*mathematics*

MDPI

*Article*

# Synthesis Methodology for Discrete MIMO PID Controller with Loop Shaping on LTV Plant Model via Iterated LMI Restrictions

**Artem E. Konkov * and Yuri V. Mitrishkin †**

Faculty of Physics, Lomonosov Moscow State University, 119991 Moscow, Russia
* Correspondence: konkov@physics.msu.ru
† Deceased author. Prof. Dr. Yuri V. Mitrishkin passed away on 22 January 2024.

**Abstract:** This paper presents a methodology for synthesizing discrete MIMO PID controllers through iterative solutions of LMIs. It justifies the necessity of direct synthesis of discrete controllers in digital control systems. The proposed methodology allows for synthesizing an LTI controller for an LTV plant model, ensuring the robust stability of the synthesized system. Robustness is further ensured by the small-gain theorem and a novel approach to loop shaping, enabling the use of arbitrary shape functions. As a result, this methodology provides a broad statement of the problem with performance criteria necessary for practical application in control systems. Numerical examples are used to illustrate the methodology, which is implemented using the MATLAB toolbox, freely available for use.

**Keywords:** discrete time; MIMO PID; LTV; PLV; loop shaping; LMI; convex-concave procedure

**MSC:** 93B52

check for updates

**Citation:** Konkov, A.E.; Mitrishkin, Y.V. Synthesis Methodology for Discrete MIMO PID Controller with Loop Shaping on LTV Plant Model via Iterated LMI Restrictions. *Mathematics* **2024**, *12*, 810. https://doi.org/10.3390/math12060810

Academic Editor: Zhijia Zhao

Received: 3 January 2024
Revised: 1 March 2024
Accepted: 7 March 2024
Published: 9 March 2024

**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

In this paper, we present an extension to the method for tuning multiple-input multiple-output (MIMO) proportional–integral–derivative (PID) controllers using iterated linear matrix inequality (LMI) restrictions to address discrete-time control systems. The original method was introduced for continuous-time systems by Boyd et al. in [1]. Our contributions and the novelty of this study lie in the following:

- Reformulating key concepts to accommodate discrete-time systems.
- Formulating matrix constraints for synthesizing a stationary MIMO PID controller for a non-stationary plant, where the MIMO PID is represented by a linear time-invariant (LTI) model and the plant is represented by a linear time-varying (LTV) model.
- Enhancing the original method by incorporating loop-shaping capabilities, utilizing a novel approach where the shape functions can be arbitrarily defined.
- Integrating numerous performance criteria essential for practical application into the problem statement.
- Devoting significant attention to ensuring the necessary robust stability margins of control systems.

Furthermore, we offer a comprehensive methodology for synthesizing discrete controllers, delve into the problem statement, and provide a full explanation of the synthesis method itself, including additional derivations and calculations.

This paper is structured as follows. Further in this section, we will discuss the necessity of employing MIMO PID in MIMO control systems. Subsequently, we will explore the reasons for the application of a digital control system, addressing the factors that necessitate synthesizing MIMO PID in discrete time. In Section 2, we provide the notations used and briefly outline the control theory concepts employed in our paper. Section 3 formulates the discrete MIMO PID controller synthesis problem as a semi-definite programming (SDP)

problem. We outline the performance criteria for the control system in terms of $\mathcal{H}_\infty$ norms. In Section 4, we demonstrate our approach to loop shaping. Section 5 provides a detailed description of the discrete MIMO PID controller synthesis method, demonstrating the transition from the general SDP problem to quadratic matrix inequalities (QMIs) and subsequently to linear matrix inequalities (LMIs) using the iterative convex-concave procedure (CCP). In Section 6, we briefly introduce the Matlab toolbox developed for controller synthesis and result visualization. Section 7 presents several examples of the synthesis of discrete MIMO PID controllers. Sections 8 and 9 conclude and summarize this work.

### *1.1. The PID Controller*

PID controllers are widely used in control systems due to their simplicity and effectiveness in controlling processes [2]. Their popularity is due to several key advantages that they offer. Firstly, they offer flexibility in control design, as control engineers can tune the proportional, integral, and derivative gains to meet specific application requirements. The basic equation of a continuous-time PID controller in parallel form is given by

$$u(t) = K_P\, e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt},$$

where $u(t)$ is the controller output, $e(t)$ is the controller input, and the tunable controller coefficients are $K_P$, $K_I$, and $K_D$ [3]. Secondly, PID controllers are robust and can efficiently handle disturbances in the system while tolerating small variations in parameters. Thirdly, they are cost-effective controllers that provide reliable control for a variety of applications. In addition, PID controllers effectively mitigate nonlinear effects, such as actuator saturation, for which specialized techniques, like anti-windup, exist. Furthermore, because PID controllers are linear, powerful and well-established methods for analyzing the synthesized control system can be applied.

However, it is important to note that there is no obligation to universally use PID controllers. The rational choice of the controller type depends on the plant under control. While PID controllers may not always be the best option for certain types of plants, they can still be used to solve most control tasks in real-world systems and processes [4].

There are many different methods available for tuning PID controllers in single-input single-output (SISO) control systems [3]. An experienced engineer can even manually tune a SISO PID controller. However, significant difficulties arise when there is a need to simultaneously control multiple outputs in a MIMO control system.

### *1.2. The MIMO PID Controller*

The approach where a MIMO system is controlled by a set of SISO PID controllers (i.e., multi-loop SISO PID control) works well only when the control loops within the MIMO system are sufficiently decoupled. Sequential tuning of the loops results in a degradation of the control performance of the previous loop as each subsequent loop is tuned. Utilizing loop decoupling techniques operates poorly, especially in systems with a large number of inputs and outputs. Additionally, the implementation of loop decoupling encounters challenges when dealing with an LTV plant model. Decoupling may only be effective for an LTI plant model and would be disrupted if the model parameters change. Decoupling methods for some classes of LTV plant models, such as linear parameter-varying (LPV) plant models, exist [5], but there are still significant limitations to the application of such methods [6]. In any case, the use of decoupling significantly complicates the design of MIMO control systems. For example, it can be difficult to ensure that the decoupling does not compromise the robust stability of the MIMO system.

We employ a method for tuning MIMO PID controllers that does not require prior loop decoupling and allows for the simultaneous calculation of all MIMO PID controller parameters that would satisfy specified control performance criteria.

The MIMO PID controller with $q$ inputs and $m$ outputs consists of $q \cdot m$ SISO PID controllers connected in a cross-coupled manner such that the $q$-th input is fed into each $q$-th SISO PID controller, and all $m$-th outputs are summed. An example of a MIMO PID controller with $q = 2$ and $m = 3$ is shown in Figure 1. The primary motivation for the utilization of MIMO PID controllers lies in the necessity to simultaneously control multiple outputs of a MIMO plant with well-coupled loops.

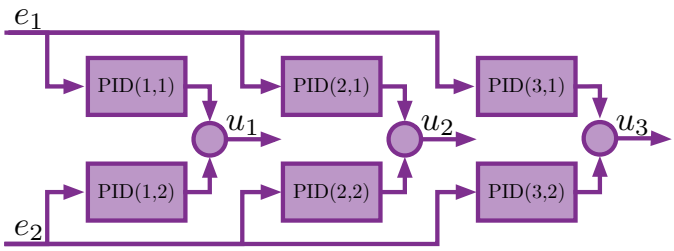



**Figure 1.** An example of a $3 \times 2$ MIMO PID controller with 2 inputs $(e_1, e_2)$ and 3 outputs $(u_1, u_2, u_3)$.

The interest in tuning MIMO PID controllers has been long-standing. It appears that the earliest algorithms for MIMO PID tuning can be traced back to [7,8]. Practically applicable tuning algorithms emerged with the active adoption of the LMI method in control theory [9]. It seems that the pioneering work was [10], where the synthesis of MIMO PID was formulated as a static output-feedback (SOF) synthesis problem. This idea was further developed in subsequent works [11], including the discrete-time MIMO PID synthesis method [12], although direct loop shaping was not explicitly performed in this work. Moreover, these algorithms involve less optimal practices, such as the random generation of a set of initial matrices for MIMO PID.

Another approach to synthesizing discrete MIMO PID controllers was proposed in [13], using the non-monotonic Lyapunov–Krasovskii theorem via LMIs. While this approach is intriguing, it does not allow for synthesis in the frequency domain, and the performance criteria of the system are defined in terms of weighting matrices. The issues of the robust stability of the synthesized system were not investigated.

The concept of loop shaping traces back to [14,15]. Its essence lies in defining performance requirements and robust stability margins of the closed-loop control system in terms of requirements on the singular values of the open-loop transfer function. The use of loop shaping in controller synthesis allows for direct manipulation of the frequency characteristics of the system, facilitating a trade-off between performance and robust stability margins [14]. Loop shaping gained popularity, particularly after the work in [16]. One of the early works employing loop shaping in the synthesis of MIMO PID controllers was the study in [17]. In this work, akin to classical loop shaping, a coprime factorization was used, and shaping functions were defined in the form of a pre-compensator and a post-compensator. The work in [11] is interesting in that it utilized a weighted sensitivity design.

As mentioned earlier, our approach is based on the concepts outlined in [1], which built upon the concepts introduced in [18]. The latter utilized a convex-concave procedure for tuning continuous SISO PID controllers in the frequency domain. The core principle of this method revolves around constraining the $\mathcal{H}_\infty$-norm of the closed-loop transfer functions by bounding their maximum gains across a finite yet sufficiently extensive set of frequencies. We apply the same principle but for the synthesis of a discrete MIMO PID controller. Moreover, we perform loop shaping by directly specifying the shape functions without any pre-compensators or post-compensators. The synthesis of a MIMO PID controller on an LTV plant model is rarely addressed, but its possibility was mentioned in [1]. Our study is possibly the first to adopt such an approach.

### *1.3. The Necessity of Employing Digital Control*

When it comes to the practical implementation of control systems, the choice of the device on which the controller will be implemented is a crucial question. Analog devices were the first historically, and for SISO control systems, this option is quite suitable. An analog SISO PID controller can be implemented using three operational amplifiers.

However, for the MIMO PID controller, significantly more operational amplifiers are required. For example, to implement a MIMO PID with $q = 2$ inputs and $m = 3$ outputs, one would need 21 operational amplifiers (18 to implement all its components and another 3 for summing the outputs $3 \cdot q \cdot m + q$; see Figure 1). Adding one more control loop ($q = 3$, $m = 4$) would increase the number of operational amplifiers to 40. The practical realization of such a device, encompassing calibration, parameter fine-tuning, or future modifications, presents formidable challenges. Furthermore, considerations must extend to the finite bandwidths and restricted signal ranges inherent in operational amplifiers. Consequently, anticipating dependable and uninterrupted functionality from such a system is, to say the least, an ambitious proposition and, in some cases, entirely impossible. Implementing the MIMO PID controller as a digital device can avoid many of these problems.

A basic digital control system (see also Figure 1.1 in [19]) is depicted in Figure 2.

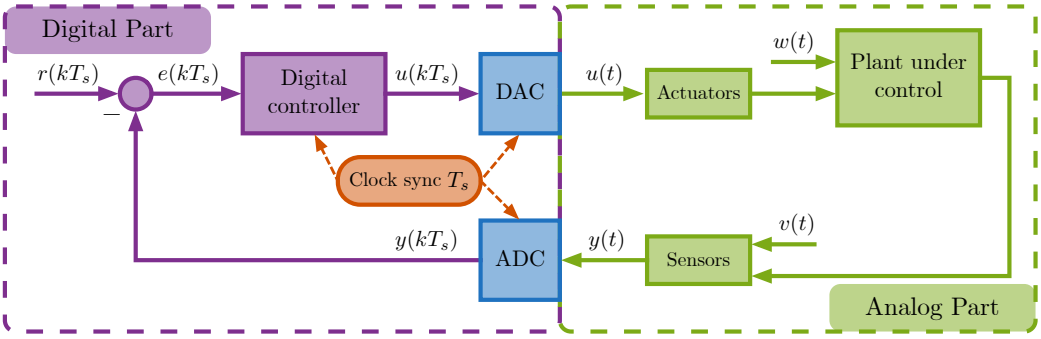



**Figure 2.** A block diagram of a basic digital control system.

The digital control system consists of two parts. In the analog part, there is a plant, which can be affected by external input disturbances $w(t)$, actuators receiving control signals $u(t)$, and sensors measuring the outputs of the plant $y(t)$, which can also be influenced by external output disturbances $v(t)$. In the digital part, there is a digital (discrete) controller, which, based on the input error vector $e(kT_s)$, computes the vector of control signals $u(kT_s)$. Additionally, there is a block that forms the reference vector $r(kT_s)$, an arithmetic unit calculating the error vector $e(kT_s) = r(kT_s) - y(kT_s)$. An Analog-to-Digital Converter (ADC) transforms the analog signal $y(t)$ into a sequence of samples (digital signal) $y(kT_s)$, whereas a Digital-to-Analog Converter (DAC) transforms the sequence of samples $u(kT_s)$ back into an analog signal $u(t)$. All components in the digital part must be time-synchronized, and the receiving, processing, and transmitting of samples should occur with a consistent sample time $T_s$. The terms "discrete controller" and "digital controller" are synonyms; the former is commonly used in the context of control theory, whereas the latter is used in the context of engineering.

Usually, the entire digital component is realized as a unified device. This device can be microcontrollers of various types, FPGA modules, or general-purpose computers. When the discrete controller is only a part of a more complex control algorithm or when there are increased demands for accuracy and fault tolerance, real-time operating system (RTOS)-controlled computers are used. The latter are also commonly referred to as real-time target machines. When simulating such a digital device in the feedback loop of the control system, it is commonly referred to as HIL (Hardware-in-the-Loop).

The number of channels in the ADC and DAC in the digital device should correspond to the number of inputs and outputs of the discrete controller. The role of the DAC in the digital system can also be performed by actuators. For example, if the actuators are voltage

inverters operating in PWM mode, the signal from the digital output of the device can be directly applied to the gates of the inverter transistors.

### *1.4. The Necessity of Discrete Controller Synthesis*

There are two approaches to obtaining the parameters for a discrete controller:

1. Initially synthesize a continuous (analog) controller using the continuous-time plant model and then discretize the controller. This is currently the predominant approach.
2. Initially discretize the plant model and then synthesize a discrete controller using the discrete-time plant model. This is what we use in this paper.

The discretization algorithm for both the controller and the plant model should match the type of ADCs and DACs used, typically being ZOH (Zero-Order Hold), and occasionally FOH (First-Order Hold).

The main challenge here is fundamental. According to the Kotelnikov theorem, proposed by Vladimir A. Kotelnikov in 1933 [20], also known as the Whittaker–Kotelnikov–Shannon sampling theorem or Nyquist–Shannon sampling theorem, any analog signal $y(t)$ consisting of frequencies from 0 to $f_{nyq}$ can be accurately transmitted and reconstructed using the sequence of samples $y(kT_s)$ taken consecutively at a rate of at least $f_s = 2f_{nyq}$. Here, $f_{nyq}$ is the Nyquist frequency, $f_s = 1/T_s$ is the sampling frequency (sample rate), and $T_s$ is the sample time. In digital devices, the upper bandwidth limit is critically determined by the Nyquist frequency of its ADC and DAC, a departure from analog devices where one must account for the bandwidths of individual components in the analog circuit.

The fundamental issue with the first approach lies in the fact that when synthesizing an analog controller, the effects associated with its subsequent discretization are not directly taken into consideration (see Sections 3.2 and 10 in [19]). However, for the proper operation of a digital control system, it is critically imperative to adhere to the conditions outlined in the Kotelnikov theorem. This issue is typically mitigated by introducing anti-aliasing filters in the feedback loop. Additionally, when loop shaping is employed during controller synthesis, a bandwidth is selected with a margin within the range from 0 to $f_{nyq}$ (see Section 13.3 in [3]). In cases where achieving a large margin is impractical, high-sample-rate ADCs and DACs are deployed (see Section 11 in [19]). However, it is crucial to note that an increase in the sample rate in ADCs and DACs comes with an augmented cost. Presently, there exists a significant cost discrepancy between devices with sampling frequencies of 1 kHz and 1 MHz.

In a broader context, the problem of maintaining acceptable control performance and robust stability margins during controller discretization has been extensively researched. Obviously, most of these studies focus on SISO systems (see Chapter 8 in [21]), and in the context of MIMO systems, the issues are more intricate [22]. In any case, the reality persists that the performance and robust stability margins of the discretized system are inferior to their pre-discretization counterparts, and the extent of this degradation is contingent on the sample time–larger sample times result in diminished performance and robust stability margins. The closer the crossover frequencies of the control system are to the Nyquist frequency, the more the robust stability margin degrades in the discretized control system (see Section 5.2 in [19]). This becomes more apparent when regarding continuous systems as a specific case of discrete systems when $T_s \to 0$.

In summary, opting for the first approach will yield a control system significantly different from the one obtained when synthesizing the controller in continuous time. It is a well-known fact that the frequency responses of the discretized system and the original continuous system will closely match only at low frequencies.

In the second approach, all discretization effects are inherently captured in the discrete-time model of the plant on which the discrete controller is synthesized. The synthesized digital control system possesses precisely the performance and robust stability margins specified during the discrete controller synthesis. This extends to the frequency responses of the digital system as well. By applying loop shaping, we can directly shape the frequency responses of the digital control system. For instance, achieving a certain robust stability

margin in the closed-loop system may eliminate the need for using anti-aliasing filters in the feedback loop of the digital control system. All of this serves as the primary motivation for the direct synthesis of a discrete controller.

