma \left(P^* P_c + P_c^* P - P_c^* P_c \right). \tag{13}$$

To keep the notation short, we will use

$$\Phi = P^* P_c + P_c^* P - P_c^* P_c. \tag{14}$$

Note that (13) has a solution if and only if $\Phi > 0$, as by assumption $\gamma > 0$. Equation (13) can be implemented using SOCP, contrasting with semi-definite programming (SDP) required for the more general LMI formulation used in the MIMO case. The second-order rotated cone $\mathcal{C}_r$ is defined as $\mathcal{C}_r = \left\{ (F, \gamma, \Phi) \ | \ |F|^2 < \gamma \Phi, \ \gamma, \, \Phi > 0 \right\}$ and can be embedded in the cone of positive semi-definite matrices:

$$\left\{ \begin{array}{c} |F|^2 < \gamma \Phi \\ \gamma, \, \Phi > 0 \end{array} \right. \iff \begin{bmatrix} \Phi & F^\top \\ F & \gamma I \end{bmatrix} \succ 0. \tag{15}$$

Although SOCP can be solved using generic SDP solvers, the primal-dual interior-point algorithms tailored specifically for SOCP will outperform SDP solvers. Due to a large number of constraints and design variables considered, the SOCP formulation results in a significant reduction in optimization time, often an order of magnitude faster, and offers better numerical conditioning. The constraint (13) corresponds to a second-order cone constraint with complex-valued coefficients in F, but can be replaced with $\tilde{F} = \begin{bmatrix} \Re(F) \\ \Im(F) \end{bmatrix}$, where $\Re(F)$ and $\Im(F)$ are the real and imaginary parts of F.

C. Soft $\mathcal{H}_\infty$ requirements

In this section, performance defined as the infinity-norm of the weighted sensitivity functions is considered:

$$\begin{aligned} & \min_{K_\mathsf{p}} \gamma \\ & \text{subject to} \\ & \left\| \begin{bmatrix} W_1 \mathcal{S}_\mathsf{p} \\ W_2 \mathcal{U}_\mathsf{p} \\ W_3 \mathcal{T}_\mathsf{p} \end{bmatrix} \right\|_\infty^2 < \gamma \qquad \forall \mathsf{p} \in \mathbb{P} \end{aligned} \tag{16}$$

where γ is an upper-bound on the mixed-sensitivity problem and W_1, W_2, W_3 are (not necessarily continuous) weighting functions. Using the parametrization of K_p given in (7), the different closed-loop sensitivity functions are defined as:

$$\begin{aligned} \mathcal{S}_\mathsf{p} &= (1 + G_\mathsf{p} K_\mathsf{p})^{-1} = Y_\mathsf{p} (Y_\mathsf{p} + G_\mathsf{p} X_\mathsf{p})^{-1} \\ \mathcal{U}_\mathsf{p} &= K_\mathsf{p} (1 + G_\mathsf{p} K_\mathsf{p})^{-1} = X_\mathsf{p} (Y_\mathsf{p} + G_\mathsf{p} X_\mathsf{p})^{-1} \\ \mathcal{T}_\mathsf{p} &= G_\mathsf{p} K_\mathsf{p} (1 + G_\mathsf{p} K_\mathsf{p})^{-1} = G_\mathsf{p} X_\mathsf{p} (Y_\mathsf{p} + G_\mathsf{p} X_\mathsf{p})^{-1} \end{aligned}$$

We will use the shorthand notation

$$P_\mathsf{p} = Y_\mathsf{p} + G_\mathsf{p} X_\mathsf{p} \tag{17}$$

for the common denominator of the sensitivity functions.

The $\mathcal{H}_\infty$ norm corresponds to the supremum of the singular value of the vector of weighted sensitivity functions over $\omega \in \Omega$. Therefore, (16) can be reformulated to an optimization problem as follows:

$$\begin{aligned} & \min_{K_\mathsf{p}} \gamma \\ & \text{subject to} \\ & \left| \begin{bmatrix} W_1(\omega) \mathcal{S}_\mathsf{p}(e^{j\omega}) \\ W_2(\omega) \mathcal{U}_\mathsf{p}(e^{j\omega}) \\ W_3(\omega) \mathcal{T}_\mathsf{p}(e^{j\omega}) \end{bmatrix} \right|^2 < \gamma \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \end{aligned} \tag{18}$$

For conciseness, the dependency on ω is omitted in further equations. The above inequality can be multiplied on both sides by $P_\mathsf{p}^* P_\mathsf{p}$:

$$\left| \begin{bmatrix} W_1 Y_\mathsf{p} \\ W_2 X_\mathsf{p} \\ W_3 G_\mathsf{p} X_\mathsf{p} \end{bmatrix} \right|^2 < \gamma P_\mathsf{p}^* P_\mathsf{p} \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \ . \tag{19}$$

The inequality in (19) can be convexified around an arbitrary P_c using (12), and a convex approximation of (16) is then given by:

$$\begin{aligned} &\min_{X_\mathsf{p},Y_\mathsf{p}} \gamma \\ &\text{subject to} \\ &\left|\left[\begin{array}{c} W_1 Y_\mathsf{p} \\ W_2 X_\mathsf{p} \\ W_3 G_\mathsf{p} X_\mathsf{p} \end{array}\right]\right|^2 < \gamma \Phi_\mathsf{p} \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P}\;, \\ &\gamma, \Phi_\mathsf{p} > 0 \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \end{aligned} \tag{20}$$

where

$$\Phi_\mathsf{p} = P_\mathsf{p}^* P_c + P_c^* P_\mathsf{p} - P_c^* P_c. \tag{21}$$

This is a semi-infinite programming problem and can be implemented by second-order conic constraints.

D. Soft $\mathcal{H}_2$ requirements

In this section, the performance defined as the two-norm of the weighted sensitivity functions is considered:

$$\begin{aligned} &\min_{K_\mathsf{p}} \gamma \\ &\text{subject to} \\ &\left\|\left[\begin{array}{c} W_1 \mathcal{S}_\mathsf{p} \\ W_2 \mathcal{U}_\mathsf{p} \\ W_3 \mathcal{T}_\mathsf{p} \end{array}\right]\right\|_2^2 < \gamma \qquad \forall \mathsf{p} \in \mathbb{P}. \end{aligned} \tag{22}$$

