UNIVERSITY OF CALIFORNIA SAN DIEGO

Robust Input Shaping for Linear Time-Invariant Models with Uncertainties

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Alicia Dautt-Silva

Committee in charge:

Professor Raymond A. de Callafon, Chair
Professor Thomas R. Bewley
Professor Robert R. Bitmead
Professor Maurício C. de Oliveira
Professor Philip E. Gill

2022

The dissertation of Alicia Dautt-Silva is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

A mis papás Alicia y Alejandro, que son un ejemplo de trabajo, esfuerzo y perseverancia.

A mi doctora y hermana Joan, para quien quisiera yo ser el mismo ejemplo.

A mis terapeutas de 4 patitas, Tita, Luna, y Colin.

Y a mi esposo y mejor amigo Daniel, que no sólo comparte los triunfos conmigo, sino la carga por igual.

EPIGRAPH

All grown-ups were once children – although few of them remember it.

— Antoine de Saint-Exupéry

Our life together is so precious

Together we have grown, we have grown

Although our love is still special

Let's take a chance and fly away

Somewhere alone

— John Lennon

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

A few years ago I decided to start this adventure, unsure how the path would be. This was my adventure. But many people supported me on this and truly made it happen.

The first person in this long list is my dear advisor professor Raymond A. de Callafon, who dared to take an old newbie under his mentorship and had the patience all these years to guide my effort. Thanks for all his time and guidance (academical and practical). This just would have not happened without him.

I am fortunate to have a "dream team" committee of professors whose guidance since the instruction years and help for the development of this work is most valuable. Professor Bob Bitmead is one the best professors and his writing lessons are the most valuable ones on how to analyze and explain a topic. I'll be waiting for his book on Standard Notation. Professor Tom Bewley was the greatest challenge for every course, and I still remember the most challenging midterms. His NR text is still my first go-to book for all topics, from linear algebra to controls. Professor Maurício de Oliveira was my first interaction with the controls world in academia, and first impressions last forever. His practical and common sense advice is always the best. Thanks to professor Philip Gill for his kindness and support, for allowing me to audit his Optimization courses and sharing the material these past years.

Mi vida en la academia empezó en México, donde siempre recuerdo con mucho cariño al profesor Dagoberto Cruz Sibaja, cuya invitación al mundo de las Olimpiadas de Matemáticas definió en gran parte mi camino: mi carrera, conocer a mi esposo y llegar a UCSD al Doctorado.

In the last quarters I finally got myself into the Writing Hub program. This was the best decision I made towards the end of my PhD program. I really cannot think of a way to complete my final tasks without them. The seminars and support they offer is something all graduate students should experiment as an exercise at least once. Students

might be surprised of how helpful it is. A big thanks to my writing consultants Beverly French and Carolina Mayes.

Thanks for all their help to the MAE PhD program coordinator Krista Meinke, and the Graduate Academic Affairs Advisor Van Lee-Yamamoto. Thanks to Lusia Veksler from MAE department for all this years of support, for answering lots of questions and solving incredible tasks.

A big thanks to my former boss Dr. Dennis Braunreiter, who encouraged me to pursue the PhD program. And to all my friends and colleagues for their (technical and moral) support: Dr. Bob Jacques, Steve Chang, Gene Foltyn, Omez Mesina, Dr. William Whitty, Dr. Paul Frihauf, professor Dean Richert, Nora Unwin, Shawn Tsai, Loris Moussessian, Eric Leboffe, Jordan Prazak, Jason Anderson, Catalin Popescu, David Boe, Jim Pate, Dr. Maurice Pessot, Dr. Rick Venger, Vince Rye, Wes Roberts, Amy Tran, Dr. Bill Capinski, Patrick Newton, Tom Morrison, Tim Herrin, Tanya Crowley, Dr. Alan Sussman, Eron Johnson and Dr. Amir Valibeygi. And in particular a big thanks to Dr. Leo Camacho, who has been there for my technical "nearly a doctor" questions, for all the tips and the time for venting.

Thanks to General Atomics for allowing me the time and supporting me during the whole PhD program. And to my current boss Dr. Dave Humphreys, who has created a great and healthy team environment for the Plasma Controls group, and to all my teammates, in particular to Dr. Will Wehner for his guidance during my first year at the Energy group. To my partners in "lab": Dr. Gil Collins, Dr. Chris McGuffey, Dr. Mario Manuel, Dr. Brian Sammuli and Dr. Mike Jaris who make the daily challenges a happy adventure.

And finally, thanks to my husband Daniel for all the reading and reviewing of typos and grammatical errors in the past years. I really appreciate the effort on reading topics completely unrelated to your field. Thank you for taking care of me during this time and for your unconditional support.