## 2. Terminology and Notations

The following symbols/notations are used in this paper:

- $A \succeq 0$ denotes that the Hermitian matrix $A$ is positive semi-definite: $x^{\mathrm{H}}Ax \geq 0, \forall x$, $A = A^{\mathrm{H}}, x \in \mathbb{C}^n, A \in \mathbb{C}^{n\times n}$;
- $x \geq y$ denotes that the scalar $x$ is greater than or equal to the scalar $y$;
- $A \preceq 0$ denotes that the Hermitian matrix $A$ is negative semi-definite: $x^{\mathrm{H}}Ax \leq 0, \forall x$, $A = A^{\mathrm{H}}, x \in \mathbb{C}^n, A \in \mathbb{C}^{n\times n}$;
- $x \leq y$ denotes that the scalar $x$ is less than or equal to the scalar $y$;
- $A \succeq B$ denotes that the matrix $A - B$ is positive semi-definite: $A - B \succeq 0$;
- $A^{\mathrm{H}}$ denotes the (Hermitian) conjugate transpose of the matrix $A$;
- $\|A\|_2 = \overline{\sigma}(A)$ denotes the spectral norm of the matrix $A$;
- $\overline{\sigma}(A)$ denotes the maximum singular value of the matrix $A$;
- $\underline{\sigma}(A)$ denotes the minimum singular value of the matrix $A$;
- $\|G(z)\|_\infty$ denotes the $\mathcal{H}_\infty$-norm of the matrix function $G(z)$;
- $A^\dagger$ denotes the Moore–Penrose inverse of the matrix $A \in \mathbb{C}^{q\times m}$:
  if the matrix $AA^{\mathrm{H}} \in \mathbb{C}^{q\times q}$ is invertible, then $A^\dagger = A^{\mathrm{H}}\left(AA^{\mathrm{H}}\right)^{-1}$, $AA^\dagger = I_q$;
  if the matrix $A^{\mathrm{H}}A \in \mathbb{C}^{m\times m}$ is invertible, then $A^\dagger = \left(A^{\mathrm{H}}A\right)^{-1}A^{\mathrm{H}}$, $A^\dagger A = I_m$;
- $A^{-1}$ denotes the inverse of the square matrix $A \in \mathbb{C}^{n\times n}$:
  if the matrix $A$ is invertible (nonsingular), then $AA^{-1} = A^{-1}A = I_n$;
- $I_n$ denotes the $n \times n$ identity matrix, $I_n \in \mathbb{R}^{n\times n}$;
- $\exp(x) = e^x$, where $e$ is the base of natural logarithms (the Euler number);
- $T_s$ is the sample time and $f_s = 1/T_s$ is the sampling frequency;
- $\omega = 2\pi f$ is the angular frequency.

The singular values of a matrix $A \in \mathbb{C}^{m\times n}$ are the non-negative square roots of the eigenvalues of $AA^{\mathrm{H}}$ [23]

$$\sigma_i(A) = +\sqrt{\lambda_i(AA^{\mathrm{H}})}, \quad \sigma_i(A) \in \mathbb{R}^1,$$

arranged in non-increasing order:

$$\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_n(A) \geq 0, \quad \overline{\sigma}(A) = \sigma_1(A), \quad \underline{\sigma}(A) = \sigma_n(A).$$

If the matrix $A$ is invertible, then

$$\underline{\sigma}(A) = 1/\overline{\sigma}(A^{-1}), \quad \overline{\sigma}(A) = 1/\underline{\sigma}(A^{-1}).$$

The condition number $\kappa(A)$ is defined as the ratio of the maximum and minimum singular values of the matrix $A$:

$$\kappa(A) = \frac{\overline{\sigma}(A)}{\underline{\sigma}(A)}.$$

The invertible matrix $A \in \mathbb{C}^{n\times n}$ is a unitary matrix if

$$A^{\mathrm{H}}A = AA^{\mathrm{H}} = AA^{-1} = I_n, \quad \text{then} \quad \kappa(A) = 1, \quad \sigma_1(A) = \cdots = \sigma_n(A) = 1.$$

### *2.1. Some Important Aspects of Control Theory*

The Z-transform, first proposed in [24], is obtained from the discrete Laplace transform, first proposed in [25], by substituting

$$\exp(sT_s) \longrightarrow z, \quad \sum_{n=0}^{\infty} x[n] \exp(-sT_s n) \longrightarrow \sum_{n=0}^{\infty} x[n] z^{-n},$$

where $s$ is the Laplace transform variable and $z$ is the Z-transform variable.

The discrete MIMO transfer function $G(z) \in \mathbb{C}^{q \times m}$ is a matrix function that relates the Z-transforms of the inputs vector $u(z) \in \mathbb{C}^m$ and the outputs vector $y(z) \in \mathbb{C}^q$ for the system with zero initial conditions: $y(z) = G(z)u(z)$.

The discrete-time MIMO frequency response is the complex matrix function that is represented by

$$G\Big(\exp(j\omega T_s)\Big) \in \mathbb{C}^{q \times m},$$

and it is obtained by applying the discrete-time Fourier transform to the impulse response of $G(z)$. To obtain the value of the discrete-time MIMO frequency response at frequency $\omega$, we need to substitute $z \longrightarrow \exp(j\omega T_s)$ in the discrete-time MIMO transfer function $G(z)$. This substitution follows from the definitions of the Z-transform and the discrete Fourier transform, as shown below:

$$z \longrightarrow \exp(j\omega T_s), \quad \sum_{n=0}^{\infty} x[n] z^{-n} \longrightarrow \sum_{n=0}^{\infty} x[n] \exp(-j\omega T_s n).$$

The details of this substitution are presented in Chapter 8 in [26].

The maximum gain of the discrete MIMO transfer function $G(z)$ at the frequency $\omega$, denoted as $\|G\big(\exp(j\omega T_s)\big)\|_2$, is the spectral norm (maximum singular value) of the matrix, which represents the value of the discrete-time MIMO frequency response at the frequency $\omega$

$$\Big\|G\Big(\exp(j\omega T_s)\Big)\Big\|_2 = \overline{\sigma}\Big(G\Big(\exp(j\omega T_s)\Big)\Big) \in \mathbb{R}^1,$$

whereas the minimum gain of the discrete MIMO transfer function at the frequency $\omega$ is the minimum singular value of this frequency response.

The $\mathcal{H}_\infty$-norm of the discrete MIMO transfer function $G(z)$, denoted as $\|G(z)\|_\infty$, is the frequency peak that yields the maximum gain of the discrete MIMO transfer function over all frequencies $\omega \in [0, \ldots, \omega_{nyq}]$

$$\Big\|G(z)\Big\|_\infty = \sup_{\omega} \left\{ \overline{\sigma}\Big(G\Big(\exp(j\omega T_s)\Big)\Big) \right\} \in \mathbb{R}^1.$$

The Nyquist frequency is defined as half of the sampling frequency:

$$f_{nyq} = f_s/2 = 1/(2T_s), \quad \omega_{nyq} = 2\pi f_{nyq} = \pi/T_s.$$

According to the fundamental Kotelnikov theorem [20], the maximum frequency component that can be reconstructed from a discrete signal is bounded by the Nyquist frequency. Therefore, it is important to choose an appropriate sampling frequency to accurately represent the plant model.

### *2.2. The Symmetry of the Discrete Fourier Transform*

In the definition of the $\mathcal{H}_\infty$-norm of the discrete MIMO transfer function, it was stated that all frequencies $\omega \in [0, \ldots, \omega_{nyq}]$. This follows from the symmetry of the discrete Fourier transform, which we briefly demonstrate below. A more detailed proof of the symmetry of the discrete Fourier transform can be found in Section 3.2 in [27].

The discrete Fourier transform $X[k]$ of a sequence of samples (digital signal) $x[n]$ is defined by the following equation:

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi kn/N), \quad k = f_k T_s = \frac{\omega_k}{2\pi} T_s, \quad k = 0, 1, \ldots, N-1,$$

where $k$ is the frequency index and $n = 0, 1, \ldots, N-1$ is the time index.

Let us compute the discrete Fourier transform for a frequency that is an integer multiple of the Nyquist frequency, i.e., for $k = k_{nyq}N$:

$$k_{nyq} = \frac{\omega_{nyq}}{2\pi} T_s = \frac{1}{2}, \quad k_{nyq}N = \frac{N}{2},$$

$$X[k_{nyq}N] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi k_{nyq}Nn/N) = \sum_{n=0}^{N-1} x[n] \exp(-j\pi)^n.$$

Using the Euler identity $\exp(j\pi) = \exp(-j\pi) = -1$, we obtain

$$X[k_{nyq}N] = \sum_{n=0}^{N-1} x[n](-1)^n.$$

This means that when computing $X[k]$, the frequency component $X[k_{nyq}N]$ will serve as a factor that ensures complex conjugation. That is, when we consider frequencies above the Nyquist frequency, we do not obtain any additional information since those frequency components are mirrored and can be represented by those below the Nyquist frequency. Therefore, it is sufficient to consider frequencies in the range $[0, \ldots, \omega_{nyq}]$ when computing the $\mathcal{H}_\infty$-norm of the discrete MIMO transfer function.

When we perform the synthesis of the discrete MIMO PID controller on the frequency range $[0, \ldots, \omega_{nyq}]$, we can say that "we are synthesizing a MIMO PID controller across the entire frequency range". Note that this cannot be said in the case of continuous-time MIMO PID synthesis, as described in [1], because then the synthesis will need to be performed on an infinite frequency range $[0, \ldots, \infty]$ (in practice, $[0, \ldots, \omega_n]$, where $\omega_n$ is a sufficiently large frequency).

### *2.3. The Small-Gain Theorem*

The small-gain theorem, initially proposed in [28], is a simple and intuitively understandable tool for computing robust stability margins in terms of $\mathcal{H}_\infty$-norms. To apply this theorem in robust analysis, the so-called $M - \Delta$ configuration (see Chapter 7 in [29] and Section 8.2 in [30]) is utilized, where the control system with uncertainties is divided into a nominal (unperturbed) part, represented by the transfer function $M(z)$, and a part with uncertainties (perturbed), represented by the transfer function $\Delta(z)$. Such a control system will be stable if and only if (see Section 10.4 in [31]):

$$\left\| M(z) \right\|_\infty \cdot \left\| \Delta(z) \right\|_\infty < 1.$$

We consider only two cases of uncertainties, which are sufficient for our goal (and in most practical cases), where only the plant model has output multiplicative uncertainty (Figure 3) and output additive uncertainty (Figure 4). The controller is assumed to be precisely known without uncertainties.

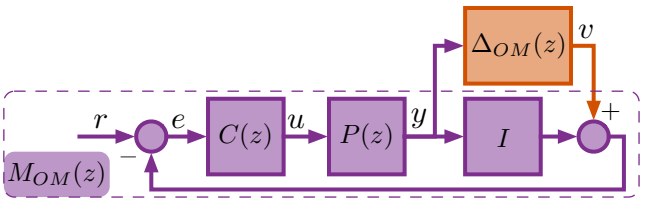



**Figure 3.** A control system with output multiplicative uncertainty in the plant model.

In the case of output multiplicative (OM) uncertainty, the transfer functions $M_{OM}(z)$ and $\Delta_{OM}(z)$ relate the vectors $v$ and $y$ as follows: $v = \Delta_{OM}(z)y$, $y = M_{OM}(z)v$, and the transfer function $M_{OM}(z)$ takes the form

$$M_{OM}(z) = -P(z)C(z)\Big(I + P(z)C(z)\Big)^{-1} = -\Big(I + P(z)C(z)\Big)^{-1}P(z)C(z). \tag{1}$$

Refer to Appendix A for the proof of this identity.

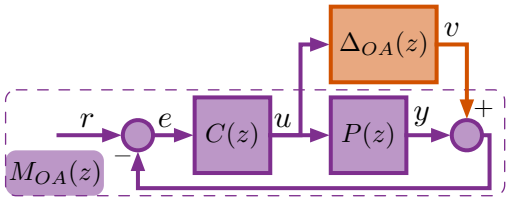



**Figure 4.** A control system with output additive uncertainty in the plant model.

In the case of output additive (OA) uncertainty, the transfer functions $M_{OA}(z)$ and $\Delta_{OA}(z)$ relate the vectors $v$ and $u$ as follows: $v = \Delta_{OA}(z)u$, $u = M_{OA}(z)v$, and the transfer function $M(z)$ takes the form

$$M_{OA}(z) = -C(z)\Big(I + P(z)C(z)\Big)^{-1}. \tag{2}$$

Thus, if the $\mathcal{H}_\infty$-norm of $M(z)$ in the control system is bounded, the value $1/\|M(z)\|_\infty$ will determine the margin of robust stability for the closed-loop system with uncertainties. This is because it serves as the bound for the $\mathcal{H}_\infty$-norm of $\Delta(z)$, representing the maximum gain of the possible feedback perturbations in the closed-loop system:

$$\|\Delta(z)\|_\infty < \frac{1}{\|M(z)\|_\infty}.$$

## 3. The Statement of the Problem

We aim to design the stationary discrete MIMO PID controller for the discrete linear non-stationary plant model. In this context, we have the LTI controller, while the plant will be described by an LTV model. Below, we demonstrate our approximation of the LTV plant model through an array of LTI models.

Through this design approach, we guarantee the robust stability margin and performance for all linear plant models included in the array. The proposed synthesis of the MIMO PID controller will be achieved by solving a system of LMIs. If a numerical solution exists for these LMIs, the performance and stability of the closed-loop system with the non-stationary plant model are guaranteed, even in the worst-case scenario.

### *3.1. The Plant Model*

The MIMO LTV plant model is generally represented as a linear continuous-time model in the state-space form:

$$\begin{cases} \dot{x}(t) = A(t)x(t) + B(t)u(t), \\ y(t) = C(t)x(t) + D(t)u(t), \end{cases} \tag{3}$$

where $y(t) \in \mathbb{R}^q$, $u(t) \in \mathbb{R}^m$, $x(t) \in \mathbb{R}^s$; $q$ represents the number of outputs; $m$ is the number of inputs; and $s$ is the number of states (model order). The state-space matrices $A(t)$, $B(t)$, $C(t)$, and $D(t)$ vary in time. We consider two of the most practically relevant cases, in our view, that allow representing the LTV model as an array of LTI models.

3.1.1. The LPV Model

The first case is the linear parameter-varying (LPV) model

$$\begin{cases} \dot{x}(t) = A\big(\theta(t)\big)x(t) + B\big(\theta(t)\big)u(t), \\ y(t) = C\big(\theta(t)\big)x(t) + D\big(\theta(t)\big)u(t), \end{cases} \tag{4}$$

where $\theta(t)$ is the function of time-varying plant parameters [32]. Here, it is important that the function $\theta(t)$ is deterministic, and therefore, the varying state-space matrices are deterministic (see Section 1.2 in [33]). Most often, an LPV model is obtained by linearizing a nonlinear model at several time points. It can also be derived through system identification methods [34] or when the state-space matrices contain parameters with variations explicitly defined over time.

In this case, we use the index $n$ to denote the time point $t_n$ at which the LTI model was calculated, obtaining an array of state-space matrices:

$$\theta_n = \theta(t_n), \quad \Big\{A_n = A(\theta_n),\ B_n = B(\theta_n),\ C_n = C(\theta_n),\ D_n = D(\theta_n)\Big\}.$$

The conditions described at the beginning of Section 3 (guaranteeing performance and robustness) will be satisfied if the interval for calculating the LTI models is equal to the sample time during its discretization. If this is challenging, our problem formulation requires $\theta(t)$ to be a linear function, leading to linear interpolation of parameters between time points $t_n$ during modeling. In any case, it is essential to consider how well the LPV model with linear interpolation of parameters will describe the dynamics of a real plant. Moreover, the grid on which the interpolation of the array of LTI models will be performed may be irregular.

3.1.2. The Uncertain State-Space Model

The second case is the uncertain state-space model

$$\begin{cases} \dot{x}(t) = A\big(\delta(t)\big)x(t) + B\big(\delta(t)\big)u(t), \\ y(t) = C\big(\delta(t)\big)x(t) + D\big(\delta(t)\big)u(t), \end{cases} \tag{5}$$

where $\delta(t)$ is the function that represents the uncertain plant parameters. Here, it is important that the function $\delta(t)$ is non-deterministic, but the bounds of the interval in which it can vary must be known. All state-space matrices of such a system must be representable in the following form:

$$A(\delta(t)) = A_0 + A_\Delta\big(\delta(t)\big), \quad A_\Delta\big(\delta(t)\big) = \delta_1(t)A_1 + \delta_2(t)A_2 + \cdots + \delta_{N_i}(t)A_{N_i}$$

where $A_0$ is the nominal part and $A_\Delta(\delta(t))$ is the perturbed part that can vary randomly within the bounds specified by $\delta(t)$ (see Section 4.3.1 in [35]).