Similar to the $\mathcal{H}_\infty$ case, (22) can be reformulated to an optimization problem on the spectral norm as follows:

$$\begin{aligned} &\min_{K_\mathsf{p}} \gamma \\ &\text{subject to} \\ &\left|\left[\begin{array}{c} W_1(\omega) \mathcal{S}_\mathsf{p}(e^{j\omega}) \\ W_2(\omega) \mathcal{U}_\mathsf{p}(e^{j\omega}) \\ W_3(\omega) \mathcal{T}_\mathsf{p}(e^{j\omega}) \end{array}\right]\right|^2 < \mu(\omega, \mathsf{p}) \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \\ &\int_{-\pi}^{\pi} \mu(\omega, \mathsf{p}) d\omega < \gamma \quad \forall \mathsf{p} \in \mathbb{P}, \end{aligned} \tag{23}$$

where $\mu(\omega, \mathsf{p})$ is an upper bound on the Euclidean norm of the vector of weighted sensitivity functions at ω. The dependency on ω is again omitted in further equations when possible. The inequality in (23) can be rewritten substituting K_p with (7), and multiplying both sides by $P_\mathsf{p}^* P_\mathsf{p}$, where P_p is defined in (17):

$$\begin{aligned} &\left|\left[\begin{array}{c} W_1 Y_\mathsf{p} \\ W_2 X_\mathsf{p} \\ W_3 G_\mathsf{p} X_\mathsf{p} \end{array}\right]\right|^2 < \mu(\omega, \mathsf{p}) P_\mathsf{p}^* P_\mathsf{p} \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \\ &\int_{-\pi}^{\pi} \mu(\omega, \mathsf{p}) d\omega < \gamma \quad \forall \mathsf{p} \in \mathbb{P}. \end{aligned} \tag{24}$$

The first inequality in (24) can be convexified around an arbitrary P_c using (12), and a convex approximation of (22) is given by:

$$\begin{aligned} &\min_{X_\mathsf{p},Y_\mathsf{p}} \gamma \\ &\text{subject to} \\ &\left|\left[\begin{array}{c} W_1 Y_\mathsf{p} \\ W_2 X_\mathsf{p} \\ W_3 G_\mathsf{p} X_\mathsf{p} \end{array}\right]\right|^2 < \mu(\omega, \mathsf{p}) \Phi_\mathsf{p} \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \\ &\mu(\omega, \mathsf{p}), \Phi_\mathsf{p} > 0 \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \\ &\int_{-\pi}^{\pi} \mu(\omega, \mathsf{p}) d\omega < \gamma \quad \forall \mathsf{p} \in \mathbb{P}, \end{aligned} \tag{25}$$

where Φ_p is defined in (21).

E. Hard requirements

For many applications, a set of hard requirements may be given in the design specifications. Common examples include a modulus margin and disturbance attenuation in some frequency bands. These hard requirements must be satisfied irrespectively of the objective value attained to ensure the good functioning of the machine during routine operations. Such requirements formulated as constraints on the weighted norms of the different closed-loop transfer functions can be enforced using convex constraints. In this section, we consider constraints to shape the different closed-loop transfer functions individually. For the $\mathcal{H}_\infty$ case, this corresponds to:

$$\begin{aligned} \left\|\underline{W_1} \mathcal{S}_\mathsf{p}\right\|_\infty < 1 \quad \left\|\underline{W_2} \mathcal{U}_\mathsf{p}\right\|_\infty < 1 \quad \left\|\underline{W_3} \mathcal{T}_\mathsf{p}\right\|_\infty < 1 \\ \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P}. \end{aligned} \tag{26}$$

Following the same steps as in Sec. IV-C, a second-order rotated cone constraint can be obtained:

$$\left|\underline{W_1} Y_\mathsf{p}\right|^2 < \Phi_\mathsf{p} \quad \left|\underline{W_2} X_\mathsf{p}\right|^2 < \Phi_\mathsf{p} \quad \left|\underline{W_3} G_\mathsf{p} X_\mathsf{p}\right|^2 < \Phi_\mathsf{p} \tag{27}$$

for all $\omega \in \Omega$ and for all $\mathsf{p} \in \mathbb{P}$.

For the $\mathcal{H}_2$ case, considering only a weighted norm on the sensitivity transfer function for brevity:

$$\left\|\underline{W_1} \mathcal{S}_\mathsf{p}\right\|_2^2 < 1, \quad \forall \mathsf{p} \in \mathbb{P}, \tag{28}$$

the same approach as in Section IV-D can be followed to derive the following convex constraints:

$$\begin{aligned} &\left|\underline{W_1} Y_\mathsf{p}\right|^2 < \mu(\omega, \mathsf{p}) \Phi_\mathsf{p} \qquad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \\ &\mu(\omega, \mathsf{p}), \Phi_\mathsf{p} > 0 \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \\ &\int_{-\pi}^{\pi} \mu(\omega, \mathsf{p}) d\omega < 1 \quad \forall \mathsf{p} \in \mathbb{P}. \end{aligned} \tag{29}$$

F. Closed-loop stability

Given an initial stabilizing (possibly LPV) controller with the same fixed parts as in (7):

$$K_c = \frac{N_\mathsf{p}}{D_\mathsf{p}} = \frac{N(z, \mathsf{p})}{D(z, \mathsf{p})} \frac{F_X(z, \mathsf{p})}{F_Y(z, \mathsf{p})}, \tag{30}$$

and defining

$$P_c = D_\mathsf{p} + G_\mathsf{p} N_\mathsf{p} \tag{31}$$

the closed-loop stability can be guaranteed under *Assumption* 1, using Theorem 1. According to this theorem, the following conditions must hold:

(a) $P_{\mathsf{p}}(e^{j\omega})$ and $P_c(e^{j\omega})$ have the same winding number (wno) around the origin when ω traverses the Nyquist contour :

$$\text{wno}(P_{\mathsf{p}}) = \text{wno}(P_c) \tag{32}$$

(b) The initial controller K_c and the final controller K_{p} have the same poles with the same multiplicity on the unit circle,

(c) The order of Y_{p} is equal to the order of D_{p}.

Note that none of these conditions are particularly restrictive. Condition (a) is satisfied when convexifying $P_{\mathsf{p}}^* P_{\mathsf{p}}$ around P_c, as $\Phi_{\mathsf{p}} > 0$ enforces $P_{\mathsf{p}}^* P_c + P_c^* P_{\mathsf{p}} > 0$ which guarantees $\text{wno}(P_{\mathsf{p}}) = \text{wno}(P_c)$ (see *Proposition* 1 below).

If K_{p} and K_c have the same structure, F_Y should contain all of K_{p} and K_c poles on the unit circle. It is sufficient to verify that

$$Y_{\text{var}}(e^{j\omega}, \mathsf{p}) \neq 0 \qquad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P} \tag{33}$$

to obtain condition (b). If the weighting function W_1 has any poles on the unit circle, the fixed part in the controller numerator F_Y should be chosen such as these poles are canceled. It is then *unlikely* to have an additional pole exactly on the unit circle, and (33) is *almost always* satisfied. This does not prohibit controllers with unstable poles (but stabilizing the closed loop). It is possible to add a constraint to enforce that K_{p} and K_c have the same number of stable poles by enforcing:

$$\text{wno}(Y_{\text{var}}) = \text{wno}(D), \tag{34}$$

assuming D only has stable poles. A sufficient condition for (34) can be implemented using the following constraint:

$$Y_{\text{var}}^* D + Y_{\text{var}} D^* > 0, \quad \forall \omega \in \Omega, \forall \mathsf{p} \in \mathbb{P}, \tag{35}$$

which also guarantees that (33) is satisfied. If the initial controller K_c is stable, this ensures that K_{p} is also stable.

Condition (c) can be obtained by padding K_c: multiplying both the numerator and denominator of K_c by powers of z to obtain the correct order.