Chapter 4, in part, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Multivariable Dynamic Input Shaping for Two-Axis Fast Steering Mirror." *IFAC-PapersOnLine*, Volume 53, Issue 2, 2020, pp. 8644-8649. The dissertation author is the first author of this article.

Chapter 5, in part, has been submitted for publication A. Dautt-Silva and R. A. de Callafon (2018) "Dynamic Input Shaping and Image Motion Compensation for a Dual-Mirror System." The dissertation author is the first author of this article.

Chapter 6, in full, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Reference signal shaping for closed-loop systems with causality constraints." *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1838-1843. The dissertation author is the first author of this article.

Chapter 7, in part, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Robust Input Shaping Design for Uncertain Second-order Dynamics Systems." *2022 CCTA Conference on Control Technology and Applications*. The dissertation author is the first author of this article.

Chapter 8, in part, has been submitted for publication A. Dautt-Silva and R. A. de Callafon (2018) "Robust Input Shaping for Second-order Dynamic Systems with Parameters Variations." The dissertation author is the first author of this article.

Appendix A, in full, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Optimal Input Shaping with Finite Resolution Computed from Step-Response Experimental Data." *2018 Annual American Control Conference (ACC)*, pp. 6703-6708. The dissertation author is the first author of this article.

Appendix B, in full, is a reprint of A. Dautt-Silva and R. A. de Callafon (2022) "Application of Dynamic Input Shaping for a Dual-Mirror System." *ASME. J. Dyn. Sys., Meas., Control*, 144(4). The dissertation author is the first author of this article.

VITA

2000	B.S. in Electronic Cybernetics *summa cum laude*, CETYS University, Mexicali, BC, México
2017	M.Sc. in Engineering Sciences (Mechanical Engineering), University of California San Diego
2022	Ph.D. in Engineering Sciences (Mechanical Engineering), University of California San Diego
2017	Staff Controls Engineer at General Atomics.
2010-2017	Automation Engineer at CYMER an ASML company.

PUBLICATIONS

A. Dautt-Silva, "Systematic Dynamic modeling based on step-response experiments,", thesis for Master of Science. 2017.

A. Dautt-Silva and R. A. De Callafon, "Optimal Input Shaping with Finite Resolution Computed from Step-Response Experimental Data," 2018 Annual American Control Conference (ACC), 2018, pp. 6703-6708, doi: 10.23919/ACC.2018.8430805.

Alicia Dautt-Silva, Raymond A. de Callafon, "Multivariable Dynamic Input Shaping for Two-Axis Fast Steering Mirror," IFAC-PapersOnLine, Volume 53, Issue 2, 2020, Pages 8644-8649, ISSN 2405-8963, doi: 10.1016/j.ifacol.2020.12.464.

A. Dautt-Silva and R. A. de Callafon, "Reference signal shaping for closed-loop systems with causality constraints," 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 1838-1843, doi: 10.1109/CDC45484.2021.9683283.

Dautt-Silva, A., and de Callafon, R. A. (January 25, 2022). "Application of Dynamic Input Shaping for a Dual-Mirror System." ASME. J. Dyn. Sys., Meas., Control. April 2022; 144(4): 044503. doi: org/10.1115/1.4053188.

Dautt-Silva, A., and de Callafon, R. A. (submitted January 2022). "Dynamic Input Shaping and Image Motion Compensation for a Dual-Mirror System."

Dautt-Silva, A., and de Callafon, R. A. (submitted February 2022). "Robust Input Shaping Design for Uncertain Second-order Dynamics Systems." 2022 Conference on Control Technology and Applications.

Dautt-Silva, A., and de Callafon, R. A. (submitted April 2022). "Robust Input Shaping Design for Uncertain Second-order Dynamics Systems."

ABSTRACT OF THE DISSERTATION

Robust Input Shaping for Linear Time-Invariant Models with Uncertainties

by

Alicia Dautt-Silva

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2022

Professor Raymond A. de Callafon, Chair

Working with a physical system, the search for a model could be computed mainly via physical equations or system identification when input/output data is available. A model could be thoroughly detailed but it is never an exact representation of the real physical system; and in many applications, noise is clearly present. *Model uncertainty* refers to the differences between the models and the real system. Parametric uncertain models consist of a set of models derived from a range of the uncertain parameters.