Such a representation of the plant model is typical for situations where its parameters have measurement uncertainties. In this case, the perturbation parameters form a set represented by a regular polyhedron:

$$\Delta_I = \Big\{\delta(t) \mid \delta_i(t) \in [\delta_i^-, \delta_i^+],\ i = 1, \ldots, N_i\Big\},$$

with a set of extreme points: $\Delta_E = \Big\{\delta_i \mid \delta_i = \delta_i^- \text{ or } \delta_i^+,\ i = 1, \ldots, N_i\Big\}$.

If the perturbation parameters are normalized, they form a polytope:

$$\Delta_P = \Big\{\delta(t) \mid \sum_{i=1}^{N_i} \delta_i(t) = 1, \delta_i(t) \geq 0\Big\},$$

with a set of extreme points $\Delta_E = \left\{e_i^{\mathrm{T}} \,|e_i \in \mathbb{R}^{N_i},\ i = 1,\ldots,N_i\right\}$, where $e_i$ is the basis vector, with each $i$-th element equal to 1, while all the others are zero. The details of these representations are described in Section 4.3 in [35].

According to Theorem 4.3 in [35], to guarantee the performance and robustness of the closed-loop system, similar to what was proposed for the LPV model, it is sufficient to synthesize the controller on an array of LTI models calculated at the extreme points of the perturbation parameters set.

In this case, we use the index $n$ to denote the extreme points $\delta_n$ of the perturbation parameters set at which the LTI model was calculated, obtaining an array of state-space matrices:

$$\delta_n \in \Delta_E, \quad \Big\{A_n = A(\delta_n),\ B_n = B(\delta_n),\ C_n = C(\delta_n),\ D_n = D(\delta_n)\Big\}.$$

#### 3.1.3. The Array of Discrete Transfer Functions

Both of the described cases result in approximating the LTV plant model as an array of LTI models with an index $n$ for the synthesis of a robust discrete controller:

$$\begin{cases} \dot{x}_n(t) = A_n x_n(t) + B_n u_n(t), \\ y_n(t) = C_n x_n(t) + D_n u_n(t), \end{cases}$$

where $x_n(t)$, $u_n(t)$, and $y_n(t)$ are the state, input, and output vectors of the $n$-th LTI plant model, respectively. These cases can also be combined. In fact, by combining them, it is possible to represent any LTV model as an array of LTI models.

After that, it is necessary to discretize each LTI model in the array using the ZOH or FOH method, depending on the type of ADC and DAC used in the digital control system. When using the ZOH method (see Section 4.3.3 in [19]), we obtain

$$t(T_s k) \le t \le t(T_s k + T_s), k \in \mathbb{Z},$$

$$A_n^d = \exp(A_n T_s), \quad B_n^d = A_n^{-1}(A_n^d - I_s)B_n, \quad C_n^d = C_n, \quad D_n^d = D_n.$$

Here, the index $k$ denotes the sample index. Thus, we obtain an array of LTI discrete-time state-space plant models:

$$\begin{cases} x_n(T_s k + T_s) = A_n^d x(T_s k) + B_n^d u_n(T_s k), \\ y_n(T_s k) = C_n^d x_n(T_s k) + D_n^d u_n(T_s k). \end{cases}$$

As our method operates with transfer functions, we also derive an array of discrete transfer functions:

$$P_n(z) = C_n^d (z I_s - A_n^d)^{-1} B_n^d + D_n^d, \quad P_n(z) \in \mathbb{C}^{q \times m}, \quad n \in \mathbb{N}. \tag{6}$$

### *3.2. The Transfer Function of the Controller*

We employ the feedback connection scheme depicted in Figure 5, which is the classic scheme commonly used in tracking control systems. The scheme is based on the utilization of the MIMO PID controller, which provides feedback control to the output of the plant model.

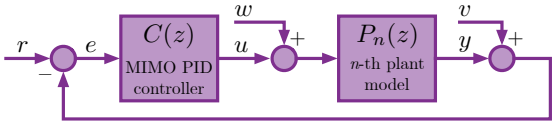



**Figure 5.** Classical feedback interconnection of reference tracking control system.

Here, the vector $r$ is the reference vector, the vector $e$ denotes the error signal, the vector $u$ is the plant control input, and the vector $y$ is the plant output. Additionally, the

vector $w$ represents the input disturbance vector and the vector $v$ represents the output disturbance vector.

The discrete LTI model of the MIMO PID controller in the control law

$$u(z) = C(z)\, e(z), \quad u(z) \in \mathbb{C}^m, \quad e(z) \in \mathbb{C}^q$$

is defined as a discrete transfer function in the form:

$$C(z) = K_P + K_I \frac{zT_s}{z-1} + K_D \frac{z-1}{zT_s}, \quad C(z) \in \mathbb{C}^{m \times q}, \tag{7}$$

where the MIMO PID controller coefficients $\{K_P, K_I, K_D\} \in \mathbb{R}^{m \times q}$ are arbitrary (non-diagonal) matrices. An evident requirement for MIMO tracking control systems is that the number of inputs ($m$) and outputs ($q$) of the plant should satisfy

$$m \geq q.$$

Here, we use the backward Euler numerical method

$$\frac{zT_s}{z-1}$$

for both discrete differentiation and integration. We can use another method for integration and differentiation, such as

$$\begin{aligned} \text{Forward Euler}: &\quad \frac{T_s}{z-1}, \\ \text{or Tustin (Trapezoidal)}: &\quad \frac{T_s}{2} \cdot \frac{z+1}{z-1}. \end{aligned}$$

Details of these methods can be found in Section 6.1 in [19]. However, it is advisable to avoid using the forward Euler method in the differentiator since it would require introducing a derivative-filter time constant. This constant cannot be a tunable parameter in our problem formulation due to nonlinearity, and it would need to be manually specified as a scalar or matrix in case different derivative filter time constants are required in each control loop. The use of the backward Euler method in the differentiator, apart from not requiring the use of a derivative-filter time constant, also allows for implementing finite control in the differentiator, as it sets some of the closed-loop poles to zero $z = 0$. This is somewhat analogous to "deadbeat control" where all closed-loop poles are set to zero. In the continuous system, this is equivalent to the unrealizable case when the pole $s = -\infty$.

Moreover, we can add or remove components of the MIMO PID controller; for example, we can remove components $K_P$ and $K_D$ when not needed, turning the "PID" controller into "PI", "ID", or "I". We can add integrator or differentiator components of the second order and higher. Our problem formulation explicitly requires the presence of an integrator, which is natural since without it, minimizing the error vector in the steady state in a MIMO system would be impossible.

### *3.3. The Array of Discrete Transfer Functions*

In accordance with the diagram in Figure 5, we define the following array of discrete transfer functions:

- The open-loop transfer function $L(z)$, which is a transfer function from the error vector $e$ to the output vector $y$

$$y(z) = L_n(z)e(z), \quad L_n(z) = P_n(z)C(z), \quad y(z) \in \mathbb{C}^q, \quad L_n(z) \in \mathbb{C}^{q \times q},$$

  where $P_n(z)$ represents the $n$-th plant model from the array of linear plant models (6).

- The sensitivity function $S(z)$, which is a closed-loop transfer function from the reference vector $r$ to the error vector $e$

$$e(z) = S_n(z)r(z), \quad S_n(z) = \Big(I_q + L_n(z)\Big)^{-1}, \quad S_n(z) \in \mathbb{C}^{q\times q}.$$

- The complementary sensitivity function $T(z)$, which is a closed-loop transfer function from the reference vector $r$ to the output vector $y$

$$y(z) = T_n(z)r(z), \quad T_n(z) = I_q - S_n(z) = L_n(z)S_n(z), \quad T_n(z) \in \mathbb{C}^{q\times q}.$$

- The static and low-frequency sensitivity function $F(z)$, which is the sensitivity function for small frequencies. If $C(z)$ is given by (7), it takes the form:

$$F_n(z) = \Big(P_n(1)K_I T_s\Big)^{-1}(z-1), \quad F_n(z) \approx S_n(z) \text{ for small } \omega,\ z = \exp(j\omega T_s), \tag{8}$$

  where $P_n(1)$ is the static gain of the $n$-th plant model, which means the frequency response at zero frequency ($\omega = 0$). See Appendix B for a detailed explanation of the static and low-frequency sensitivity function.
- The Q-parameter (a term from the Youla parametrization approach [36]), is a closed-loop transfer function from the reference vector $r$ to the input vector $u$

$$u(z) = Q_n(z)r(z), \quad Q_n(z) = C(z)S_n(z), \quad Q_n(z) \in \mathbb{C}^{m\times q}.$$

- The closed-loop transfer function from the input disturbance vector $w$ to the error vector $e$ is:

$$e(z) = W_n(z)w(z), \quad W_n(z) = -S_n(z)P_n(z), \quad w(z) \in \mathbb{C}^m, \quad W_n(z) \in \mathbb{C}^{q\times m}.$$

- The closed-loop transfer function from the output disturbance vector $v$ to the error vector $e$ is:

$$e(z) = V_n(z)v(z), \quad V_n(z) = -S_n(z), \quad v(z) \in \mathbb{C}^q, \quad V_n(z) \in \mathbb{C}^{q\times q}.$$

### *3.4. The Performance Criteria*

Using these closed-loop transfer functions, we formulate the performance criteria [31] of the closed-loop control system with the MIMO PID controller in terms of the $\mathcal{H}_\infty$-norm constraints on the closed-loop transfer functions.

The restriction of the $\mathcal{H}_\infty$-norm of the sensitivity function $S_n(z)$ guarantees the performance of the closed-loop system for all plant models $P_n(z)$. This also determines the maximum overshoot in the closed-loop control system.

The restriction of the $\mathcal{H}_\infty$-norm of the complementary sensitivity function $T_n(z)$ guarantees the robust stability of the closed-loop system for all plant models $P_n(z)$. Additionally, according to the small-gain theorem, the value $1/\|T(z)\|_\infty$ determines the margin of robust stability for the control system in the case of output multiplicative uncertainties in the plant model (1) since

$$\|M_{OM}(z)\|_\infty = \|T(z)\|_\infty \tag{9}$$

and serves as a bound for the maximum gain of the possible output multiplicative disturbances in the plant model

$$\|\Delta_{OM}(z)\|_\infty < \frac{1}{\|T(z)\|_\infty}.$$

The transfer functions $S(z)$ and $T(z)$ are related by the fundamental equation

$$S(z) + T(z) = I_q.$$

In a stable closed-loop system with a small steady-state tracking error, it holds that

$$\|S(\exp(0))\|_2 = \left\|T(\exp(j\omega_{nyq}T_s))\right\|_2 \approx 0, \quad \left\|S(\exp(j\omega_{nyq}T_s))\right\|_2 = \|T(\exp(0))\|_2 \approx 1.$$

In the ideal case, $\|S(z)\|_\infty = \|T(z)\|_\infty = 1$, which means perfect tracking across the entire frequency range. In reality, such a restriction would be too stringent and may require excessively large input signals. Rational values for $\|S(z)\|_\infty$ and $\|T(z)\|_\infty$ typically range from 1.1 to 1.8.

By applying a constraint on the $\mathcal{H}_\infty$-norm of $Q(z)$, we limit the magnitude of the input of the plant model. This is useful in accounting for physical limitations, such as those in the actuator devices. Additionally, according to the small-gain theorem, the value $1/\|Q(z)\|_\infty$ determines the margin of robust stability for the control system in the case of output additive uncertainties in the plant model (2) since

$$\|M_{OA}(z)\|_\infty = \|Q(z)\|_\infty \tag{10}$$

and serves as a bound for the maximum gain of the possible output additive disturbances in the plant model

$$\|\Delta_{OA}(z)\|_\infty < \frac{1}{\|Q(z)\|_\infty}.$$

The rational values of $\|Q(z)\|_\infty$ should be multiples of $1/\underline{\sigma}(P(1))$ (see Section 3.1 in [1]).

By constraining the $\mathcal{H}_\infty$-norms of $W(z)$ and $V(z)$, we limit the effect of the input and output disturbance vectors on the error vector. Note that there is no need to constrain the $\mathcal{H}_\infty$-norm of $V_n(z)$ because

$$\|V_n(z)\|_\infty = \| - S_n(z)\|_\infty = \|S_n(z)\|_\infty.$$

We aim to achieve the lowest possible settling time for the closed-loop system while adhering to the constraints of the $\mathcal{H}_\infty$-norm of other transfer functions. We cannot constrain the $\mathcal{H}_\infty$-norm of $F(z)$ since, as defined in (8), an unbounded increase in $z$ results in an infinite value of $\|F(z)\|_\infty$. Instead, we minimize the spectral norm of the matrix

$$\left(P_n(1)K_I T_s\right)^{-1}, \tag{11}$$

which we hereafter refer to as the matrix of the static and low-frequency sensitivity (SLFS). This will result in attaining the best possible static and low-frequency sensitivity, which requires minimizing

$$\min \left\| \left(P_n(1)K_I T_s\right)^{-1} \right\|_2.$$

Minimizing the spectral norm of the SLFS matrix, along with constraining the $\mathcal{H}_\infty$-norm of $S(z)$ and $T(z)$, ensures the decoupling of control loops in the closed-loop system, at least at low frequencies. This is discussed in more detail in Section 4.6.

### *3.5. The SDP Formulation*

We can now formulate the problem of synthesizing a MIMO PID controller as the semi-definite programming (SDP) problem:

$$\begin{cases} \min & \|(P_n(1)K_I T_s)^{-1}\|_2, \\ \text{s.t.} & \|S_n(z)\|_\infty \le S_{\max}, \\ & \|T_n(z)\|_\infty \le T_{\max}, \\ & \|Q_n(z)\|_\infty \le Q_{\max}, \\ & \|W_n(z)\|_\infty \le W_{\max}, \\ \text{for each} & n. \end{cases} \tag{12}$$

The problem statement (12) repeats the one used in [1] for continuous-time systems. Such a problem statement suffices for synthesizing a control system with the maximum possible performance while ensuring robust stability margins in terms of the small-gain theorem (see Section 2.3). However, the formulation (12) does not ensure the direct placement of the open-loop transfer function $L(z)$ within a strictly defined frequency range, hindering, for instance, the specification of the bandwidth of the closed-loop system. To overcome this, we extend the statement with a loop-shaping procedure, thereby revealing the possibilities associated with frequency-dependent bounds, implicitly pointed out in Section 7 in [1].

## 4. Loop Shaping

As we mentioned in Section 1, the loop-shaping approach involves defining the performance requirements and robust stability margins of the closed-loop control system in terms of the singular values of the open-loop transfer function [37]. Using loop shaping in controller synthesis allows for direct manipulation of the frequency characteristics of the system. The essence of loop shaping lies in synthesizing a control system with a constraint on the maximum gain of the closed-loop transfer functions in a defined frequency region. The functions that limit the maximum gain of the transfer functions within the closed-loop system are commonly referred to as "shape functions". Thus, it is possible to synthesize a closed-loop system with a desired crossover region, localize the range of the $\mathcal{H}_\infty$-norm placement of the closed-loop transfer functions, limit the $\mathcal{H}_\infty$-norms themselves, and achieve the desired bandwidth with a desired trade-off between performance and robustness.

### *4.1. The Frequency Region*

We begin by defining the frequency region $\Omega$ within which we will perform the synthesis of the discrete controller

$$\Omega = \{\omega \mid 0 < \omega_0 \leq \omega \leq \omega_{nyq}\}. \tag{13}$$

Due to the symmetry of the discrete Fourier transform (see Section 2.2), this frequency region is bounded by the Nyquist frequency $\omega_{nyq}$, and $\omega_0$ is a sufficiently small frequency. $\omega_0 \neq 0$ because, in this case, we would obtain infinite values for the frequency response of the transfer function with unit poles ($z = 1$). For example, the MIMO PID discrete transfer function always has unit poles, which correspond to an integrator (7). A typical value for $\omega_0$ is $10^{\lg(T_s)}$.

The MIMO closed-loop system with $q$ outputs has $q$ gain crossover frequencies $\omega_{c\,j}$, which are defined as the frequencies at which the singular values of the open-loop transfer function $L(z)$ first cross the 1-magnitude (0 dB) level from above:

$$\sigma_1\Big(L(\omega_{c\,1})\Big) = 1, \ldots, \sigma_q\Big(L(\omega_{c\,q})\Big) = 1.$$

They also determine the time-domain characteristics of the transient response for each control loop (see Section 4.9 in [3]). Now, we define the crossover region with the boundaries $\omega_b^{\min}$ and $\omega_b^{\max}$:

$$\omega_b^{\min} \leq \omega_{c\,1} \leq \cdots \leq \omega_{c\,q} \leq \omega_b^{\max}.$$

So, the frequency region is divided into three parts $\Omega = \Omega_L \cup \Omega_C \cup \Omega_H$:

$$\begin{aligned} \text{low frequencies:} \quad & \Omega_L = \{\omega \mid \omega_0 \quad\;\; \leq \omega < \omega_b^{\min}\}; \\ \text{desired crossover region:} \quad & \Omega_C = \{\omega \mid \omega_b^{\min} \leq \omega \leq \omega_b^{\max}\}; \\ \text{high frequencies:} \quad & \Omega_H = \{\omega \mid \omega_b^{\max} < \omega \leq \omega_{nyq}\}. \end{aligned} \tag{14}$$

*4.2. The Shape Function*

Unlike classical loop shaping, where shape functions are defined as weighting functions [37], we are free to specify the shape functions in any form. This is one of the most important advantages of our methodology. Moreover, we can specify the shape functions for any transfer functions of the closed-loop system.