V. Implementation Considerations

A. Gridding

The optimization problems in this paper are formulated as semi-infinite programming and contain an infinite number of constraints. A common approach to handle this type of problem is to choose a large set of frequency $\Omega_M = \{\omega_1, \ldots, \omega_M\} \subset \Omega$ and operating points $\mathbb{P}_N = \{\mathsf{p}_1, \ldots, \mathsf{p}_N\} \subset \mathbb{P}$, and then solve the SIP constraints at the chosen frequencies and operating points. The frequency range and the number of operating points should be taken dense enough such that all dynamics are properly captured.

When the $\mathcal{H}_2$ constraints are evaluated on a finite set of frequencies and operating points, a numerical integration scheme can be used to approximate the integral

$$\int_{-\pi}^{\pi} \mu(\omega, \mathsf{p}_n) d\omega \approx \frac{1}{2} \sum_{m=2}^{M} (\mu_{m,n} + \mu_{m-1,n}) (\omega_m - \omega_{m-1})$$

and $\mu_{m,n} > 0$ a scalar optimization variable representing the value of $\mu(\omega_m, \mathsf{p}_n)$.

The closed-loop stability and controller poles are only guaranteed when (32)-(34) are met for the continuum of frequencies and operating points. In the sampled case, the final winding number can be incorrect due to coarse gridding. This is prone to happen when the controller's poles are close to the unit circle, which is often the case for high-order controllers.

We propose to add additional constraints to ensure the winding number condition in (32) is met for at least the closed polygonal chain $\overline{P_{\mathsf{p}}}$ with vertices $\{P_{\mathsf{p}}(e^{j\omega_m})\}$, $m = 1, \ldots, M$ and $\omega_m \in \Omega_M$. For the chain to be closed, the last segment must connect $P_{\mathsf{p}}(e^{j\omega_M})$ to $P_{\mathsf{p}}(e^{j\omega_1})$.

Assume that the polygonal chain does not intersect the origin. Given a second closed polygonal chain $\overline{P_c}$ with the correct winding number, the aim is to build a series of convex constraints to guarantee $\text{wno}(\overline{P_{\mathsf{p}}}) = \text{wno}(\overline{P_c})$.

Let

$$\begin{aligned} \overline{P_{\mathsf{p}}}(\omega) &= (1-\lambda) P_{\mathsf{p}}(e^{j\omega_m}) + \lambda P_{\mathsf{p}}(e^{j\omega_{m+1}}) \\ \lambda &= \frac{\omega_{m+1} - \omega}{\omega_{m+1} - \omega_m}, \quad \omega_m \leq \omega \leq \omega_{m+1} \end{aligned} \tag{36}$$

be a continuous parametrization of the polygonal chain $\overline{P_{\mathsf{p}}}$, and similarly $\overline{P_c}(\omega)$ a continuous parametrization of $\overline{P_c}$.

Proposition 1: If

$$\text{Arg}\left\{ \overline{P_{\mathsf{p}}}(\omega) \, \overline{P_c}^*(\omega) \right\} \neq \pi \;\; \forall \omega \in \Omega \tag{37}$$

holds, then the two closed polygonal chains $\overline{P_{\mathsf{p}}}$ and $\overline{P_c}$ have the same winding number.

Inequality (37) can be explained using the analogy of two clock hands. If one hand rotates faster than the other and both return to their starting position after an hour, at some point during the hour, the hands will be facing opposite directions, which is not allowed. Thus, they must rotate at nearly the same angular velocity.

The condition in (37) is not convex in the controller parameters as the principal argument is a non-linear operator, and depends on the continuum Ω. Suppose that n_m is the closest point to the origin on the line segment $\overline{P_c}(\omega)$ when $\omega_m \leq \omega \leq \omega_{m+1}$. Therefore, after a rotation by $\alpha = \text{Arg}\{\mathsf{n}_m^*\}$, the line segment $e^{j\alpha}\overline{P_c}(\omega)$ lies on the right half-plane and equivalently $\Re\{\overline{P_c}(\omega)\mathsf{n}_m^*\} > 0$, when $\omega_m \leq \omega \leq \omega_{m+1}$. Thus, by construction

$$\left| \text{Arg}\left\{ \overline{P_c}(\omega) \mathsf{n}_m^* \right\} \right| < \frac{\pi}{2} \tag{38}$$

always holds. Additionally, if

$$\begin{aligned} \Re\{P_{\mathsf{p}}(e^{j\omega_m}) \mathsf{n}_m^*\} &> 0 \\ \Re\{P_{\mathsf{p}}(e^{j\omega_{m+1}}) \mathsf{n}_m^*\} &> 0 \\ m = 1, \ldots, M& \end{aligned} \tag{39}$$

holds, then $\overline{P_{\mathsf{p}}}$ and $\overline{P_c}$ have the same winding number. Because each segment $\omega_m \leq \omega \leq \omega_{m+1}$ satisfies

$$\begin{aligned} &\left| \text{Arg}\left\{ \overline{P_{\mathsf{p}}}(\omega) \left(\overline{P_c}(\omega) \right)^* \right\} \right| \leq \\ &\qquad \left| \text{Arg}\left\{ \overline{P_{\mathsf{p}}}(\omega) \mathsf{n}_m^* \right\} \right| + \left| \text{Arg}\left\{ \overline{P_c}(\omega) \mathsf{n}_m^* \right\} \right| < \pi \end{aligned}$$

and by virtue of *Proposition 1*, both chains $\overline{P_\mathsf{p}}(\omega)$ and $\overline{P_c}(\omega)$ have the same winding number.

Rearranging the indices in (39) results in two additional constraints at every frequency:

$$\begin{aligned} \Re\{P_\mathsf{p}(e^{j\omega_m})\mathsf{n}_m^*\} > 0 \\ \Re\{P_\mathsf{p}(e^{j\omega_m})\mathsf{n}_{m-1}^*\} > 0 \end{aligned} \tag{40}$$

where $\mathsf{n}_0 = \mathsf{n}_M$. This is a tractable condition that should be implemented to guarantee both polygonal chains have the same winding number. This constraint also conserves the property that any starting controller K_c is still a (possibly suboptimal) solution of the optimization problem (10), ensuring the optimal controller cannot achieve worse performance than K_c.

The same reasoning can also be used to constrain the location of the poles in the controller K_p, by enforcing the correct winding number of the polygonal chains formed by the variable part of the controller:

$$\text{wno}(\overline{Y}_{\text{var}}) = \text{wno}(\overline{D}). \tag{41}$$

Following the same derivation as for the improved closed-loop stability constraint, the following constraints can be added at each frequency (and operating point) to ensure the correct pole location:

$$\begin{aligned} \Re\{Y_{\text{var}}(e^{j\omega_m}, \mathsf{p})\tilde{\mathsf{n}}_m^*\} > 0 \\ \Re\{Y_{\text{var}}(e^{j\omega_m}, \mathsf{p})\tilde{\mathsf{n}}_{m-1}^*\} > 0 \end{aligned} \tag{42}$$

where $\tilde{\mathsf{n}}_m$ is the closest point of the segment with edges $\{D(e^{j\omega_m}, \mathsf{p}), D(e^{j\omega_{m+1}}, \mathsf{p})\}$ to the origin.