In linear time-invariant (LTI) control systems, input shaping (IS) is a technique originally used for defining a shaped command input to eliminate or reduce unwanted

system vibration. With the input shaping schemes being inherently open-loop, uncertainties in the model can lead to system performance degradation. By its nature, the nominal model-based input shaping cannot compensate for errors in the model, since it rejects any possible variations in the parameters. For parametric uncertain models, a process to compute a robust input signal that works for the set of models derived from the uncertain parameters is required.

This work presents the analysis on input shaping design for model uncertainty. Analyzing the parametric variability in the transfer function, it is assumed that a variation within an interval is known and is applied via a normalized weighted perturbation matrix and via a finite number of uncertainties. Robust input shaping is based on the extreme models, initially derived from the extreme values of the uncertain parameters. Robustness is then achieved by considering the extreme cases in the range of the parameters variations. The result is a single input signal that guarantees tracking of a planned trajectory within a specified accuracy and operating constraints. The solution proposed in this manuscript is to address the task of trajectory planning as multivariable optimal input shaping design that takes into account both dynamics and uncertainty of the model. The contribution of the proposed analysis is a multivariable input shaping design, solved via convex optimization to compute a robust input signal that guarantees tracking of a planned trajectory within a specified accuracy and within the operating constraints.

Chapter 1

Introduction

1.1 Motivation

The motivation of the study for this dissertation is based in a dual-mirror configuration system. To illustrate the effectiveness of dynamic input shaping for FSMs, a beam target trajectory problem with fast dynamic transitions (a square trajectory) is chosen. The beam source will reflect in the mirrors while attempting to follow the trajectory.

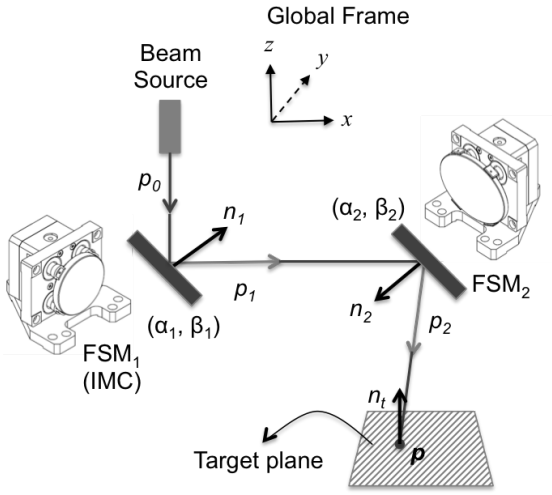


Figure 1.1: Mechanical layout of a dual fast steering mirror system with two mirrors and a beam source.

The FSMs have strong mechanical vibrations and cross coupling between the rotational axes. Applying the target command for the FSM, results not only in mechanical vibrations of the angular rotation of the FSM around one of its main orthogonal axis; it also creates a dynamic coupling in the form of vibrations around the other orthogonal axis. The mirror dynamics is not capable of following the trajectory for fast transition of the desired target, Figure 1.2.

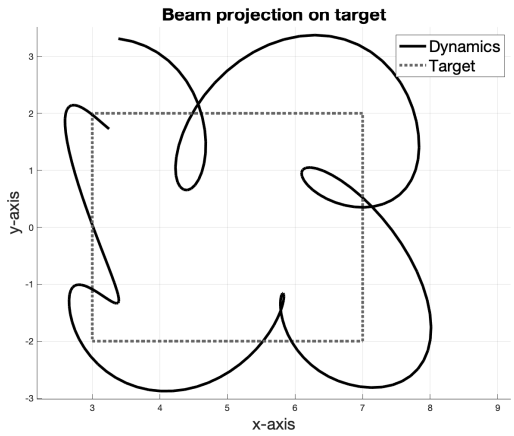


Figure 1.2: Comparison of the target and the dynamics of the system.

This dissertation focuses in developing a method for dynamic trajectory planning of systems that guarantees that optical signals are projected within a pre-specified accuracy on a target plane.

1.2 Problem statement and contributions

1.2.1 Literature review

In linear time-invariant (LTI) control systems, the concept of input shaping is a technique originally used for defining a shaped command input to eliminate or reduce unwanted system vibration [1]. The input profile brings the system from an initial point to

a desired target. Input shaping is usually formulated as an open-loop problem, which can be reformulated as a convex optimization problem by imposing linear constraints on the input and output signals. By approaching the input shaping problem as a general convex optimization problem, the desired response can be further optimized by defining constraint functions and limits or bounds for these constraints. In general, input shaping is capable of adapting to single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems in continuous- and discrete-time [2, 3, 4].