For the array of closed-loop transfer functions

$$G_n(z) = \Big\{ S_n(z), T_n(z), Q_n(z), W_n(z) \Big\},$$

we define the set of shape functions

$$b_G(\omega) = \Big\{ b_S(\omega), b_T(\omega), b_Q(\omega), b_W(\omega) \Big\}$$

such that for any $n$

$$\Big\| G_n\Big( \exp(j\omega T_s) \Big) \Big\|_2 \leq b_G(\omega), \quad \omega \in \Omega, \quad b_G(\omega) \in \mathbb{R}^1$$

and for the $\mathcal{H}_\infty$-norms of $G_n(z)$, it holds that

$$\Big\| G_n(z) \Big\|_\infty \leq G_{\max}, \quad G_{\max} = \sup_\omega \Big\{ b_G(\omega) \Big\}, \quad G_{\max} = \Big\{ S_{\max}, T_{\max}, Q_{\max}, W_{\max} \Big\}.$$

Here are examples of some of the most commonly requested shape functions:

- Keep a constant, maintaining the problem formulation as (12):

$$b_G(\omega) = G_{\max}, \quad \omega \in \Omega$$

- Restrict the desired crossover region, limiting where the $\mathcal{H}_\infty$-norms of the closed-loop system transfer functions can reside:

$$b_G(\omega) = \begin{cases} 1, & \omega \in \Omega_L \\ G_{\max}, & \omega \in \Omega_C \\ 1, & \omega \in \Omega_H \end{cases}, \tag{15}$$

- Limit the desired crossover region only from the high-frequency side, restricting only the bandwidth of the closed-loop system:

$$b_G(\omega) = \begin{cases} G_{\max}, & \omega \in \Omega_L \cup \Omega_C \\ 1, & \omega \in \Omega_H \end{cases},$$

- Guarantee the placement of the singular values of the closed-loop transfer functions at high frequencies:

$$b_G(\omega) = (\omega_b^{\max}/\omega)^\alpha, \quad \omega \in \Omega_H, \tag{16}$$

or at low frequencies:

$$b_G(\omega) = \Big( \omega/\omega_b^{\min} \Big)^\alpha, \quad \omega \in \Omega_L.$$

These relationships are derived through a slight modification of the Butterworth filter, where $\alpha$ denotes the filter order.

- Another interesting way to set the shape function as the gain of the Chebyshóv filter of the first kind (see Section 6.2 in [38]), is to slightly modify it:

$$b_G(\omega) = \frac{1}{\sqrt{1+\epsilon^2 \cos(\alpha \arccos(\omega/\omega_b^{\max}))^2}} + G_{max} - 1, \tag{17}$$

where

$$\epsilon = \sqrt{1/(2-G_{max})^2 - 1}$$

is the ripple factor and $\alpha$ is the filter order. One notable drawback of this filter is ripples in the passband. However, in our context, these ripples can be advantageous, as they allow for precise placement of the gain of the closed-loop transfer function in some cases.

- One more interesting result can be obtained using the Zolotarev filter (elliptic filter):

$$b_G(\omega) = \frac{1}{\sqrt{1+\epsilon^2 R_\alpha^2(\xi, \omega/\omega_b^{\max})}}$$

where $R_\alpha$ is the $\alpha$-th order Chebyshóv rational function, $\epsilon$ is the ripple factor, and $\xi$ is the selectivity factor.

In the presented examples, we defined the shape functions in three frequency regions, but if needed, more regions can be specified. There are also no limitations on the form of the shape functions. Figure 6 shows three examples of shape functions, with the parameters $G_{\max} = 1.4$, $\omega_b^{\min} = 100$ Hz, and $\omega_b^{\max} = 1$ kHz; $b_{G1}(\omega)$ has the form (15),

$$b_{G2}(w) = \begin{cases} G_{\max}, & \omega \in \Omega_L \cup \Omega_C, \\ (\omega_b^{\max}/\omega)^\alpha, & \omega \in \Omega_H, \end{cases}$$

and $b_{G3}(\omega)$ has the form (17).

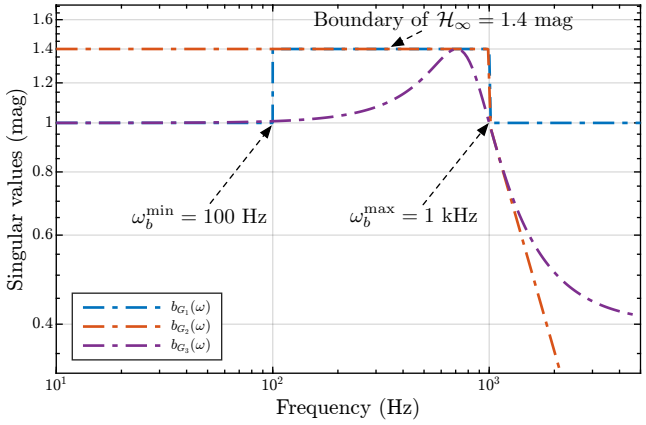



**Figure 6.** Examples of three shape functions.

### *4.3. The Trade-Off between Performance and Robustness*

The problem of achieving a trade-off between performance and robustness is based on the fact that, by definition, the sensitivity function $S(z)$ approximates the open-loop transfer function $L(z)$ at low frequencies, whereas a complementary sensitivity function $T(z)$ approximates it at high frequencies if the following conditions are satisfied:

$$\text{for} \quad \omega \in \Omega_L, \quad \text{if} \quad \underline{\sigma}\Big(L\big(\exp(j\omega T_s)\big)\Big) \gg 1, \quad \overline{\sigma}\Big(L^{-1}\big(\exp(j\omega T_s)\big)\Big) \ll 1:$$

$$S(z) = \Big(I_q + L(z)\Big)^{-1} \approx L^{-1}(z), \quad T(z) = I_q - S(z) \approx I_q, \tag{18}$$

$$\Big\|S^{-1}\Big(\exp(j\omega T_s)\Big)\Big\|_2 \approx \Big\|L\Big(\exp(j\omega T_s)\Big)\Big\|_2,$$

$$\text{for} \quad \omega \in \Omega_H, \quad \text{if} \quad \overline{\sigma}\Big(L\big(\exp(j\omega T_s)\big)\Big) \ll 1:$$
$$T(z) = L(z)\Big(I_q + L(z)\Big)^{-1} \approx L(z), \quad S(z) = I_q - T(z) \approx I_q, \tag{19}$$
$$\Big\|T\Big(\exp(j\omega T_s)\Big)\Big\|_2 \approx \Big\|L\Big(\exp(j\omega T_s)\Big)\Big\|_2.$$

Thus, the constraint on $S(z)$ in $\Omega_L$ determines the performance of the closed-loop system. Similarly, the constraint on $T(z)$ in $\Omega_H$ determines the robustness margin of the closed-loop system. These constraints are commonly referred to as the "performance bound" and the "robustness bound", respectively. We suggest calling the regions they bound the *performance area* and the *robustness area*. We illustrate them in Figure 7. Note that the performance area is always symmetric.

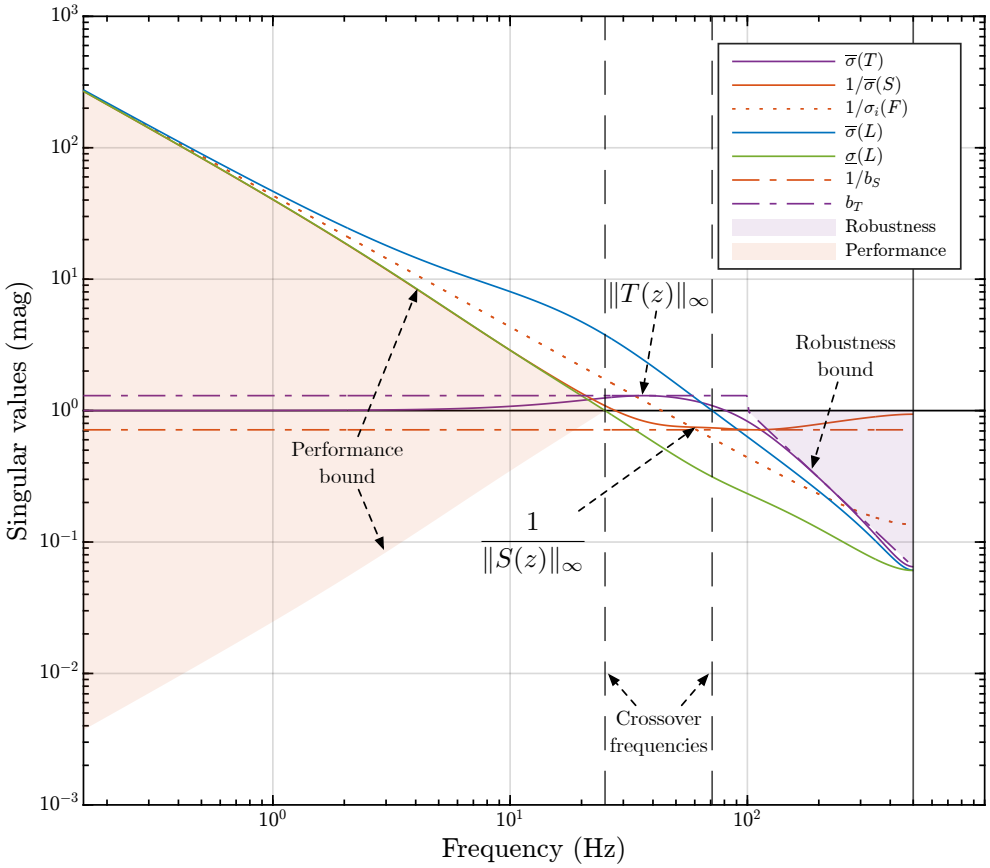



**Figure 7.** An example illustrating loop shaping. The singular values of the transfer functions $L(z)$, $S(z)$, and $T(z)$ of the MIMO control system. The performance area and robustness area are shaded.

### *4.4. The Robustness Bound*

We denote the robustness bound as $b_R(\omega)$. As mentioned above in (19), by the robustness bound, we mean the shape function for $T(z)$ at high frequencies:

$$b_R(\omega) = b_T(\omega) \quad \text{for} \quad \omega \in \Omega_H. \tag{20}$$

The robustness bound is the lower border for the robustness area, which we shaded in purple in Figure 7. The value of the robustness area

$$A_R = \int_{\omega_b^{\max}}^{\omega_{nyq}} \Big(1 - b_R(\omega)\Big) d\omega \tag{21}$$

provides a numerical estimate of the guaranteed robust stability margin of the closed-loop system. For example, if the shape function $b_T(\omega)$ for $\omega \in \Omega_H$ takes the form (16), the robustness bound will take the form:

$$b_R(\omega) = (\omega_b^{\max}/\omega)^{\alpha},$$

and then the robustness area is calculated by

$$A_R = \int_{\omega_b^{\max}}^{\omega_{nyq}} \Big(1 - b_R(\omega)\Big) d\omega = \omega_{nyq} - \frac{\omega_{nyq}(\omega_b^{\max}/\omega_{nyq})^{\alpha}}{1-\alpha} + \frac{\omega_b^{\max}}{1-\alpha} - \omega_b^{\max}, \quad \alpha > 1.$$

Thus, we guarantee that the singular values of the array of the transfer functions $T_n(z)$ of the synthesized system will be bounded over the entire frequency region $\Omega$

$$\Big\| T_n\Big(\exp(j\omega T_s)\Big)\Big\|_2 \le b_T(\omega), \quad \omega \in \Omega, \tag{22}$$

consequently preventing the singular values of the open-loop transfer function array $L_n(z)$ from entering the robustness area at high frequencies:

$$\Big\| L_n\Big(\exp(j\omega T_s)\Big)\Big\|_2 \le b_R(\omega), \quad b_T(\omega) = b_R(\omega) \quad \text{for} \quad \omega \in \Omega_H. \tag{23}$$

Throughout this paper, the index $n$ denotes the plant model number in the array (6).

### *4.5. The Performance Bound*

In our methodology, the performance bound carries a more intricate meaning. Unlike the robustness bound, we cannot define it in a similarly straightforward manner because our controller synthesis is iterative. The sensitivity function of the closed-loop system undergoes significant changes with each iteration. The necessity of iterative controller synthesis is explained in Section 5.2. Therefore, the shape function for $S(z)$ at low frequencies should always be equal to one:

$$b_S(\omega) = 1 \quad \text{for} \quad \omega \in \Omega_L.$$

We start by defining the small-frequency region $\Omega_S$ as a subset of the low-frequency region $\Omega_L$ (14):

$$\Omega_S = \{\omega \mid \omega_0 \le \omega < \omega_s < \omega_b^{\min}\}, \quad \Omega_S \subset \Omega_L, \tag{24}$$

where $\omega_s$ is the boundary of the small-frequency region.

In Section 3, we introduced the static and low-frequency sensitivity (SLFS) function $F(z)$ (8). Furthermore, we justified the necessity of minimizing it through the minimization of the spectral norm of the SLFS matrix (11). According to its definition, it should approximately hold for small frequencies:

$$F(z) \approx S(z), \quad \omega \in \Omega_S.$$

This implies that for their spectral norms (maximum gains), the following holds:

$$\Big\| S\Big(\exp(j\omega T_s)\Big)\Big\|_2 \approx \Big\| F\Big(\exp(j\omega T_s)\Big)\Big\|_2, \quad \omega \in \Omega_S. \tag{25}$$

The function $F(z)$, as defined in (8), is linear by definition. Therefore, utilizing its singular values in the low-frequency region $\Omega_L$ as the performance bound would provide an overly optimistic estimate of the performance area. Therefore, we define the performance bound $b_P(\omega)$ as the minimum singular value of the open-loop transfer functions $L(z)$ at $\Omega_L$, in accordance with (18). Given that we are working with an array of models, we opt for

the worst case, selecting the open-loop transfer function with the smallest singular values from the array $L_n(z)$:

$$b_P(\omega) = \min_n \left\{ \underline{\sigma}\Big( L_n\big( \exp(j\omega T_s) \big) \Big) \right\}, \quad \omega \in \Omega_L. \tag{26}$$

Since our target of the SDP problem in (12) is minimizing the spectral norm of the SLFS matrix and this minimization will be performed iteratively, the performance bound will be floating, and with each iteration, the performance area that is bounded by the performance bound will increase. The performance area is a symmetric area bounded at the bottom by the function $b_P^{-1}(\omega)$ and at the top by the function $b_P(\omega)$. The value of the performance area is defined as

$$A_P = 2 \int_{\omega_0}^{\omega_b^{\min}} \Big( 1 - b_P(\omega) \Big) d\omega. \tag{27}$$

The objective of iteratively minimizing the spectral norm of the SLFS matrix is to maximize the slope of

$$\overline{\sigma}\Big( F\big( \exp(j\omega T_s) \big) \Big)$$

and approach a condition number to unity

$$\kappa\Big( F\big( \exp(j\omega T_s) \big) \Big) \approx 1.$$

Thus, by iteratively minimizing the spectral norm of the SLFS matrix, it can be guaranteed that the performance area will iteratively increase. Figure 7 shows the singular values of the transfer functions of the system for which the spectral norm of the SLFS matrix has been reasonably well minimized.

It should be noted that in Figure 7, we present the inverse singular values for $S(z)$ and $F(z)$ to emphasize that

$$S(z) \approx F(z) \approx L^{-1}(z) \quad \text{or} \quad S^{-1}(z) \approx F^{-1}(z) \approx L(z) \quad \text{for} \quad \omega \in \Omega_S,$$

and we have shaded the performance area in red. In this case, the boundary of the small-frequency region $\omega_s$ (24) is approximately 1 Hz.

### *4.6. Guarantee of Control-Loop Decoupling*

It is important to explain why our problem statement avoids the prior decoupling of control loops and how the problem statement itself provides the decoupling of loops in a closed-loop control system. For tracking control systems, it is necessary to ensure that the control loops are decoupled in the steady state, so it is sufficient for us to ensure good decoupling only at low frequencies. Decoupling the control loops in a MIMO system over the entire frequency range is generally impossible and not required.

The closed-loop system is decoupled at the frequency $\omega$ when the frequency response of the complementary sensitivity function has a diagonal form (see Section 3.4.1 in [29]). Understanding this is simpler in terms of the condition number and the Relative Gain Array (RGA) matrix (see Section 3.6 in [29]).

The condition number of the transfer function $T(z)$ at the frequency $\omega$ is defined as the ratio of the maximum and minimum singular values of its frequency response at the frequency $\omega$:

$$\kappa\Big( T\big( \exp(j\omega T_s) \big) \Big) = \frac{\overline{\sigma}\Big( T\big( \exp(j\omega T_s) \big) \Big)}{\underline{\sigma}\Big( T\big( \exp(j\omega T_s) \big) \Big)}.$$

A convenient numerical criterion for assessing control-loop decoupling is the Relative Gain Array (RGA) matrix (see Section A.4.2 in [29]):

$$\mathrm{RGA}\Big(T\big(\exp(j\omega T_s)\big)\Big) = T\big(\exp(j\omega T_s)\big) \times \Big(T\big(\exp(j\omega T_s)\big)^{\dagger}\Big)^{\mathrm{H}}, \tag{28}$$

where $\times$ denotes the Schur product (element-by-element multiplication). If the control loops are completely decoupled at the frequency $\omega$, the RGA matrix at that frequency will be the identity matrix.