All inequalities for the negative frequencies range $[-\pi,\ 0]$ will be automatically satisfied if they are met for the positive range $[0,\ \pi]$. It is therefore sufficient to only implement any of the inequalities at the positive frequency range.

B. Iterative algorithm and initial controller

To solve the optimization problem, an initial stabilizing controller K_c satisfying the hard requirements is needed with the appropriate structure as described in (30). The optimal controller found by convex optimization will depend on this initial controller as an inner approximation of (16) or (22) is solved. This controller may be far from the global optimum. To improve the results, we propose to use an iterative approach that solves the optimization problem multiple times, using the final controller from one iteration as the initial controller for the next iteration. This procedure is initialized with a stabilizing controller K_c. The objective from the soft requirements will be non-increasing: the optimal solution at one iteration is a (possibly suboptimal) solution of the next iteration satisfying the constraints. The algorithm converges to a local optimum or a saddle point of the original non-convex problem [24].

If the initial controller does not satisfy the hard requirements, the optimization problem cannot be guaranteed to have a feasible solution. It is recommended to first solve a sub-problem, with as objective to find a controller satisfying the hard requirements, and seed the iterative algorithm with this new controller. This controller can be found by solving (possibly iteratively) the following sub-problem:

$$\begin{aligned} &\min_{K_\mathsf{p}} \ \varepsilon \\ &\text{subject to} \\ &\left\| \ R_{\text{hard}}(G_\mathsf{p}, K_\mathsf{p}) \ \right\|_p < 1 + \varepsilon, \\ &\varepsilon \geq 0 \end{aligned} \tag{43}$$

where p is the type of norm used in the original problem. If (43) converges to any $\varepsilon \neq 0$, then the hard requirements are not achievable from the starting point K_c and a different set of requirements should be chosen (or a different initial controller, or controller order). If the controller converges to $\varepsilon = 0$, the final controller should be used as starting point for the original synthesis problem.

VI. Experimental results

The proposed method has been applied on a real 3D Cartesian robot, which was described in Sec. III and whose FRFs at different operating points are shown in Fig. 3–5. The operating space corresponds to the unit cube with one m^3 volume:

$$\mathsf{p} \in [0,\ 1] \times [0,\ 1] \times [0,\ 1].$$

The sampled normalized frequency grid Ω_M is chosen as 350 logarithmically spaced points, ranging from 10^{-2} to π (included). The operating space is sampled at a discrete number of points $\mathbb{P}_N$. For each axis, the control design focus is to minimize the $\mathcal{L}_2$ norm of the tracking error given a ramp reference signal:

$$\min_{K_\mathsf{p}} \ \|e(t)\|_{\mathcal{L}_2} \tag{44}$$

By Parseval's theorem, minimizing the $\mathcal{L}_2$ norm in the time domain is equivalent to minimizing the $\mathcal{H}_2$ norm in the frequency domain. The z-transform is applied on (44): the z-transform of a ramp is $\frac{1}{z^2-2z+1}$ and the corresponding objective to minimize is $\left\| \mathcal{S}_\mathsf{p} \frac{z^2}{z^2-2z+1} \right\|_2$. Thus, the following weighting functions for the soft requirements are chosen:

$$W_1 = \frac{z^2}{z^2 - 2z + 1},\ W_2 = W_3 = 0. \tag{45}$$

Hard requirements are specified through upper bounds on the norm of different closed-loop sensitivity functions and must be satisfied irrespectively of the value attained in (44). $\underline{W_1}$ is chosen such that a modulus margin of 2 is guaranteed. For the x and y-axis, $\underline{W_2}$ is chosen such that the maximal gain of the input sensitivity is 80dB, with a large attenuation near the Nyquist frequency to reduce noise effects and avoid saturation of the actuators. For the z-axis, this filter is chosen as

$$\underline{W_2}(\omega) = \begin{cases} 0 & 0 \ \leq \omega \leq 0.5 \\ -60\text{dB} & 0.5 < \omega \leq \pi \end{cases}.$$

For this machine, it is required that high-frequency oscillations are attenuated during closed-loop reference tracking. Therefore, $\underline{W_3}$ is chosen such that the complementary sensitivity has a maximal gain of 2 at low frequencies and roll-off with a slope of -20dB per decade, starting around $\omega = 0.3$.

For each axis, the problem to solve is:

$$\begin{aligned} &\min_{K_{\mathsf{p}}} \gamma \\ &\text{subject to} \\ &\| \; R_{\text{soft}}(G_{\mathsf{p}}, K_{\mathsf{p}}) \; \|_2 < \gamma, \quad \| \; R_{\text{hard}}(G_{\mathsf{p}}, K_{\mathsf{p}}) \; \|_\infty < 1 \\ &\hspace{18em} \mathsf{p} \in \mathbb{P}, \end{aligned} \tag{46}$$

where R_{soft} and R_{hard} are defined in (22) and (26) using the weights described above. This problem is implemented at the sampled Ω_M and $\mathbb{P}_N$ using (25) and (27). The stability constraint (40) is added to conserve the polygonal winding number throughout iterations. To prevent the design of unstable but stabilizing controllers, the constraint (42) is added to the optimization problem. Without these two last constraints, reducing the number of frequency points, or increasing the controller order, leads to destabilizing controllers.

To account for the many resonant peaks of the Cartesian robot, a final controller of order 16 is desired for each axis. This order is chosen because increasing the order over 16 leads only to marginally better performance, and overall complexity should remain low. A simple initial controller with a double integrator and a zero very close to 1 is tested on the system and known to be stabilizing:

$$K_0 = 100 \cdot \frac{z - 0.99}{z^2 - 2z + 1}.$$

Note that this controller stabilizes all three axes, as they all have, by design, similar low-frequency dynamics. This initial controller is far from resulting in satisfactory performance, the hard requirements are not satisfied, and the controller does not have the correct order to start the iterative algorithm. Nevertheless, it has the correct structure: a double integrator is required to cancel the two poles present in W_1. To obtain the proper order, the controller numerator and denominator are multiplied by z^{14}:

$$K_0 = 100 \cdot \frac{z^{14}}{z^{14}} \frac{z - 0.99}{z^2 - 2z + 1} = 100 \cdot \frac{z^{15} - 0.99 z^{14}}{z^{14}} \cdot \frac{1}{F_Y} \tag{47}$$

The double integrator present in the controller has poles on the unit circle and is added to the fixed part F_Y. Using this controller, the hard requirements are not satisfied, but a valid controller K_c can be obtained after solving (43) once, which is then used as a new starting point when solving (46).