In input shaping, the original input (unmodified) signal is passed through an input shaper and this new shaped signal is fed to the system. The purpose of this new signal is to remove oscillation that can be caused by the unmodified input signal [5]. Input shapers used in the design of control systems decrease the overshoot or undershoot, as well as the time for a steady response in an oscillating system [6]. For the most common methods of input shaping, e.g. zero vibration (ZV), zero vibration derivative (ZVD) and extra insensitive (EI); the primary measure of performance is the residual vibration. New tools have emerged to solve the IS problem [7, 8]. A further analysis on the past 50 years of input shaping is found in [9].

Models are not an exact match with real systems, and their quality depends on how close their responses are to those of the true plants; a set of models is required to compensate for the deficiencies. To sum up, a model that duplicates the true physical plant behavior can never be computed. It is at this point where the term *uncertainty* appears, referring to the differences between the models and the real system [10]. In robust control, the presence of uncertainty and disturbances are taken into consideration [10]. To account for the presence of uncertainty the system is not modeled as a single system. This work addresses parametric uncertainty for input shaping considering a set $\mathcal{G}$ of discrete-time models, also referred as *interval plant* or *family of plants* [11].

However, with input shaping schemes being inherently open-loop, uncertainties in

the model can lead to system performance degradation [12]. The fact that input shaping does not take into account parametric variability has been approached by different methods where further measures are taken into account, such as: weighting the ranges of system parameter values according to the expected modeling errors [13, 14]; allowing the shaper to contain negative impulses [15]; a probabilistic approach of the uncertain model parameters with a polynomial chaos approach [16]; design of closed loop input shapers [17]; just to mention a few approaches. These methods show significant improvement, although most of them are sill based on the ZV, ZVD and EI methods.

1.2.2 Problem formulation

A model $G(\theta) \in \mathcal{G}$ with uncertainty on the parameter θ constitutes the set of models for which input shaping must be performed. We assume that the set $\mathcal{G}$ is represented by a nominal model $G(\hat{\theta})$ with a nominal parameter $\hat{\theta}$ and parameter uncertainty modeled by a finite number of independent perturbations. A robust input shaping design produces an input signal that is guaranteed to hold for the nominal model as well as for all the uncertain models within the set $\mathcal{G}$. The input signal is said to be robust if it holds for all the models while satisfying the defined constraints.

Applying input shaping to the set $\mathcal{G}$ can be validated by applying it to the extreme cases. The definition of the extremes is key to this process. We can start by defining the set $\mathcal{G}$ as a parametrized structured set where the nominal model $G(\hat{\theta})$ has one or more parameters only known to lie within an interval. These terms are the uncertainties on the system, reflected as parametric variability. Robustness for input shaping results in the performance holding for all $\mathcal{G}$, while applying the computed input for the extreme models.

This work presents two approaches to input shaping design for model uncertainty. The first approach consists of analyzing the parametric variability in the coefficients of the transfer function of the model. For the second approach, model uncertainty is analyzed

in the form of an impulse response. For both approaches, it is assumed that a variation within an interval is known and is applied via a normalized weighted perturbation matrix and via a finite number of uncertainties.

A closed-loop system might compensate for the uncertainties and disturbances. As such, input shaping can be formulated for the reference signal in a closed-loop system subject to linear constraints on amplitude and rate-of-change.

The purpose of this research is the computation of optimal input signals for output tracking of a dynamic system. The input shaping problem formulation is as follows: a desired output is defined for the dynamic model and an input is designed to track it. The main objective of this work is to design an input shaping technique in order to enhance control inputs such that a robust response can be achieved.

The solution proposed in this manuscript is to address the task of trajectory planning as multivariable optimal input shaping design that takes into account dynamics and uncertainty of the model. The contribution of the proposed procedure is a multivariable input shaping design, solved via convex optimization to compute a robust input signal that guarantees tracking of a planned trajectory within a specified accuracy and the operating constraints.

For this dissertation, several input shaping problems were simulated.

- The input shaping trajectory for a single mirror optical system, for a dual-mirror optical system and for a mass-spring damper system; all solved as a LP problem.
- An exercise on a second-order system and the improvement of input shaping in a closed-loop approach solved as a QP problem.
- An input shaping design for a 2dof mass-spring-damper system model with uncertainties with two approaches on the uncertainty analysis, solved as a QP problem.