If the condition number at the frequency $\omega$ is equal to one, the frequency response $T\big(\exp(j\omega T_s)\big)$ at the frequency $\omega$ is a unitary matrix. Consequently, all its singular values are equal to one, and therefore, the RGA matrix at the frequency $\omega$ will be an identity matrix. The same is true for the approximation:

$$\kappa\Big(T\big(\exp(j\omega T_s)\big)\Big) \approx 1 \quad \text{then} \quad \mathrm{RGA}\Big(T\big(\exp(j\omega T_s)\big)\Big) \approx I_q. \tag{29}$$

Minimizing the spectral norm of the SLFS matrix ensures that at small frequencies, the condition number of the transfer function $S(z)$ is close to one. This property also holds for $L(z)$ and $T(z)$. For $L(z)$, this fact can be observed in Figure 7. Additionally, we illustrate this in Figure 8, which shows the singular values of the control system in Figure 7. The difference is that for $S(z)$ and $T(z)$, all singular values are shown. The blue line in Figure 8 represents the condition number of $T(z)$. It can be seen that in a closed-loop system with a sufficiently well-minimized spectral norm of the SLFS matrix, the decoupling of channels is disrupted only at high frequencies.

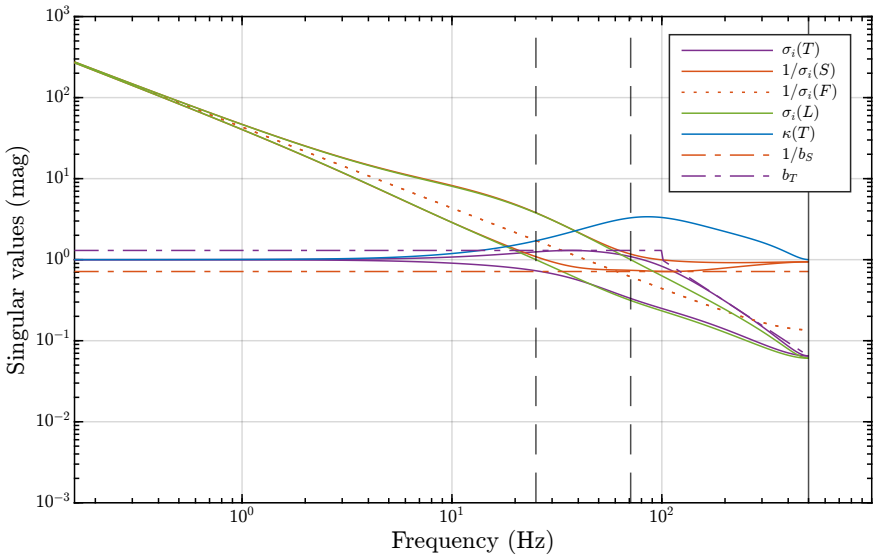



**Figure 8.** The singular values of the transfer functions $L(z)$, $S(z)$, and $T(z)$ of the MIMO control system in Figure 7 with the addition of the condition number for $T(z)$.

### *4.7. The Problem Statement with Loop Shaping*

We are finally ready to add loop shaping to the original problem statement (12):

$$\begin{cases} \min & \left\| \Big(P_n(1)K_I T_s\Big)^{-1} \right\|_2, \\ \text{s.t.} & \left\| S_n\Big(\exp(j\omega T_s)\Big) \right\|_2 \le b_S(\omega), \\ & \left\| T_n\Big(\exp(j\omega T_s)\Big) \right\|_2 \le b_T(\omega), \\ & \left\| Q_n\Big(\exp(j\omega T_s)\Big) \right\|_2 \le b_Q(\omega), \\ & \left\| W_n\Big(\exp(j\omega T_s)\Big) \right\|_2 \le b_W(\omega), \\ \text{for each} & n. \end{cases} \tag{30}$$

This problem formulation can be easily reduced back to the form in (12) by defining all shape functions as constants.

The transfer functions of the closed-loop system are interrelated, obviating the necessity of imposing "extremely strict" constraints on the maximum gains of all transfer functions. Imposing such "extremely strict" constraints may lead to the system (30) having no numerical solution. A more judicious strategy involves contextual considerations, introducing constraints selectively where they are critical to the reliable operation of a closed-loop control system.

Let us summarize the problem statement we described in Sections 3 and 4:

- We obtain an optimal controller, meaning that the closed-loop system has the maximum possible performance area (27) under constraints on $Q(z)$ and $W(z)$, along with a specified robust stability margin, in terms of the small-gain theorems (9) and (10), as well as the defined robustness area (21).
- In our view, the presence of a "floating" performance bound is a significant advantage over classical loop shaping, where both bounds are strictly defined, preventing the synthesis of an optimal controller.
- The correct choice of the robustness bound $b_R(\omega)$ is particularly important in digital systems, as it can provide a sufficiently large robustness area, thus guaranteeing the absence of aliasing near the Nyquist frequency. Consequently, there is no need to include an anti-aliasing filter in the feedback loop.
- By varying the robustness bound, we make a trade-off between the performance and robust stability margin of the closed-loop system. A larger robustness area leads to a smaller performance area, and vice versa.
- By minimizing the spectral norm of the SLFS matrix, we simultaneously achieve two objectives: ensuring the decoupling of control loops at low frequencies and maximizing control performance (performance area). Additionally, we obtain a closed-loop system with crossover frequencies as close as possible. Thus, the closed-loop system exhibits nearly identical time responses for its control loops.
- We synthesize an LTI controller for an LTV plant model, represented as a transfer function array with index $n$ (6). If the system of inequalities (30) has a solution, all specified constraints will be satisfied for the entire transfer function array of the closed-loop system with an LTV plant model (6).

## 5. The Synthesis Method

We aim to determine the unknown parameters of the MIMO PID controller, which is a fixed-structure controller. Additionally, we are utilizing the output feedback. Therefore, the standard LMI approach using the Bounded Real Lemma (Theorem 5.3 in [35]) to bound the $\mathcal{H}_\infty$-norm, which is typically effective for synthesizing static state feedback, cannot be applied in our case.

The concept of the $\mathcal{H}_\infty$-norm of a matrix function is a generalization of the concept of the spectral norm of a matrix to matrix complex functions, as can be seen from the definitions provided in Section 2.1. This fact enables us to reformulate the problem of constraining the $\mathcal{H}_\infty$-norm of the transfer function $G(z)$ as the problem of constraining a sufficiently large set of spectral norms of frequency responses $G\big(\exp(j\omega_k T_s)\big)$ on a finite set of frequencies $\Omega_k$:

$$\left\|G(z)\right\|_\infty = \sup_\omega \left\{\overline{\sigma}\Big(G\Big(\exp(j\omega T_s)\Big)\Big)\right\} \longrightarrow \left\|G\Big(\exp(j\omega_k T_s)\Big)\right\|_2 = \overline{\sigma}\Big(G\Big(\exp(j\omega_k T_s)\Big)\Big),$$

where $\omega_k \in \Omega_k$,

$$\Omega_k = \{\omega_k \mid \omega_k \in \Omega, \quad k = 1, 2, \ldots, N_k\}, \quad |\Omega_k| = N_k. \tag{31}$$

The value $N_k$ represents the cardinality of $\Omega_k$, the number of frequencies at which the controller will be synthesized, and $\Omega$ is defined in (13).

We use the index $k$ to denote the frequency response at frequency $\omega_k$. Applying this to the $n$-th plant model

$$P_{nk} = P_n\Big(\exp(j\omega_k T_s)\Big)$$

and the transfer function of the MIMO PID controller (7)

$$C_k = K_P + K_I\frac{\exp(j\omega_k T_s)T_s}{\exp(j\omega_k T_s)-1} + K_D\frac{\exp(j\omega_k T_s)-1}{\exp(j\omega_k T_s)T_s}, \tag{32}$$

as well as all other transfer functions, and for the shape functions of these transfer functions

$$b_{Sk} = b_S(\omega_k),\ b_{Tk} = b_T(\omega_k),\ b_{Qk} = b_Q(\omega_k),\ b_{Wk} = b_W(\omega_k).$$

We obtain the new statement of the problem:

$$\begin{cases} \min & \left\| \Big(P_n(1)K_I T_s\Big)^{-1} \right\|_2, \\ \text{s.t.} & \|S_{nk}\|_2 \le b_{Sk}, \\ & \|T_{nk}\|_2 \le b_{Tk}, \\ & \|Q_{nk}\|_2 \le b_{Qk}, \\ & \|W_{nk}\|_2 \le b_{Wk}, \\ \text{for each} & n,k. \end{cases}$$

### *5.1. The QMI Form*

First, we introduce the scalar $\gamma$ using the standard epigraph transformation method from Section 4.2.4 in [9]:

$$\min \left\| \Big(P_n(1)K_I T_s\Big)^{-1} \right\|_2 \iff \begin{cases} \min \gamma \\ \left\| \Big(P_n(1)K_I T_s\Big)^{-1} \right\|_2 \le \gamma. \end{cases}$$

To address the problem of constraining the spectral norm of the matrix in the system, we utilize the generalized eigenvalue problem from Section 3.3 in [9], which allows us to rewrite the constraint in terms of a matrix inequality:

$$\|A\|_2 \le \gamma \iff A^{\mathrm{H}}A \preceq \gamma^2 I_m, \quad A \in \mathbb{C}^{q\times m}.$$

We use the following abbreviations

$$Z_{nk} = I + L_{nk} = I + P_{nk}C_k = S_{nk}^{-1}, \quad Z_{nk} \in \mathbb{C}^{q\times q}, \tag{33}$$

and

$$X_n = P_n(1)K_I, \quad X_n \in \mathbb{R}^{q\times q}, \tag{34}$$

to avoid inverting matrices:

$$\begin{aligned} \left\| \Big(P_n(1)K_I T_s\Big)^{-1} \right\|_2 \le \gamma &\iff \left(\Big(P_n(1)K_I T_s\Big)^{-1}\right)^{\mathrm{H}} \Big(P_n(1)K_I T_s\Big)^{-1} \preceq \gamma^2 I_q \\ &\iff X_n^{\mathrm{H}}\left(\Big(P_n(1)K_I\Big)^{-1}\right)^{\mathrm{H}} \Big(P_n(1)K_I\Big)^{-1} X_n \preceq X_n^{\mathrm{H}}\gamma^2 T_s^2 X_n \\ &\iff X_n^{\mathrm{H}}X_n \succeq \gamma^{-2}T_s^{-2} I_q. \end{aligned}$$

We transform other spectral norms similarly:

$$\|S_{nk}\|_2 \le b_{Sk} \iff S_{nk}^{\mathrm{H}}S_{nk} \preceq b_{Sk}^2 I_q \iff Z_{nk}^{\mathrm{H}}S_{nk}^{\mathrm{H}}S_{nk}Z_{nk} \preceq Z_{nk}^{\mathrm{H}}b_{Sk}^2 Z_{nk}$$

$$
\begin{aligned}
&\iff Z_{nk}^{\mathrm{H}} Z_{nk} \succeq b_{Sk}^{-2} I_q, \\
\|T_{nk}\|_2 \le b_{Tk} &\iff S_{nk}^{\mathrm{H}} L_{nk}^{\mathrm{H}} L_{nk} S_{nk} \preceq b_{Tk}^2 I_q \iff Z_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} L_{nk}^{\mathrm{H}} L_{nk} S_{nk} Z_{nk} \preceq Z_{nk}^{\mathrm{H}} b_{Tk}^2 Z_{nk} \\
&\iff Z_{nk}^{\mathrm{H}} Z_{nk} \succeq b_{Tk}^{-2} L_{nk}^{\mathrm{H}} L_{nk}, \\
\|Q_{nk}\|_2 \le b_{Qk} &\iff S_{nk}^{\mathrm{H}} C_k^{\mathrm{H}} C_k S_{nk} \preceq b_{Qk}^2 I_q \iff Z_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} C_k^{\mathrm{H}} C_k S_{nk} Z_{nk} \preceq Z_{nk}^{\mathrm{H}} b_{Qk}^2 Z_{nk} \\
&\iff Z_{nk}^{\mathrm{H}} Z_{nk} \succeq b_{Qk}^{-2} C_k^{\mathrm{H}} C_k, \\
\|W_{nk}\|_2 \le b_{Wk} &\iff P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} S_{nk} P_{nk} \preceq b_{Wk}^2 I_m \\
&\iff \left(P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}}\right)^{\dagger} \left(P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} S_{nk} P_{nk}\right) P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} \preceq \left(P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}}\right)^{\dagger} \left(b_{Wk}^2 I_m\right) P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} \\
&\iff S_{nk} P_{nk} P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} \preceq \left(S_{nk} P_{nk} P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}}\right)^{-1} S_{nk} P_{nk} P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} b_{Wk}^2 \\
&\iff Z_{nk} S_{nk} P_{nk} P_{nk}^{\mathrm{H}} S_{nk}^{\mathrm{H}} Z_{nk}^{\mathrm{H}} \preceq Z_{nk} b_{Wk}^2 Z_{nk}^{\mathrm{H}} \\
&\iff P_{nk} P_{nk}^{\mathrm{H}} \preceq Z_{nk} b_{Wk}^2 Z_{nk}^{\mathrm{H}} \iff Z_{nk} Z_{nk}^{\mathrm{H}} \succeq b_{Wk}^{-2} P_{nk} P_{nk}^{\mathrm{H}}.
\end{aligned}
$$

Then, by exploiting the duality property of SDP (Chapter 5 in [39]), we can reformulate the original optimization problem into its dual form

$$
\begin{cases} \min \gamma \\ \left\| \left(P_n(1) K_I T_s\right)^{-1} \right\|_2 \le \gamma, \end{cases} \iff \begin{cases} \max \mu = \gamma^{-2} \\ X_n^{\mathrm{H}} X_n \succeq \mu T_s^{-2} I_q. \end{cases}
$$

Thus, instead of minimizing the maximum singular value of the matrix $X_n^{-1}$ (34)

$$
\left\| X_n^{-1} \right\|_2 = \overline{\sigma}\left(X_n^{-1}\right),
$$

we maximize the minimum singular value of the matrix $X_n$ since

$$
\underline{\sigma}(X_n) = 1/\overline{\sigma}\left(X_n^{-1}\right).
$$

We can now express the problem of synthesizing the MIMO PID controller for the array of plant models as the system of quadratic matrix inequalities (QMIs):

$$
\begin{cases} \max & \mu, \\ \text{s.t.} & X_n^{\mathrm{H}} X_n \succeq \mu T_s^{-2} I_q, \\ & Z_{nk}^{\mathrm{H}} Z_{nk} \succeq b_{Sk}^{-2} I_q, \\ & Z_{nk}^{\mathrm{H}} Z_{nk} \succeq b_{Tk}^{-2} L_{nk}^{\mathrm{H}} L_{nk}, \\ & Z_{nk}^{\mathrm{H}} Z_{nk} \succeq b_{Qk}^{-2} C_k^{\mathrm{H}} C_k, \\ & Z_{nk} Z_{nk}^{\mathrm{H}} \succeq b_{Wk}^{-2} P_{nk} P_{nk}^{\mathrm{H}}, \\ \text{for each} & n, k. \end{cases} \tag{35}
$$

### *5.2. The Convex-Concave Procedure*

The QMIs in (35) are not linear with respect to the MIMO PID controller parameters. Furthermore, the left-hand side of the inequalities is not convex, thus necessitating the conversion of it into a system of LMIs using the convex-concave procedure (CCP) [40]. The CCP is an optimization method that involves iteratively solving convex sub-problems and concave sub-problems until convergence is reached. It is often applied to problems in control theory and signal processing, where a convex sub-problem is typically solved using a convex optimization technique such as gradient descent or the interior point method, whereas a concave sub-problem is solved using a dual formulation or a linear program. This procedure ensures that the objective function is monotonically increasing with each iteration and converges to a globally optimal solution under certain assumptions.