A. x-axis tuning

To obtain improved performance over traditional control approaches, an LPV controller is considered. The scheduling vector should reflect the dependency of the dynamics (6) w.r.t. the operating points. A first guess is a quadratic form in the operating points, and the scheduling vector for the controller is chosen accordingly:

$$\theta_x = [1, \mathsf{x}, \mathsf{y}, \mathsf{z}, \mathsf{xy}, \mathsf{xy}, \mathsf{yz}, \mathsf{x}^2, \mathsf{y}^2, \mathsf{z}^2].$$

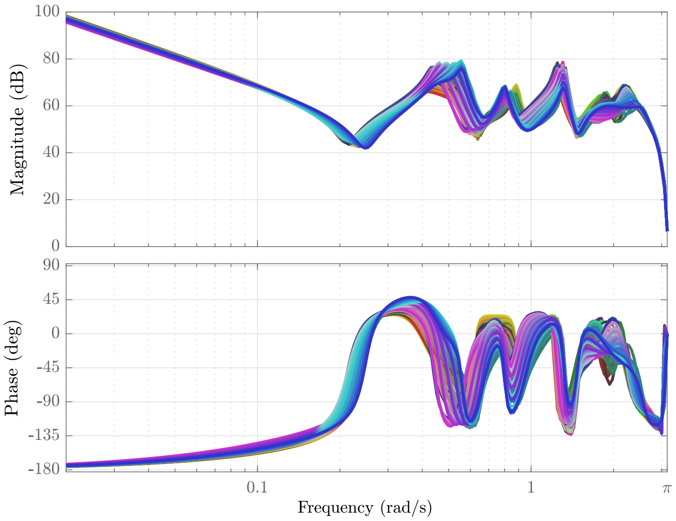


Fig. 6: FRF of the (frozen) x-axis LPV controller obtained after solving (46). The same color code for the FRF as the x-axis dynamics, shown in Fig. 3, is used.

To reduce the number of scheduling parameters, (46) is first solved with additional L_1 regularization [25]:

$$\begin{aligned} &\min_{K_{\mathsf{p}}} \gamma + \sum_{k=0}^{n_\theta} \left(\lambda_1 \sum_{i=0}^{a} |x_{ik}| + \lambda_2 \sum_{i=0}^{b} |y_{ik}| \right) \\ &\text{subject to } (40), (42), \\ &\| \; R_{\text{soft}}(G_{\mathsf{p}}, K_{\mathsf{p}}) \; \|_2 < \gamma, \quad \| \; R_{\text{hard}}(G_{\mathsf{p}}, K_{\mathsf{p}}) \; \|_\infty < 1 \\ &\hspace{16em} \omega \in \Omega, \mathsf{p} \in \mathbb{P}. \end{aligned} \tag{48}$$

With the appropriate choice of $\lambda_1 > 0$ and $\lambda_2 > 0$, solving (48) can help identify coefficients contributing little to the overall solutions. Scheduling the controller coefficient with xz, yz or z^2 has been noted to have only a marginal impact on the final value of γ. The scheduling vector is therefore chosen as:

$$\theta_x = [1, \mathsf{x}, \mathsf{y}, \mathsf{z}, \mathsf{xy}, \mathsf{x}^2, \mathsf{y}^2]. \tag{49}$$

The discussed iterative procedure in Sec. V-B is applied, until convergence to a final controller, whose frequency response is shown in Fig. 6. It can be seen from the color change that the first resonant mode of the controller's FRF mostly changes with the x-position, and the second resonant mode mostly with the y-axis position, similar to the model dynamics shown in Fig. 3. This highlights that the LPV controller can correctly deal with the changing resonant modes of the system. The closed-loop sensitivity functions are shown in Fig. 7 along with the inverse of the $\underline{W_1}, \underline{W_2}$ and $\underline{W_3}$ functions. The hard requirements are satisfied, and can be seen by the different sensitivity functions lying under of the inverse of the weighting functions. This controller is stabilizing everywhere, including all trajectories that have been tested.

For comparison, (46) is solved using no position dependency in the controller ($\theta_x = 1$), which corresponds to designing a robust controller with multimodel uncertainty. The controller can be computed much faster, but results in a final $\mathcal{H}_2$ norm over twice as large. The robustness and performance is a direct trade-off due to the large variance in the model dynamics. When performance is required, using a position-dependent controller shows clear benefits: a greater bandwidth can be achieved, without sacrificing robustness specified through the hard requirements. The cost to pay is a larger complexity in the controller and longer computation times.

To compare both controllers, a ramp from $\mathsf{x} = 0$ to $\mathsf{x} = 1$ with intermediate stops is used as reference x_d. The tracking error with this reference using the robust or the LPV controller can be found in Fig. 8. To fit in the plots, the reference x_d is scaled and centered. The axes y and z are moving such that different operating points are visited during the tracking of this reference. The RMS (root mean squared) value of the error using the non-LPV controller K is $30.48\,\mu$m, whereas using the LPV controller K_p is $14.51\,\mu$m. The maximal tracking error is two times smaller using the LPV controller. Small oscillations can be seen during steady-state and are explained by system nonlinearities and the vibrations caused by the movements of the two other axes.

B. SDP vs. SOCP

When sampling the operating space at 100 different points as for the x-axis and the frequency spectrum at 350 different points, the resulting problem has 1.4×10^5 second-order conic constraints, as many linear constraints, combined with 217 optimization variables from the controller parameters (119 in the numerator, 98 in the denominator) and 3.5×10^4 optimizations variables $\mu_{k,j}$ used to approximate the $\mathcal{H}_2$ objective.

To showcase the decreased solve time using the proposed formulation, the sampled version of (46) is solved using both the SOCP and SDP formulation. The SDP formulation is obtained by substituting the second-order conic constraints with (15). The problem is solved using a MacBook Pro 2019 with a quad-core Intel i7 2.3GHz CPU. The problem is set up with Yalmip [26] and solved using Mosek[2]. Mosek implements both SOCP and SDP solvers and serves as a fair comparison solver between both formulations.