We have a QMI of the following form:

$$Z^{\mathrm{H}}Z \succeq Y^{\mathrm{H}}Y.$$

We introduce the matrix $\check{Z}$ on the left-hand side without violating the matrix inequality:

$$(Z-\check{Z})^{\mathrm{H}}(Z-\check{Z}) = Z^{\mathrm{H}}Z - Z^{\mathrm{H}}\check{Z} - \check{Z}^{\mathrm{H}}Z + \check{Z}^{\mathrm{H}}\check{Z} \succeq 0,$$

$$Z^{\mathrm{H}}Z \succeq Z^{\mathrm{H}}\check{Z} + \check{Z}^{\mathrm{H}}Z - \check{Z}^{\mathrm{H}}\check{Z},$$

$$Z^{\mathrm{H}}Z \succeq Z^{\mathrm{H}}\check{Z} + \check{Z}^{\mathrm{H}}Z - \check{Z}^{\mathrm{H}}\check{Z} \succeq Y^{\mathrm{H}}Y.$$

To convert the matrix inequality

$$Z^{\mathrm{H}}\check{Z} + \check{Z}^{\mathrm{H}}Z - \check{Z}^{\mathrm{H}}\check{Z} - Y^{\mathrm{H}}Y \succeq 0$$

into the LMI form, we use the Schur complement lemma ([9], p. 28):

$$\begin{bmatrix} A_1 & A_2 \\ A_2^{\mathrm{H}} & A_3 \end{bmatrix} \succeq 0 \iff A_3 \succeq 0, \quad A_1 - A_2 A_3^{\dagger} A_2^{\mathrm{H}} \succeq 0, \quad A_2(I - A_3 A_3^{\dagger}) = 0.$$

By substituting

$$A_1 = Z^{\mathrm{H}}\check{Z} + \check{Z}^{\mathrm{H}}Z - \check{Z}^{\mathrm{H}}\check{Z}, \quad A_2 = Y^{\mathrm{H}}, \quad A_3 = I,$$

we obtain the LMI from the QMI:

$$Z^{\mathrm{H}}Z \succeq Y^{\mathrm{H}}Y \quad \Leftrightarrow \quad \begin{bmatrix} Z^{\mathrm{H}}\check{Z} + \check{Z}^{\mathrm{H}}Z - \check{Z}^{\mathrm{H}}\check{Z} & Y^{\mathrm{H}} \\ Y & I \end{bmatrix} \succeq 0.$$

Now, everything is in place to obtain the final LMI system, the solution of which for each $n$-th plant model and at each $k$-th frequency on each $i$-th CCP iteration will give us the matrices of the MIMO PID controller $\{K_P, K_I, K_D\}_i$, which correspond to the local optimum $\mu_i$:

$$\begin{cases} \max & \mu_i, \\ \text{s.t.} & X_n^{\mathrm{H}}\check{X}_n + \check{X}_n^{\mathrm{H}}X_n - \check{X}_n^{\mathrm{H}}\check{X}_n - \mu T_s^{-2} I_q \succeq 0, \\ & Z_{nk}^{\mathrm{H}}\check{Z}_{nk} + \check{Z}_{nk}^{\mathrm{H}}Z_{nk} - \check{Z}_{nk}^{\mathrm{H}}\check{Z}_{nk} - b_{Sk}^{-2} I_q \succeq 0, \\ & \begin{bmatrix} Z_{nk}^{\mathrm{H}}\check{Z}_{nk} + \check{Z}_{nk}^{\mathrm{H}}Z_{nk} - \check{Z}_{nk}^{\mathrm{H}}\check{Z}_{nk} & b_{Tk}^{-1}L_{nk}^{\mathrm{H}} \\ b_{Tk}^{-1}L_{nk} & I_q \end{bmatrix} \succeq 0, \\ & \begin{bmatrix} Z_{nk}^{\mathrm{H}}\check{Z}_{nk} + \check{Z}_{nk}^{\mathrm{H}}Z_{nk} - \check{Z}_{nk}^{\mathrm{H}}\check{Z}_{nk} & b_{Qk}^{-1}C_k^{\mathrm{H}} \\ b_{Qk}^{-1}C_k & I_m \end{bmatrix} \succeq 0, \\ & \begin{bmatrix} Z_{nk}\check{Z}_{nk}^{\mathrm{H}} + \check{Z}_{nk}Z_{nk}^{\mathrm{H}} - \check{Z}_{nk}\check{Z}_{nk}^{\mathrm{H}} & b_{Wk}^{-1}P_{nk} \\ b_{Wk}^{-1}P_{nk}^{\mathrm{H}} & I_m \end{bmatrix} \succeq 0, \\ \text{for each} & n, k, \end{cases} \tag{36}$$

where the matrix $\check{Z}_{nk}$ corresponds to (33), as depicted in the following equation:

$$\check{Z}_{nk} = I + P_{nk}\check{C}_k = I_q + P_{nk}\left(\check{K}_P + \check{K}_I \frac{\exp(j\omega_k T_s)T_s}{\exp(j\omega_k T_s) - 1} + \check{K}_D \frac{\exp(j\omega_k T_s) - 1}{\exp(j\omega_k T_s)T_s}\right).$$

Similarly, the matrix $\check{X}_{nk}$ corresponds to (34), as shown below:

$$\check{X}_{nk} = P_n(1)\check{K}_I.$$

Here, $\{\check{K}_P, \check{K}_I, \check{K}_D\} = \{K_P, K_I, K_D\}_{i-1}$ are the MIMO PID controller matrices obtained at the previous iteration of the CCP. Thus, we guarantee that the value of the local optimum $\mu_i$ does not decrease with each new iteration of the CCP. With each new iteration, we become closer to the global optimum $\mu_i \longrightarrow \mu$. The MIMO PID controller matrices $\{K_P, K_I, K_D\}$, which must be found during the solution of the LMIs (36), are linearly included in the LMIs (36) via Equations (32)–(34).

The algorithm for synthesizing a MIMO PID controller using the CCP is straightforward:

- Determine the parameters;
- Set $\check{Z}_{nk} = Z_{nk}^{\text{init}}$, $\check{X}_k = X_k^{\text{init}}$;
- Repeat solving (36) until $\left|\mu_i/\mu_{i-1}\right| - 1 > \tau$.

Here, $\mu_i$ is the local optimum of the i-th CCP iteration, and $\mu_{i-1}$ is the local optimum of the previous CCP iteration. This algorithm stops the CCP when the local optimum $\mu_i$ is considered to be close to the optimum $\mu$. A typical value for the tolerance $\tau$ is 1%.

It is important to note that in this context, the term "optimum" refers to the optimal solution of (36), not the globally optimal solution of the system (35). The solution of (36) generally depends on the choice of the initializing MIMO PID controller.

### *5.3. Initialization of the Controller*

The matrices $Z_{nk}^{\text{init}}$ and $X_k^{\text{init}}$ correspond to (33) and (34) with an initializing MIMO PID controller $\{K_P, K_I, K_D\}^{\text{init}}$. Any MIMO PID controller that ensures the stability of the closed-loop system can be used as an initializing controller.

For a stable plant model, an initializing MIMO PID controller from [7] can be used:

$$K_P^{\text{init}} = 0, \quad K_I^{\text{init}} = \epsilon P_1(1)^{\dagger}, \quad K_D^{\text{init}} = 0,$$

where $\epsilon$ is a small scalar and $P_1(1)$ corresponds to the static gain of the first model in the array of discrete plant models.

## 6. The MATLAB Toolbox

To implement the controller synthesis algorithm described in this paper, we have developed a MATLAB toolbox. The toolbox includes tools for visualizing the synthesis results, such as the figure shown in Figure 7. The toolbox also provides the synthesis of continuous controllers in the problem statement from [1].

We offer our toolbox for free use to the community under the MIT license. If you use our toolbox in your research, please consider citing this paper. Feedback, suggestions, and bug reports are welcome. The requirements for using our toolbox are as follows:

- MATLAB version above R2020a with:
  - Control System Toolbox;
  - Robust Control Toolbox;
- A convex optimization framework, such as CVX [41] or YALMIP [42]:
  - CVX is more conservative and has a slightly larger number of settings;
  - YALMIP is faster;
- An SDP solver, such as SDPT3 4.0 [43] or MOSEK [44]:
  - SDPT3 is more stable with numerical problems;
  - MOSEK is faster.

The toolbox is available at the following link:

https://gitlab.com/konkov/mimo-pid-lmi-synthesis-toolbox

or it can be cloned directly using the following command:

```
git clone git@gitlab.com:konkov/mimo-pid-lmi-synthesis-toolbox.git
```

*The Potential Numerical Problems*

The numerical stability problem in algorithms is quite extensive and well known. It is associated with the fact that in any computer calculations, floating-point numbers are used, which have a constant relative error but varying absolute error. This problem is thoroughly discussed in [45].

For example, when checking the controllability of a control system model using the Kalman criterion and employing double-precision numbers, the algorithm may indicate that a control system model, which is real-controllable, appears to be uncontrollable. This discrepancy is linked to the precision of determining the matrix rank when working with floating-point numbers. When using double-precision numbers (64 bits, with a 52-bit mantissa), two numbers separated by a distance less than $2^{-52}$ are considered identical. Moreover, such a plant model will be controllable, for instance, if numbers with quadruple-precision floating-point representation are used (128 bits, with a 112-bit mantissa).

Our algorithm works with the frequency responses of transfer functions, so if the condition numbers of these responses are large, numerical problems are inevitable. We recommend initiating the controller synthesis by checking the controllability of the plant model using the Kalman criterion. If the plant model is real-controllable but the algorithm indicates an incomplete rank of the controllability matrix, scaling of the model should be conducted.

## 7. Numerical Examples

We present two numerical examples in which we synthesize a discrete MIMO PID controller for an LTV plant model. In both examples, we apply loop shaping to achieve a trade-off between the performance and robust stability margin of the closed-loop system. We do not delve into a detailed analysis of the presented examples but briefly emphasize the key points. All the examples presented here, along with several additional ones, are available in our toolbox. They can be easily reproduced and thoroughly explored.

### *7.1. Example 1*

We begin with an example of the Wood–Berry binary distillation column described in [46]. This example is widely used in works dedicated to the synthesis of MIMO PID controllers. To fully demonstrate our methodology, we artificially introduce a parameter $\delta_n$ into the plant model. This parameter can be interpreted as uncertainties in the measured parameters or sensor faults (in the case of delay). This parametrization corresponds to the form of an uncertain model (5). Thus, some model parameters will change by a factor of two, which is significant. This way, we obtain an array of two continuous transfer functions, where $s$ denotes the Laplace transform variable:

$$P_n(s) = \begin{bmatrix} \dfrac{12.8\exp(-\delta_n s)}{16.7s+1} & \dfrac{-18.9\exp(-3s)}{\delta_n 21s+1} \\ \dfrac{6.6\exp(-7s)}{\delta_n 10.9s+1} & \dfrac{-\delta_n 19.4\exp(-3s)}{14.4s+1} \end{bmatrix}, \quad \delta_n \in \{1,2\}. \tag{37}$$

Following this, we discretize each model in the array using the ZOH method with a sample time of $T_s = 0.1$ s, resulting in an array of discrete transfer functions $P_n(z)$ in the form of (6). We intentionally chose such a large sampling time. The controllability matrix of all plant models in the array is of full rank.

The results of the optimization process are shown in Figures 9–11. It required six iterations to reach the optimal solution. The computation, performed using YALMIP and SDPT3 4.0 in the native MATLAB R2023b on an Apple Silicon M2 MAX SoC (System-on-Chip), took approximately 40 s.

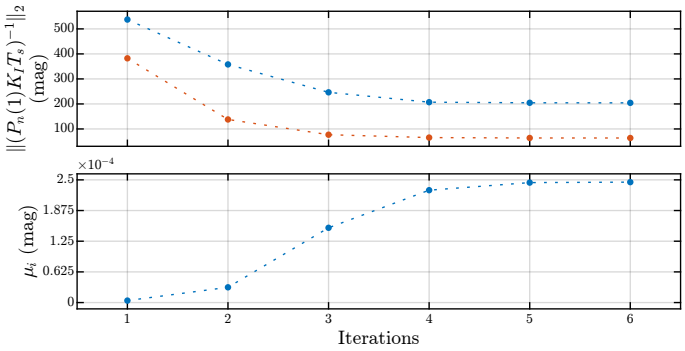



**Figure 9.** The results of the CCP for Example 1. The convergence of the local optimum $\mu_i$ to the optimum $\mu$ is demonstrated, along with the variation in the spectral norm of the SLFS matrix. The blue dashed line corresponds to $P_1(z)$, whereas the red line corresponds to $P_2(z)$.

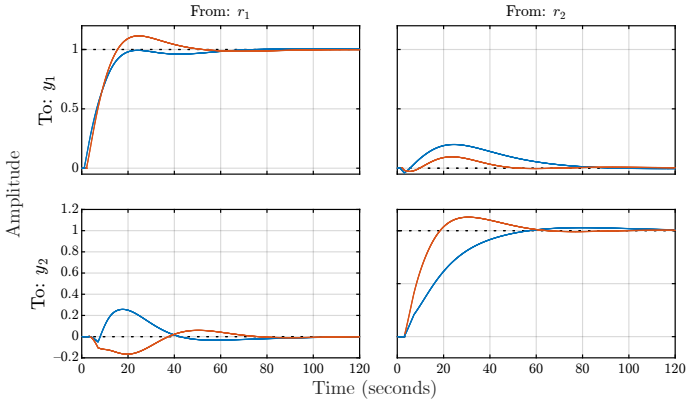



**Figure 10.** Example 1: Step responses of the synthesized digital MIMO control system, structured according to the diagram in Figure 5. The blue line corresponds to $P_1(z)$, whereas the red line corresponds to $P_2(z)$.

Solving the system of LMIs (36) at $N_k = 400$ (31) frequencies through the CCP, we obtain the discrete MIMO PID controller in the form of (7) with the following parameters:

$$K_P = \begin{bmatrix} 0.1246 & -0.0298 \\ 0.0017 & -0.0340 \end{bmatrix}, \; K_I = \begin{bmatrix} 0.0182 & -0.0008 \\ 0.0047 & -0.0038 \end{bmatrix}, \; K_D = \begin{bmatrix} -0.0016 & 0.0059 \\ -0.0018 & -0.0019 \end{bmatrix}.$$

We place particular emphasis on Figure 11, which illustrates the singular values of the transfer function array (Section 3.3) of the synthesized digital control system and their shape functions (Section 4.2). Here, the following performance criteria are defined (Section 3.4): $S_{\max} = T_{\max} = 1.3$, $Q_{\max} = 0.25$. The shape functions $b_S(\omega)$ and $b_Q(\omega)$ are constants. The shape function $b_T(\omega)$ is a constant only for $\omega \in \Omega_L \cup \Omega_C$ (14); at high frequencies, it serves as the robustness bound (20). We do not impose any restrictions on $W_n(z)$, but one can see in Figure 11 the effect of the input external perturbation on the control performance. The key point here is that all these constraints are simultaneously satisfied for all transfer functions in the array, even considering that certain model parameters (37) change by a factor of two. This is clearly visible in Figure 11.

Regarding the trade-off between performance and robustness (Section 4.3), we have achieved a very large robustness margin while maintaining the reasonable performance of the closed-loop control system. The robustness area (21) is $A_R = 4.78$, and the performance area (27) is $A_P = 0.044$ (with $\omega_0 = 10^{-3}$ rad/s). The control loops are also well-decoupled (Section 4.6), as evident from both Figures 10 and 11.

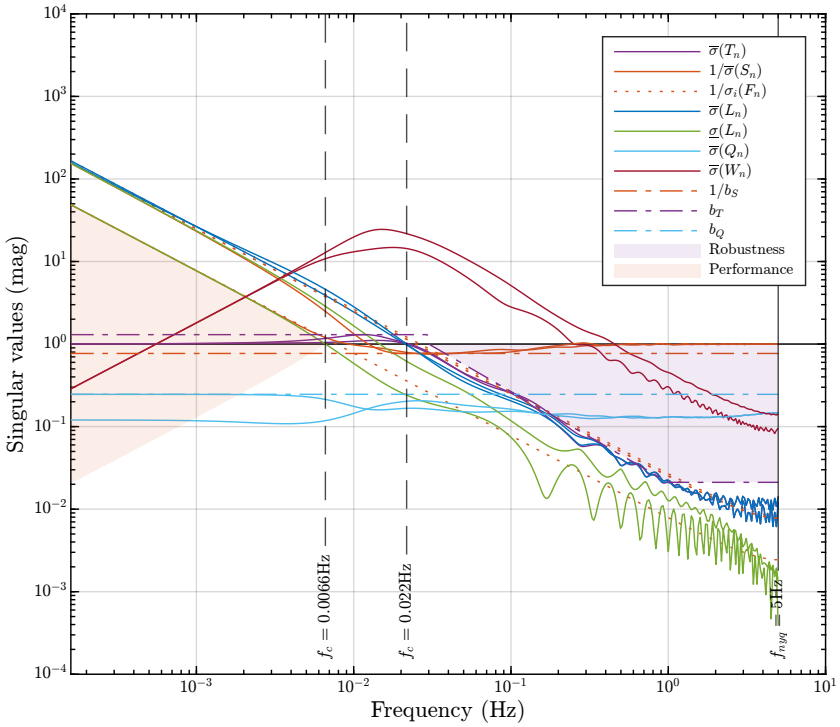



**Figure 11.** Example 1: Singular values of the transfer function array $L_n(z)$, $S_n(z)$, $Q_n(z)$, and $T_n(z)$ of the synthesized digital MIMO control system. The performance area and robustness area are shaded. The shape functions are indicated by dashed lines.

### *7.2. Example 2*

With this example, we aim to illustrate the advantages of directly synthesizing a discrete controller in a digital control system, as discussed in Section 1.4 (approach 2). The objective is to synthesize a digital control system in which the steady state is achieved within a relatively small number of samples, for example, within 10. Furthermore, we showcase the synthesis of a MIMO PID controller on a non-square plant model. This example was sourced from the MATLAB guide (https://www.mathworks.com/help/robust/ug/specify-lmi-system-at-the-command-line.html URL accessed on 6 March 2024) and was slightly modified by us. The matrix $C_n$ is parameterized by the variables $a_n$ and $b_n$, which can be interpreted both as uncertainties (5) and as values of $\theta_n$ in the LPV model (4) in the case of linear interpolation between time points. Below is the array of matrices for the continuous-time plant model. The plant model has $m = 4$ inputs, $q = 3$ outputs, and six states:

$$A = \begin{bmatrix} -0.0872 & 0.0520 & 0.0747 & 0.1078 & -0.0969 & 0.0101 \\ -0.0558 & -0.1084 & 0.1891 & 0.0252 & 0.1064 & -0.0035 \\ -0.0262 & -0.1754 & -0.1545 & -0.0214 & 0.0092 & -0.2419 \\ 0.0609 & -0.1074 & 0.0131 & -0.2558 & 0.2321 & 0.0239 \\ -0.0717 & 0.0358 & -0.1419 & 0.1704 & -0.2653 & -0.1428 \\ -0.1294 & -0.0675 & 0.1698 & 0.1676 & -0.0365 & -0.1773 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 89.98 & -21.3 & 98.35 \\ 0 & -30.01 & 0 & -29.77 \\ -103.22 & 0 & -104.31 & 114.37 \\ 0 & -34.51 & -27.01 & -53.16 \\ -41.89 & 101.28 & -43.81 & 0 \\ 0 & 0 & -40.87 & 0 \end{bmatrix}, \quad a_n = \{1, 1.05\},\ b_n = \{1, 1.1\},$$

$$C_n = \begin{bmatrix} a_n \cdot 0.9707 & a_n \cdot 0.951 & 0.7059 & b_n \cdot 1.458 & -1.2371 & 0.3174 \\ 0 & 0 & 1.4158 & 0.0475 & -2.1935 & 0.4136 \\ a_n \cdot -0.4383 & a_n \cdot 0.6489 & -1.6045 & b_n \cdot 1.7463 & -0.3334 & -0.5771 \end{bmatrix}.$$

Similar to Example 1 (Section 7.1), we discretize each model in the array using the ZOH method, but now we utilize a sampling time of $T_s = 0.001$ s. This results in an array of discrete transfer functions $P_n(z)$ in the form of (6). The controllability matrix of all plant models in the array is of full rank. The loops of the plant model are strongly coupled, as indicated by the RGA value (28) at zero frequency ($z = 1$) for the first model in the array:

$$\mathrm{RGA}\Big(P_1(1)\Big) = \begin{bmatrix} 0.0120 & 0.2825 & 1.1801 & -0.4746 \\ -0.5586 & 0.4709 & 2.4159 & -1.3281 \\ 0.6219 & 0.1935 & -2.5969 & 2.7815 \end{bmatrix}.$$

The results are shown in Figures 12 and 13.

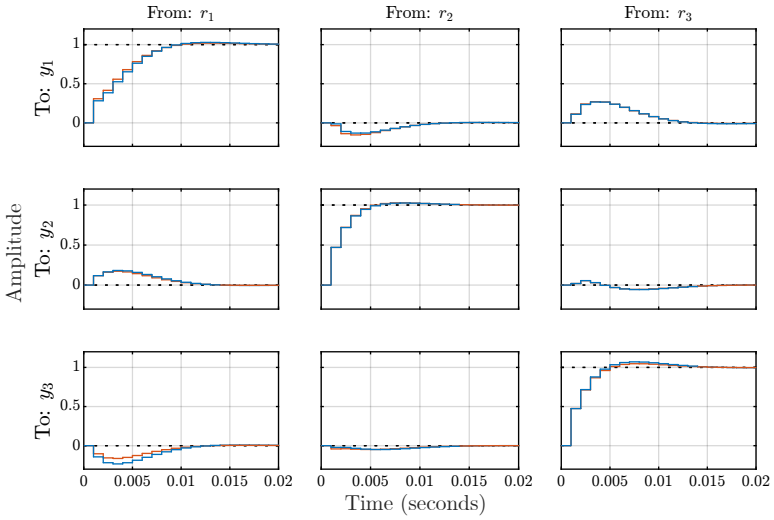



**Figure 12.** Example 2: Step responses of the synthesized digital MIMO control system, organized according to the diagram in Figure 5. The blue line corresponds to $P_1(z)$, whereas the red line corresponds to $P_2(z)$.

To solve the system of LMIs in (36) with $N_k = 300$, it required 11 iterations of the CCP. The optimization process on the same device and with the same software as in Example 1 (Section 7.1) took approximately 150 s. As a result, we obtained the discrete MIMO PID controller in the form of (7) with the following parameters:

$$K_P = \begin{bmatrix} -5.0392 & -5.5404 & 6.6605 \\ -0.7234 & -1.9489 & -0.3424 \\ -1.7436 & 5.4253 & -2.4762 \\ -2.6610 & 0.6851 & 1.1269 \end{bmatrix}, \; K_I = \begin{bmatrix} -0.9715 & 0.2322 & 0.6826 \\ -0.3079 & -0.1449 & -0.1045 \\ -0.3156 & -0.2505 & -0.4292 \\ -0.1806 & 0.0134 & 0.0423 \end{bmatrix},$$

$$K_D = \begin{bmatrix} 0.0042 & 0.0030 & -0.0019 \\ 0.0004 & 0.0003 & -0.0000 \\ -0.0019 & -0.0015 & 0.0012 \\ 0.0010 & 0.0008 & -0.0005 \end{bmatrix}.$$

The step responses of the synthesized digital MIMO control system are shown in Figure 12. We have achieved the set goal, with approximately 10 samples required for the system to reach a steady state.

The singular values of the transfer function array (Section 3.3) of the synthesized digital control system and their shape functions (Section 4.2) are shown in Figure 13. For this example, the following performance criteria are defined (Section 3.4): $S_{\max} = T_{\max} = 1.4$,

$Q_{\max} = 10$. The shape functions $b_S(\omega)$ and $b_Q(\omega)$ are constants. The shape function $b_T(\omega)$ is a constant only for $\omega \in \Omega_L \cup \Omega_C$ (14); at high frequencies, it serves as the robustness bound (20). We again do not impose any constraints on $W_n(z)$. In Figure 13, one can observe the minimal impact of the external input perturbations on the control performance. All specified constraints are simultaneously satisfied for all transfer functions in the array. The control loops are also well-decoupled (Section 4.6), as evident from both Figures 12 and 13.

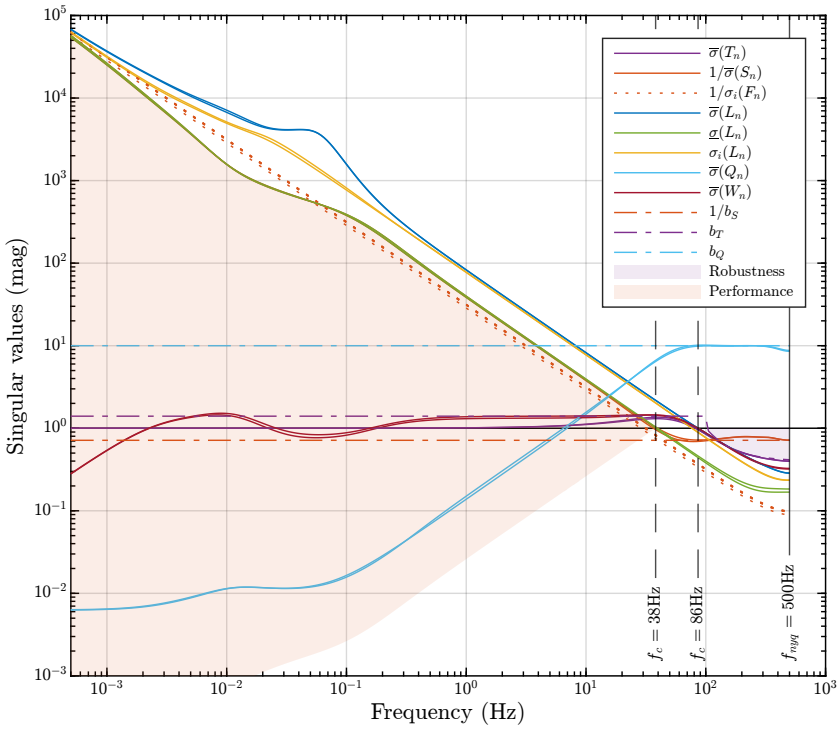



**Figure 13.** Example 2: Singular values of the transfer function array $L_n(z)$, $S_n(z)$, $Q_n(z)$, and $T_n(z)$ of the synthesized digital MIMO control system. The performance area and robustness area are shaded. The shape functions are indicated by dashed lines.

Here, we have a more balanced trade-off (Section 4.3) between performance and robustness compared to Example 1 (Section 7.1). The robustness area (21) is $A_R = 202.4247$, and the performance area (27) is $A_P = 821.33$ (with $\omega_0 = 10^{-4}$ rad/s). Please note that the axes in Figure 13 are logarithmic.

## 8. Discussion

We only considered the singular values of transfer functions and did not discuss their phases. The concept of phases in MIMO transfer functions is more complex, and the theory of phases for SISO transfer functions cannot be directly applied to the MIMO case. However, in recent times, a theory of phases for MIMO transfer functions has begun to be developed. It is possible that in future works, we will enhance our synthesis algorithm by incorporating considerations of phases in MIMO transfer functions.

Indeed, PID controllers of various types enable the solution of the vast majority of practical problems. However, our proposed methodology enables tuning any controller with a fixed structure. It is only necessary to derive the SLFS function, similar to the approach outlined in Appendix B.

In fact, our methodology can also be applied to synthesizing a controller for a nonlinear plant model or even without having a plant model at all. Ultimately, when synthesizing a controller from a plant model, only a set of singular values (gains) at different frequencies is required. In the case of a linear plant model, these singular values are obtained using the transfer function when computing the frequency response. A nonlinear plant model

cannot be represented as a transfer function, but the set of gains can be directly computed by applying Fourier transformation to the output signals. This way, the controller synthesis can be seamlessly combined with the identification of the plant model.

## 9. Conclusions

We extended the synthesis method proposed by Boyd et al. in [1] to discrete time, augmenting it with the additional statement of loop shaping. We briefly touched upon the challenges arising during the discretization of a continuous controller, especially in the MIMO case. Direct synthesis in discrete time, as opposed to discretizing a continuous controller, ensures the preservation of performance and robust stability margins in a digital control system.

We synthesized an LTI controller (7) for an LTV plant model (3) represented as a transfer function array with index $n$ (6). This approximation of the LTV plant model can be obtained using an LPV plant model (4), an uncertain plant model (5), or by simultaneously considering both. If the system of inequalities (36) has a solution on each iteration, all specified constraints will be satisfied for the entire transfer function array of the closed-loop system with an LTV plant model (3).

Regarding the controller derived from our algorithm, it is important to emphasize the following. We achieve an optimal controller, signifying that the closed-loop system attains maximum performance under constraints imposed on the transfer functions of the closed-loop system. This maximization of performance is realized through the iterative minimization of the static and low-frequency sensitivity (SLFS) of the closed-loop system, facilitated by the minimization of the spectral norm of the SLFS matrix (11). Ultimately, this translates into the maximization of the performance area $A_P$ (27) with a "floating" boundary represented by the performance bound $b_P(\omega)$ (26). By employing the convex-concave procedure (CCP), we obtain a locally optimal solution to the system (36) at each iteration, progressively converging toward the global optimum. Furthermore, the flexibility of the optimization algorithm enables termination at any point by adjusting the CCP stopping condition, contingent on the desired values of the performance area and performance bound.

The minimization of the spectral norm of the SLFS matrix serves dual objectives: ensuring the decoupling of control loops at low frequencies and maximizing control performance. Additionally, it results in a closed-loop system with crossover frequencies as proximate as possible, leading to nearly identical time responses for its control loops. The presence of a "floating" performance bound is considered an advantage of our methodology compared to classical loop shaping, where both bounds are strictly defined, impeding the synthesis of an optimal controller. The judicious selection of the robustness bound $b_R(\omega)$ (20) holds critical importance in digital systems, affording a sufficiently extensive robustness area and obviating the need for anti-aliasing filters in the feedback loop. By varying the robustness bound, a trade-off is struck between performance and the robust stability margin of the closed-loop system. A larger robustness area corresponds to a smaller performance area, and vice versa. We propose simple estimates for this trade-off, manifested in the form of performance and robustness bounds ($b_P(\omega)$, $b_R(\omega)$) (20), (26) and performance and robustness areas ($A_P$ , $A_R$) (27), (20). To fortify robust stability, our tripartite strategy involves synthesizing over an array of models, integrating $\mathcal{H}_\infty$-norm constraints guided by the small-gain theorem, and employing loop shaping to concurrently specify a robustness bound for all models in the plant model array.

The methodology proposed in this paper was developed by us as part of the development of digital plasma magnetic control systems in tokamaks. We have used some elements from this methodology in our previous works [47–50]. Of particular note is that in [50], the controller synthesis was conducted simultaneously for an array of 20 LTI plant models, and the plasma shape controller had five inputs and six outputs. The methodology presented in this paper will continue to be used by us in the future for plasma control.

We also hope that this methodology can be applied in many other fields. Given the complexity of our method and the potentially time-consuming nature of programming the algorithm, we offer a ready-to-use MATLAB toolbox for convenience. Moreover, we have strived to thoroughly explicate all nuances of the methodology, intending to simplify its comprehension for readers to the utmost.

*Limitations of This Study*

This paper is theoretical in nature and proposes a new methodology for synthesizing discrete MIMO PID controllers. The numerical examples provided serve solely to illustrate some of the capabilities of the proposed method. Our study was not aimed at comparing MIMO PID to other controller types, nor was it aimed at comparing our synthesis method to existing ones. We provide a ready-to-use toolbox to allow researchers to independently compare their own synthesis methods with our method.

A separate study dedicated to comparing different controller types may be conducted in the future. Due to the complex nature of discrete controller synthesis methods in MIMO control systems with well-coupled control loops and the potential difficulty of such investigations, it is desirable to involve experts in each type of controller under study. However, even with such involvement, such research may only apply to certain classes of plants. There are no universally effective controllers, nor are there universal methods for synthesizing them. A controller is simply a tool used to solve a specific control problem for a real plant. Comparing different types of controllers for a single class of plants is not sufficient and may not reveal the full potential of each design. Ultimately, the aim of any controller design method is to create a reliable control system that satisfies specified performance requirements while also possessing the necessary robust stability margins. In addition, the physical implementation aspects of digital controllers must be considered. Many complex control algorithms may experience difficulties operating in real time on existing hardware. It is also important to take into account the rationality of utilizing each type of controller, as it may not be rational to use a more complex algorithm when a simpler one could accomplish the task with only a slight decrease in performance.

Potential future research could be systematic, considering all these factors. Perhaps this could be a book, where each section is written by an expert on a specific type of controller. Additionally, objective comparison criteria for different types of controllers should be formulated for different classes of plants. We are ready to contribute our methodology for synthesizing MIMO PID controllers to such a book.

**Author Contributions:** Conceptualization, A.E.K.; methodology, A.E.K.; software, A.E.K.; validation, A.E.K.; formal analysis, A.E.K.; investigation, A.E.K.; resources, A.E.K.; data curation, A.E.K.; writing—original draft preparation, A.E.K.; writing—review and editing, A.E.K. and Y.V.M.; visualization, A.E.K.; supervision, Y.V.M.; project administration, A.E.K. and Y.V.M.; funding acquisition, Y.V.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Russian Science Foundation (RSF), project No. 21-79-20180, and the Russian Foundation for Basic Research (RFBR), project No. 19-31-90136.

**Data Availability Statement:** Data are contained within the article.

**Acknowledgments:** The authors extend their gratitude to the developers of the convex optimization frameworks CVX and YALMIP, and to the developers of the SDP solvers SDPT3 4.0 and MOSEK.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| ADC | Analog-to-Digital Converter |
| DAC | Digital-to-Analog Converter |
| CCP | Convex-Concave Procedure |
| FOH | First-Order Hold |

| | |
|---|---|
| LMI | Linear Matrix Inequality |
| LTI | Linear Time Invariant |
| LTV | Linear Time Varying |
| LPV | Linear Parameter Varying |
| MIMO | Multiple-Input Multiple-Output |
| PID | Proportional–Integral–Derivative |
| RTTM | Real-Time Target Machine |
| SISO | Single-Input Single-Output |
| SLFS | Static and Low-Frequency Sensitivity |
| SDP | Semi-Definite Programming |
| QMI | Quadratic Matrix Inequality |
| ZOH | Zero-Order Hold |

## Appendix A. Proof of the Identity for the Complementary Sensitivity Function

The problem is to prove that

$$T(z) = L(z)\Big(I_q + L(z)\Big)^{-1} = \Big(I_q + L(z)\Big)^{-1}L(z), \quad L(z) = P(z)C(z),$$

where $I_q$ is the identity matrix of size $q$, $P(z) \in \mathbb{C}^{q\times m}$, and $C(z) \in \mathbb{C}^{m\times q}$. This identity, while not intuitively apparent in the MIMO case, is seldom mentioned in the literature.

For brevity, function arguments are omitted. We start with the expression:

$$PC = (I_q + PC)^{-1}(I_q + PC)PC = (I_q + PC)^{-1}(PC + PCPC) = (I_q + PC)^{-1}PC(I_q + PC).$$

Performing right multiplication by $(I_q + PC)^{-1}$, we obtain:

$$PC(I_q + PC)^{-1} = (I_q + PC)^{-1}PC(I_q + PC)(I_q + PC)^{-1} = (I_q + PC)^{-1}PC.$$

This completes the proof.