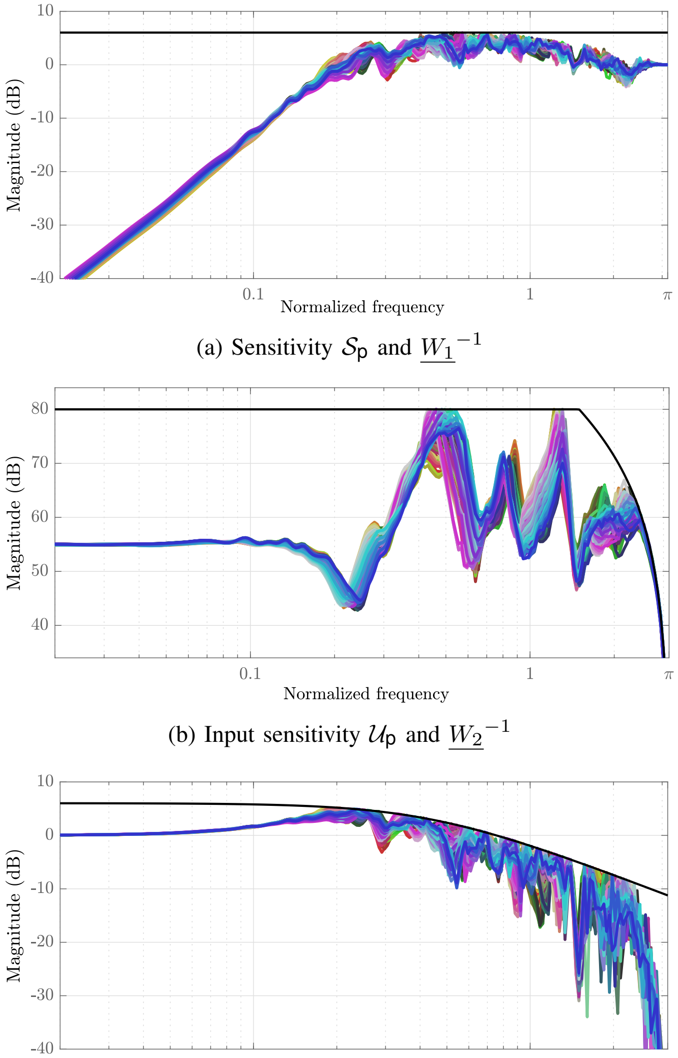
Solving the sampled version of (46) once takes on average 100 seconds per iteration using the SOCP formulation, whereas the SDP formulation takes 9 times more time to obtain the same result: on average 900 seconds. The iterative algorithm requires solving (46) multiple times, each time using the last optimal controller as the new initial controller, the second-order conic formulation makes it possible to find an optimal solution in a meaningful time frame.

C. y-axis tuning

The same procedure is applied to the y-axis. The synthesis is very similar to that of the x-axis and thus only discussed briefly. The same optimization problem is to be solved, using the same stabilizing controller and the same constraints. The only differences are the models G_p and the used scheduling vector. The L_1 regression is applied using quadratic forms in y and z and leads to the following scheduling vector for the y-axis:

$$\theta_y = [1,\, \mathsf{y},\, \mathsf{y}^2, \mathsf{z}].$$

This scheduling vector does not depend on the x-axis position; because the y-axis is mounted on top of the x-axis, and its dynamics do not significantly depend on the x position. Using this scheduling vector, a controller is synthesized, and the final closed-loop sensitivity functions are shown in Fig. 9. Using a robust controller, the performance criterion is $\approx 50\%$ higher, indicating again an LPV controller can notably improve performance.



(a) Sensitivity $\mathcal{S}_\mathsf{p}$ and $\underline{W_1}^{-1}$

(b) Input sensitivity $\mathcal{U}_\mathsf{p}$ and $\underline{W_2}^{-1}$

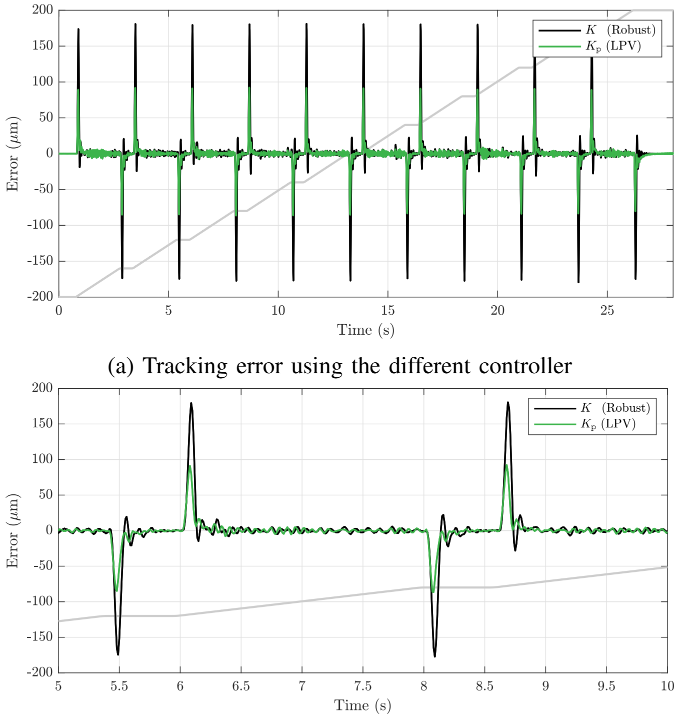
(c) Complementary sensitivity $\mathcal{T}_\mathsf{p}$ and $\underline{W_3}^{-1}$

Fig. 7: x-axis closed-loop sensitivities using the LPV controller. The same color code as Fig. 3 is used. The inverse of weighting filters (shown using the black solid lines) from the $\mathcal{H}_\infty$ hard requirements correspond to the upper bound on the magnitude of the corresponding sensitivity function.

D. z-axis tuning

The same tuning process is carried out a 3rd time for the

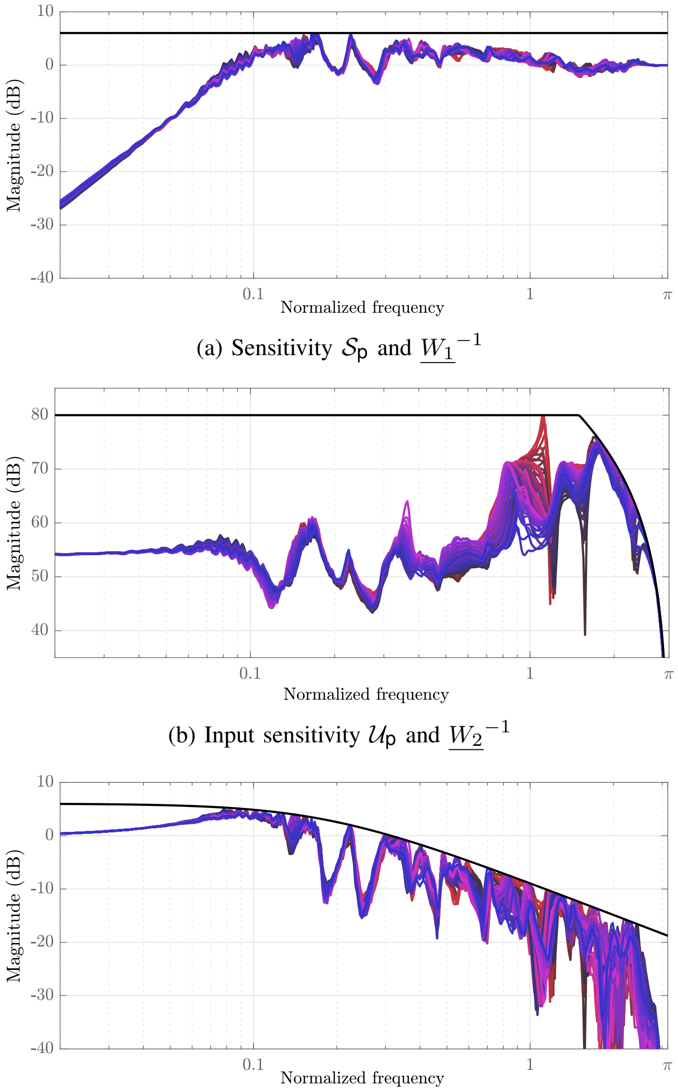
[2] http://www.mosek.com



(a) Tracking error using the different controller

(b) Zoom on the tracking error

Fig. 8: Tracking error using the LPV and robust controller. The reference x_d, scaled to fit on the plots, is shown using the solid gray line. Rapid acceleration and deceleration result in large peaks in the tracking error. The closed-loop using the LPV controller achieves a tracking error twice as small as the one using the robust controller. In-between acceleration and deceleration phases, the double-integrator present in both controllers can reject the constant-velocity reference, and only small disturbances (< 5 μm) can be seen.