## Appendix B. Explanation of the Static and Low-Frequency Sensitivity Function

We have the sensitivity function

$$S(z) = \Big(I + P(z)C(z)\Big)^{-1},\ P(z) \in \mathbb{C}^{q\times m},\ C(z) \in \mathbb{C}^{m\times q},\ S(z) \in \mathbb{C}^{q\times q},\ z \in \mathbb{C}^1, \tag{A1}$$

where $P(z)$ is the discrete MIMO transfer function of the plant model with $m$ inputs and $q$ outputs and $C(z)$ is the discrete transfer function of the MIMO PID controller

$$C(z) = K_P + K_I\frac{zT_s}{z-1} + K_D\frac{z-1}{zT_s},\ \{K_P, K_I, K_D\} \in \mathbb{R}^{m\times q}. \tag{A2}$$

We need to obtain the static and low-frequency sensitivity function, for which it is necessary to expand the sensitivity function into a series at $z = 1$, corresponding to zero frequency $\omega = 0$, since $z = \exp(j\omega T_s)$.

We utilize the expansion of the complex matrix function $S(z)$ into a Taylor series [51] (also see Chapter 5 in [52]) at the point $\zeta_0$. Since the argument $z$ is a scalar, we can approximate $S(z)$ as follows:

$$S(z) \approx \sum_{n=0}^{k}\frac{S^{(n)}(\zeta_0)}{n!}(z-\zeta_0)^n, \quad z \in \mathbb{C}^1. \tag{A3}$$

Some properties of differentials of complex matrix functions are summarized in Table 3.1 in [52]:

$$dZ^{-1} = -Z^{-1}(dZ)Z^{-1}, \tag{A4}$$

$$dZ = d\mathrm{Re}\{Z\} + jd\mathrm{Im}\{Z\}, \tag{A5}$$

$$d(Z_1Z_2) = (dZ_1)Z_2 + Z_1(dZ_2), \tag{A6}$$

$$d(Z_3 + Z_4) = dZ_3 + dZ_4. \tag{A7}$$

The Cauchy–Riemann criteria: To have a complex-valued function $f(z)$ holomorphic in an open connected subset of the complex plane, it is necessary and sufficient that it satisfies the Cauchy–Riemann conditions:

$$w = f(z) = u + iv, \quad z = x + iy, \quad \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}. \tag{A8}$$

Substituting (A2) into (A1), we obtain

$$\begin{aligned} S(z) &= \left( I + P(z)\left( K_P + K_I \frac{zT_s}{z-1} + K_D \frac{z-1}{zT_s} \right) \right)^{-1} \\ &= T_s z(z-1) \left( z(z-1)T_s I + P(z)\left( K_P z(z-1)T_s + K_I z^2 T_s^2 + K_D (z-1)^2 \right) \right)^{-1}. \end{aligned} \tag{A9}$$

We introduce the following notations:

$$\begin{aligned} &y(z) = T_s z(z-1), \quad y(z) \in \mathbb{C}^1, \\ &\tilde{C}(z) = T_s z(z-1)K_P + T_s^2 z^2 K_I + (z-1)^2 K_D, \quad C(z) = \frac{\tilde{C}(z)}{y(z)}, \\ &\tilde{S}(z) = \left( T_s z(z-1) I + P(z)\tilde{C}(z) \right)^{-1}, \quad S(z) = y(z)\tilde{S}(z). \end{aligned} \tag{A10}$$

Let us calculate the values of $y(z)$, $\tilde{C}(z)$, $S(z)$, and $\tilde{S}(z)$ at the point $z = 1$:

$$y(1) = 0, \quad \tilde{C}(1) = T_s^2 K_I, \quad S(1) = 0, \quad \tilde{S}(1) = \left( P(1) T_s^2 K_I \right)^{-1}. \tag{A11}$$

Using (A4), we can differentiate $S(z)$ as follows:

$$\frac{dS(z)}{dz} = -S(z)\left( \frac{dI}{dz} + \frac{d\big(P(z)C(z)\big)}{dz} \right) S(z) = -S(z)\left( \frac{d\big(P(z)C(z)\big)}{dz} \right) S(z). \tag{A12}$$

We can now differentiate $C(z)$ (A2):

$$\begin{aligned} \left( \frac{z}{z-1} \right)' &= \frac{z'(z-1) - z(z-1)'}{(z-1)^2} = \frac{-1}{(z-1)^2}, \\ \left( \frac{z-1}{z} \right)' &= \frac{(z-1)'z - (z-1)z'}{z^2} = \frac{1}{z^2}. \end{aligned}$$

Note that criteria (A8) is satisfied in this case. Using the properties (A5)–(A7), along with the fact that $K_P$, $K_I$, and $K_D$ are real matrices, we obtain:

$$\frac{dC(z)}{dz} = K_I T_s \frac{-1}{(z-1)^2} + \frac{K_D}{T_s} \frac{1}{z^2} = \frac{K_D T_s (z-1)^2 - K_I T_s^3 z^2}{y^2(z)}. \tag{A13}$$

By substituting (A10) and (A13) into (A12), we obtain:

$$\begin{aligned}\frac{dS(z)}{dz} &= -y(z)\tilde{S}(z)\left(\frac{dP(z)}{dz}\frac{\tilde{C}(z)}{y(z)} + P(z)\frac{K_D T_s(z-1)^2 - K_I T_s^3 z^2}{y(z)^2}\right) y(z)\tilde{S}(z)\\ &= -y^2(z)\tilde{S}(z)\left(\frac{dP(z)}{dz}\frac{\tilde{C}(z)y(z)}{y(z)^2} + P(z)\frac{K_D T_s(z-1)^2 - K_I T_s^3 z^2}{y(z)^2}\right)\tilde{S}(z)\\ &= -\tilde{S}(z)\left(\frac{dP(z)}{dz}\tilde{C}(z)y(z) + P(z)K_D T_s(z-1)^2 - P(z)K_I T_s^3 z^2\right)\tilde{S}(z).\end{aligned}$$

Using (A11), we can obtain the derivative of the function $S(z)$ at the point $z = 1$:

$$\begin{aligned}\left.\frac{dS(z)}{dz}\right|_{z=1} &= -\tilde{S}(1)\left(\left.\frac{dP(z)}{dz}\right|_{z=1}\tilde{C}(1)y(1) + P(1)K_D T_s(1-1)^2 - P(1)K_I T_s^3 1^2\right)\tilde{S}(1)\\ &= -\Big(P(1)T_s^2 K_I\Big)^{-1}\Big(-P(1)K_I T_s^3\Big)\Big(P(1)T_s^2 K_I\Big)^{-1} = \Big(P(1)K_I T_s\Big)^{-1}.\end{aligned}$$

Now, we can substitute the derived results into (A3) when $k = 1$:

$$S(z) \approx S(1) + \left.\frac{dS}{dz}\right|_{z=1}(z-1), \text{ for small } \omega.$$

We use $k = 1$ because the first term in the series $S(1) = 0$ (A11); for $k = 2$ and higher, we obtain a nonlinear static and low-frequency sensitivity function with respect to the matrices of the controller $\{K_P, K_I, K_D\}$, which makes it impossible to reduce the controller synthesis problem to LMIs form.

As a result, the static and low-frequency sensitivity function of the closed-loop system with the MIMO PID controller has the following form:

$$F(z) = \Big(P(1)K_I T_s\Big)^{-1}(z-1), \quad F(z) \approx S(z) \text{ for small } \omega,\ z = \exp(j\omega T_s).$$

## References


1. Boyd, S.; Hast, M.; Åström, K.J. MIMO PID tuning via iterated LMI restriction. *Int. J. Robust Nonlinear Control* **2016**, *26*, 1718–1731. [CrossRef]
2. Åström, K.; Hägglund, T. The future of PID control. *Control Eng. Pract.* **2001**, *9*, 1163–1175. [CrossRef]
3. Åström, K.J.; Hägglund, T. *Advanced PID Control*; ISA—The Instrumentation, Systems, and Automation Society: Research Triangle Park, NC, USA, 2006.
4. Soltesz, K.; Cervin, A. When is PID a good choice? *IFAC-PapersOnLine* **2018**, *51*, 250–255. [CrossRef]
5. Mohammadpour, J.; Grigoriadis, K.; Franchek, M.; Wang, Y.Y.; Haskara, I. LPV decoupling for control of multivariable systems. *Int. J. Control* **2011**, *84*, 1350–1361. [CrossRef]
6. Baár, T.; Bauer, P.; Luspay, T. Parameter Varying Mode Decoupling for LPV systems. *IFAC-PapersOnLine* **2020**, *53*, 1219–1224. [CrossRef]
7. Davison, E. Multivariable tuning regulators: The feedforward and robust control of a general servomechanism problem. *IEEE Trans. Autom. Control* **1976**, *21*, 35–47. [CrossRef]
8. Penttinen, J.; Koivo, H. Multivariable tuning regulators for unknown systems. *Automatica* **1980**, *16*, 393–398. [CrossRef]
9. Boyd, S.P. *Linear Matrix Inequalities in System and Control Theory*; Number vol. 15 in SIAM Studies in Applied Mathematics; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1994.
10. Zheng, F.; Wang, Q.G.; Lee, T.H. On the design of multivariable PID controllers via LMI approach. *Automatica* **2002**, *38*, 517–526. [CrossRef]
11. Feng, Z.Y.; Guo, H.; She, J.; Xu, L. Weighted sensitivity design of multivariable PID controllers via a new iterative LMI approach. *J. Process Control* **2022**, *110*, 24–34. [CrossRef]
12. Guo, H.; Feng, Z.Y.; She, J. Discrete-time multivariable PID controller design with application to an overhead crane. *Int. J. Syst. Sci.* **2020**, *51*, 2733–2745. [CrossRef]
13. Solgi, Y.; Fatehi, A.; Nikoofard, A.; Shariati, A. Design of optimal PID controller for multivariable time-varying delay discrete-time systems using non-monotonic Lyapunov-Krasovskii approach. *J. Frankl. Inst.* **2021**, *358*, 6634–6665. [CrossRef]
14. Doyle, J.; Stein, G. Multivariable feedback design: Concepts for a classical/modern synthesis. *IEEE Trans. Autom. Control* **1981**, *26*, 4–16. [CrossRef]

15. Safonov, M.; Laub, A.; Hartmann, G. Feedback properties of multivariable systems: The role and use of the return difference matrix. *IEEE Trans. Autom. Control* **1981**, *26*, 47–65. [CrossRef]
16. McFarlane, D.; Glover, K. A loop-shaping design procedure using H/sub infinity / synthesis. *IEEE Trans. Autom. Control* **1992**, *37*, 759–769. [CrossRef]
17. Genç, A.U.; Impram, S.T. A State-Space Algorithm for Designing H-inf Loop Shaping PlD Controllers. *IFAC Proc. Vol.* **2003**, *36*, 281–286. [CrossRef]
18. Hast, M.; Astrom, K.; Bernhardsson, B.; Boyd, S. PID design by convex-concave optimization. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 4460–4465. [CrossRef]
19. Franklin, G.F.; Powell, J.D.; Workman, M.L. *Digital Control of Dynamic Systems*, 3rd ed.; Addison-Wesley: Menlo Park, CA, USA, 1998.
20. Kotel'nikov, V.A. On the transmission capacity of 'ether' and wire in electric communications. *Usp. Fiz. Nauk* **2006**, *176*, 762–770. Peprint. The original paper was published in 1933. [CrossRef]
21. Åström, K.J.; Wittenmark, B. *Computer-Controlled Systems: Theory and Design*, 3rd ed.; Prentice Hall Information and System Sciences Series; Prentice Hall: Upper Saddle River, NJ, USA, 1997.
22. Chida, Y.; Nishimura, T. Discretization Method of Continuous-Time Controllers Based on Frequency Response Fitting. *Sice J. Control. Meas. Syst. Integr.* **2008**, *1*, 299–306. [CrossRef]
23. Watkins, D.S. *Fundamentals of Matrix Computations*, 3rd ed.; Pure and Applied Mathematics; Wiley: Hoboken, NJ, USA, 2010.
24. Ragazzini, J.R.; Zadeh, L.A. The analysis of sampled-data systems. *Trans. Am. Inst. Electr. Eng. Part II Appl. Ind.* **1952**, *71*, 225–234. [CrossRef]
25. Tsipkin, Y.Z. Theory of intermittent regulation. *Avtom. Telemekhanika* **1949**, *10*, 189–224.
26. Doetsch, G.; Herschel, R. *Anleitung zum Praktischen Gebrauch der Laplace-Transformation und der Z-Transformation*, 5th ed.; korrigierter nachdr. der 3., neu neubearb. aufl ed.; Oldenbourg: München, Germany, 1985.
27. Lyons, R.G. *Understanding Digital Signal Processing*, 2nd ed.; Prentice Hall PIR: Upper Saddle River, NJ, USA, 2004.
28. Zames, G. On the input-output stability of time-varying nonlinear feedback systems Part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE TRansactions Autom. Control* **1966**, *11*, 228–238. [CrossRef]
29. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design*, 2nd ed.; John Wiley: Hoboken, NJ, USA, 2005.
30. Zhou, K.; Doyle, J.C. *Essentials of Robust Control*; Prentice Hall: Upper Saddle River, NJ, USA, 1998.
31. Boyd, S.P.; Barratt, C.H. *Linear Controller Design: Limits of Performance*; Prentice Hall Information and System Sciences Series; Prentice Hall: Englewood Cliffs, NJ, USA, 1991.
32. Apkarian, P.; Gahinet, P. A convex characterization of gain-scheduled H/sub ∞/ controllers. *IEEE Trans. Autom. Control* **1995**, *40*, 853–864. [CrossRef]
33. Briat, C. *Linear Parameter-Varying and Time-Delay Systems: Analysis, Observation, Filtering & Control*; Advances in Delays and Dynamics; Springer: Berlin/Heidelberg, Germany, 2015; Volume 3. [CrossRef]
34. Tóth, R.; Heuberger, P.S.C.; Van den Hof, P.M.J. LPV system identification using series expansion models. In *Linear Parameter-Varying System Identification*; WORLD SCIENTIFIC: Singapore, 2011; pp. 259–294. [CrossRef]
35. Duan, G.R.; Yu, H.H. *LMIs in Control Systems*; CRC Press: Boca Raton, FL, USA, 2013. [CrossRef]
36. Youla, D.; Jabr, H.; Bongiorno, J. Modern Wiener-Hopf design of optimal controllers–Part II: The multivariable case. *IEEE Trans. Autom. Control* **1976**, *21*, 319–338. [CrossRef]
37. Glover, K.; Sefton, J.; Mcfarlane, D. A Tutorial on Loop Shaping using H-Infinity Robust Stabilization. *IFAC Proc. Vol.* **1990**, *23*, 117–126. [CrossRef]
38. Phillips, C.L.; Parr, J.M.; Riskin, E.A. *Signals, Systems, and Transforms*, 4th ed.; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2008.
39. Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK; New York, NY, USA, 2004.
40. Lipp, T.; Boyd, S. Variations and extension of the convex–concave procedure. *Optim. Eng.* **2016**, *17*, 263–287. [CrossRef]
41. Grant, M.; Boyd, S. CVX: Matlab Software for Disciplined Convex Programming, Version 2.1. 2014. Available online: http://cvxr.com/cvx (accessed on 6 March 2024).
42. Lofberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508), Taipei, Taiwan, 2–4 September 2004; pp. 284–289. [CrossRef]
43. Tütüncü, R.H.; Toh, K.C.; Todd, M.J. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.* **2003**, *95*, 189–217. [CrossRef]
44. ApS, M. *The MOSEK Optimization Toolbox for MATLAB Manual*, Version 10.1.27. 2024. Available online: https://docs.mosek.com/10.1/toolbox.pdf (accessed on 6 March 2024).
45. Higham, N.J. *Accuracy and Stability of Numerical Algorithms*, 2nd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002.
46. Wood, R.; Berry, M. Terminal composition control of a binary distillation column. *Chem. Eng. Sci.* **1973**, *28*, 1707–1717. [CrossRef]
47. Konkov, A.E.; Mitrishkin, Y.V.; Korenev, P.S.; Patrov, M.I. Robust Cascade LMI Design of MIMO Control System for Plasma Position, Current, and Shape Model with Time-Varying Parameters in a Tokamak. *IFAC-PapersOnLine* **2020**, *53*, 7344–7349. [CrossRef]

48. Mitrishkin, Y.V.; Korenev, P.S.; Konkov, A.E.; Kartsev, N.M. Suppression of Vertical Plasma Displacements by Control System of Plasma Unstable Vertical Position in D-Shaped Tokamak. *Autom. Remote Control* **2022**, *83*, 579–599. [CrossRef]
49. Konkov, A.E.; Mitrishkin, Y.V. Comparison Study of Power Supplies in Real-Time Robust Control Systems of Vertical Plasma Position in Tokamak. *IFAC-PapersOnLine* **2022**, *55*, 327–332. [CrossRef]
50. Konkov, A.E.; Korenev, P.S.; Mitrishkin, Y.V.; Balachenkov, I.M.; Kiselev, E.O. Real-Time Plasma Magnetic Control System with Equilibrium Reconstruction Algorithm in the Feedback for the Globus-M2 Tokamak. *Plasma Phys. Rep.* **2023**, *49*, 1552–1559. [CrossRef]
51. Deadman, E.; Relton, S.D. Taylor's theorem for matrix functions with applications to condition number estimation. *Linear Algebra Its Appl.* **2016**, *504*, 354–371. [CrossRef]
52. Hjørungnes, A. *Complex-Valued Matrix Derivatives: With Applications in Signal Processing and Communications*; Cambridge University Press: Cambridge, UK, 2011.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.