(a) Sensitivity $\mathcal{S}_\mathsf{p}$ and $\underline{W_1}^{-1}$

(b) Input sensitivity $\mathcal{U}_\mathsf{p}$ and $\underline{W_2}^{-1}$

(c) Complementary sensitivity $\mathcal{T}_\mathsf{p}$ and $\underline{W_3}^{-1}$

Fig. 9: y-axis closed-loop sensitivities using LPV controller. The same color code as Fig. 4 is used. This axis has less pronounced LPV dynamics (w.r.t. the x-axis), but due to the first resonance peak occurring at $\omega = 0.1$, a similar improvement in the soft constraints can be obtained.

final axis. The z-axis dynamics have a negligible dependency on any of the other axes, and the L_1 regression indicates that adding an LPV dependency to the controller results in only marginal improvements. Therefore, a robust controller with $\theta_z = 1$ is used for this axis. The final closed-loop sensitivity functions using the tuned controller are shown in Fig. 10.

E. Tracking of a helicoidal reference

All three controllers are used to track a helicoidal reference in the operating space. The reference positions are given by

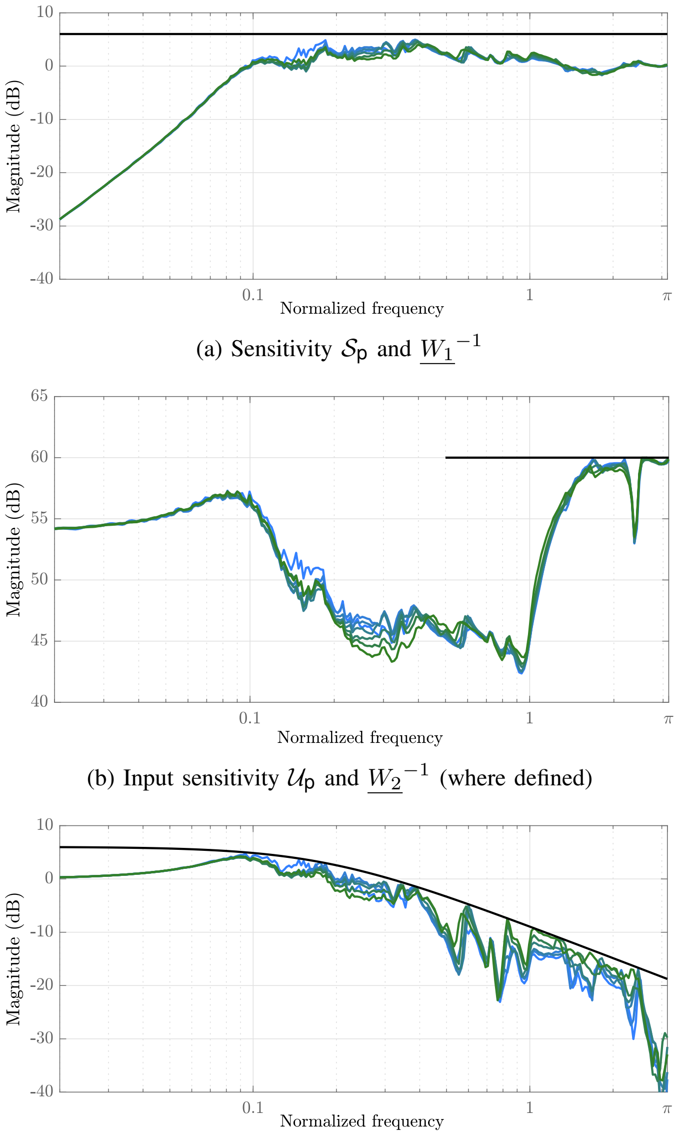
$$r(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} 0.575 - 0.425\cos(\phi(t)) \\ 0.575 - 0.425\cos(\phi(t-1)) \\ 0.9 - 0.8\frac{\phi(t)}{20\pi} \end{bmatrix}, \tag{50}$$

where

$$\phi(t) = \begin{cases} 0 & t < 0 \\ \frac{\pi t}{2} & 0 \leq t < 40 \\ 20\pi & t \geq 40 \end{cases}.$$

The reference for the z-axis is a ramp, which the double integrator in the controller of the corresponding axis can perfectly reject. This slow-moving ramp is used to compare only the axes where LPV dynamics are noticeable, while still moving in the operating space. This trajectory has a much greater velocity profile as would be seen during routine operation. Thus, it is expected that if the closed-loop is stable for this reference, all other *normal* closed-loop trajectories are stable too.

The tracking root-mean-square error (RMSE) is shown in Fig. 11, along with the tracking RMSE from the same trajectory when using a robust controller for all axes. As can be seen, this system benefits significantly from using an LPV controller: as the RMSE is twice as small for this trajectory. The operating space is covered in 4 seconds, but the closed-loop bandwidth is still large enough to maintain stability, so the assumption that the dynamics are frozen is justified. The closed-loop system remains stable at all visited operating



(a) Sensitivity $\mathcal{S}_\mathrm{p}$ and $\underline{W_1}^{-1}$

(b) Input sensitivity $\mathcal{U}_\mathrm{p}$ and $\underline{W_2}^{-1}$ (where defined)

(c) Complementary sensitivity $\mathcal{T}_\mathrm{p}$ and $\underline{W_3}^{-1}$

Fig. 10: z-axis closed-loop sensitivities using a robust controller. The same color code as Fig. 5 is used. For this axis, the lack of significant position dependency does not justify the use of an LPV controller.

points.

VII. Conclusion

A new method to tune LPV controllers has been proposed. Mixed-sensitivity objectives and constraints can be imposed on the different closed-loop sensitivity functions, with user-defined controller order. The main advantage is both $\mathcal{H}_2$ and $\mathcal{H}_\infty$ LPV synthesis is possible using only the FRF of the system at different operating points. This avoids the costly parametric modeling phase. The main drawback is that no theoretical stability guarantees are given for non-frozen dynamics or in-between operating points when solving the sampled version of the mixed-sensitivity problem, but experimental results show good stability and performance for our Cartesian robot.

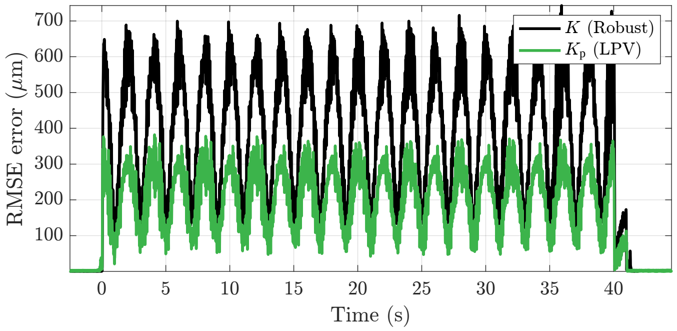


Fig. 11: End effector RMSE when tracking the helicoidal reference (50), using either the LPV or the robust controller. The LPV controller achieves a lower RMSE compared to the robust controller.

This approach lends itself particularly well to many industrial applications, where the FRF remains a common model for the dynamics of complex mechatronic systems. This approach has been applied to an industrial high precision industrial robot, showing a significant improvement over traditional control design. Moreover, the engineering costs are significantly reduced w.r.t. the classical gain-scheduled controller design. For this application, designing a gain-scheduled controller that includes the parametric identification of 100 models, designing as many model-based controllers, their interpolation, and real-time validation was estimated to be at least one week for a control engineer by our industrial partner. The proposed approach could reduce the total design time (automatic data acquisition and LPV controller design) to less than 4 hours, which reduces significantly the engineering costs of the project.

Future research directions include extending the proposed stability conditions to the whole spectrum of the FRF (and not only the polygonal chain) by integrating data-driven identification with control, e.g., accounting for inter-frequency uncertainty in the estimated FRF and improving the formulation for MIMO systems.

References

[1] Tom Oomen, Robbert Van Herpen, Sander Quist, Marc Van De Wal, Okkoo Bosgra, and Maarten Steinbuch. Connecting system identification and robust control for next-generation motion control of a wafer stage. *IEEE Transactions on Control Systems Technology*, 22:102–118, 1 2014.

[2] Hendrik Van Brussel, Paul Sas, István Németh, Pierre De Fonseca, and Pieter Van Den Braembussche. Towards a mechatronic compiler. *IEEE/ASME Transactions on Mechatronics*, 6:90–105, 3 2001.

[3] Yusuf Altintas, Alexander Verl, Christian Brecher, Luis Uriarte, and Guenter Pritschow. Machine tool feed drives. *CIRP Annals*, 60:779–796, 1 2011.

[4] Sebastian Kehne, Thomas Berners, Alexander Epple, and Christian Brecher. Optimal loop shaping of position controls of feed axes in machine tools with position-dependent behavior. *2019 IEEE International Electric Machines and Drives Conference, IEMDC 2019*, pages 571–575, 5 2019.

[5] Olof Garpinger, Tore Hägglund, and Karl Johan Åström. Performance and robustness trade-offs in PID control. *Journal of Process Control*, 24:568–577, 5 2014.

[6] Wilson J. Rugh and Jeff S. Shamma. Research on gain scheduling. *Automatica*, 36:1401–1425, 10 2000.

[7] Christian Hoffmann and Herbert Werner. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23:416–433, 3 2015.

[8] Wim Symens, Hendrik Van Brussel, and Swevers Jan. Gain-scheduling control of machine tools with varying structural flexibility. *CIRP Annals*, 53:321–324, 1 2004.

[9] Wim Paijmans, Bartand Symens, Hendrik Van Brussel, and Jan Swevers. A gain-scheduling-control technique for mechatronic systems with position-dependent dynamics. *2006 American Control Conference*, page 6 pp., 2006.

[10] Maíra M da Silva, Olivier Brüls, Jan Swevers, Wim Desmet, and Hendrik Van Brussel. Computer-aided integrated design for machines with varying dynamics. *Mechanism and Machine Theory*, 44(9):1733–1745, 2009.

[11] Liang Dong, Wencheng Tang, and Dafei Bao. Interpolating gain-scheduled $\mathcal{H}_\infty$ loop shaping design for high speed ball screw feed drives. *ISA Transactions*, 55:219–226, 3 2015.

[12] Tom Oomen. Advanced motion control for precision mechatronics: Control, identification, and learning of complex systems. *IEEJ Journal of Industry Applications*, 7:127–140, 3 2018.

[13] Rik Pintelon and Johan Schoukens. *System identification : a frequency domain approach*. Wiley, 2012.

[14] Marc Kunze, Alireza Karimi, and Roland Longchamp. Gain-scheduled controller design by linear programming with application to a double-axis positioning system. *Institute of Electrical and Electronics Engineers, Tech. Rep*, 2009.

[15] Alireza Karimi and Zlatko Emedi. $\mathcal{H}_\infty$ gain-scheduled controller design for rejection of time-varying disturbances with application to an active suspension system. *Proceedings of the IEEE Conference on Decision and Control*, pages 7540–7545, 2013.

[16] Tom Bloemers, Roland Tóth, and Tom Oomen. Data-driven LPV reference tracking for a control moment gyroscope. *IFAC-PapersOnLine*, 52:134–139, 1 2019.

[17] Tom Bloemers, Roland Tóth, and Tom Oomen. Towards data-driven LPV controller synthesis based on frequency response functions. *Proceedings of the IEEE Conference on Decision and Control*, 2019-December:5680–5685, 12 2019.

[18] Alireza Karimi and Christoph Kammer. A data-driven approach to robust control of multivariable systems by convex optimization. *Automatica*, 85:227–233, 11 2017.

[19] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*, volume 2. Wiley New York, 2007.

[20] Roland Tóth. *Modeling and Identification of Linear Parameter-Varying Systems*. Springer, 2010.

[21] Lennart Ljung. *System Identification*. Springer, 1998.

[22] Maarten Schoukens and Roland Tóth. Frequency response functions of linear parameter-varying systems. *IFAC-PapersOnLine*, 52(28):32–37, 2019.

[23] Robin de Rozario and Tom Oomen. Frequency response function identification of LPV systems: a global approach with application to mechanical systems. *IFAC-PapersOnLine*, 51:108–113, 1 2018.

[24] Yuille Alan and Rangarajan Anand. The concave-convex procedure. *Neural Computation*, 15:915–936, 4 2003.

[25] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[26] Johan Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, pages 284–289. IEEE, 2004.

Philippe Schuchert received his B.Sc. and M.Sc. degrees in mechanical engineering from École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 2016 and 2018, respectively, where he is currently pursuing this Ph.D. degree in the doctoral program for *Robotics, Control, and Intelligent Systems* part of the data-driven modeling and control group in the Automatic Control Laboratory.

From 2018 to 2020, he was a Research Assistant with the Automatic Control Laboratory. His research interest includes data-driven $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control with application to high-precision mechatronic systems.

Alireza Karimi received his Ph.D. from Institut National Polytechnique de Grenoble in France in 1997. He served as an Assistant Professor in the Department of Electrical Engineering at Sharif University of Technology in Tehran, Iran from 1998 to 2000. He then joined École Polytechnique Fédérale de Lausanne in Switzerland where he is currently a Professor of Automatic Control in the Mechanical Engineering Institute. He heads the data-driven modeling and control group in the Automatic Control Laboratory. His research focuses on data-driven controller design for power grids and mechatronic systems. He was an Associate Editor of the European Journal of Control from 2004 to 2013, and has been a member of the IEEE Control Systems Society conference editorial board since 2018.