1.3 Structure of the Dissertation

The outline of the dissertation is as follows. A summary of the different input shaping methods illustrated with a simulation is presented in Chapter 2. Chapter 3 is a derivation of the input shaping problem solved via convex optimization. Chapter 4 presents an illustration of the input shaping method in open-loop for a fast-steering mirror (FSM) problem. Chapter 5 presents an approach of compensating the open-loop input shaping process for a dual-mirror system via hardware, utilizing a second mirror to compute image motion compensation (IMC). Chapter 6 presents the input shaping method in a second-order closed-loop system, illustrating the benefit of input shaping in a closed-loop system. In Chapter 7, the robust input shaping process is introduced for a second-order model, via parametric variations of the coefficients in the transfer function of the model. Chapter 7 presents the robust input shaping process for a model via the impulse response of the nominal transfer function of the model.

Appendix A is a summary of different open-loop input shaping methods applied to two exercises. Appendix B is a technical brief on an open-loop input shaping exercise for a dual-mirror system to create the IMC dynamics. Appendix C follows the process for forward kinematics and inverse kinematics.

1.4 Nomenclature

The notation defined in Table 1.1 is valid for all sections. Additional variables introduced in some sections are defined there.

Table 1.1: Symbols.

Symbol	Description
m	number of outputs
p	number of inputs
Q	number of uncertainties
N	number of points
G	model
δ_i	uncertainties
$\mathbf{u} \in \mathbb{R}_{pN\times 1}$	inputs
$\mathbf{y} \in \mathbb{R}_{mN\times 1}$	outputs
$\underline{y} \in \mathbb{R}_{mN\times 1}$	output lower bounds
$\bar{y} \in \mathbb{R}_{mN\times 1}$	output upper bounds
$\underline{u} \in \mathbb{R}_{pN\times 1}$	input lower bounds
$\bar{u} \in \mathbb{R}_{pN\times 1}$	input upper bounds
$u, u_n \in \mathbb{R}_{pN\times 1}$	input signal for nominal model
$u_r \in \mathbb{R}_{pN\times 1}$	robust input signal
$P \in \mathbb{R}_{mN\times pQ}$	perturbation matrix
$\theta \in \mathbb{R}_{mN\times pN}$	coefficients
$\hat{\theta} \in \mathbb{R}_{mN\times pN}$	nominal coefficients
$\delta \in \mathbb{R}_{Q\times 1}$	uncertainty vector $[\delta_1, \ldots, \delta_Q]$

Chapter 2

Input shaping: an overview

Input shaping is defined as a feed-forward control technique first proposed in the 1980s. The initial purpose of IS was to shape the system input (commands) to eliminate the system vibration in the response. The concept has expanded through the years finding general applications.

From input shaping, first we explore zero vibration (ZV) [1]. Based on the knowledge that most of the systems will vibrate when an impulse is given, applying a second impulse to the system can cancel the vibration induced by the previous impulse. In ZV, is needed to derive the amplitude and locations of the impulse commands [18].

The use of input shaping techniques in control systems helps to reduce the overshoot/undershoot and the settling time of under-damped systems. Finite impulse response (FIR) filters are a different approach [19]. Unlike ZV input shapers, which are impulse-based, the FIR filters preshape an input command to reduce the vibration [4].

The inverse model and input shaping techniques approaches enforce no limits on the computed input or constraints on the system response. Convex optimization will also be explored with a view to applying both constraints, via linear program (LP) problem [20]. Finally mixed integer linear programming (MILP), in which the at least one of the

input variables is restricted to integers, will be applied [21, 22].

This chapter is an overview of the possible input shaping techniques to obtain an ideal response. The example utilized to illustrate this process is a mass-spring-damper system experiment. An Educational Control Product (ECP) is used to verify the results obtained for the mass-spring-damper system. The main objectives of this chapter is to introduce the reader to input shaping techniques in order to enhance control inputs such that a more desirable system response can be achieved.

2.1 Mass-spring-damper System

The mass-spring-damper system is used for verification and interpretation of the results we get with the processes we compare; since it's a well understood dynamic system, ZV, FIR, LP and MILP will be illustrated. We consider a step input on the mass-spring-damper system and compute the inputs for each method based on the step-response. The purpose is to move mass m_1 as fast as possible to a desired position. The objective is to find the right input signal to obtain the ideal result.

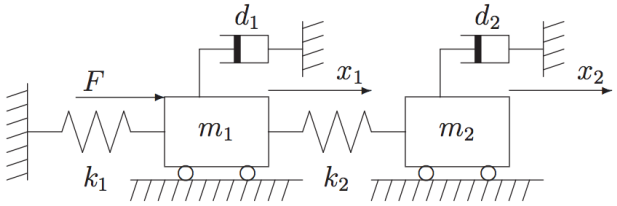


Figure 2.1: Cascade mass-spring-damper system.

The two-degree-of-freedom (2DOF) mechanical system depicted in Figure 2.1 consists of 2 masses, m_1 and m_2, each having positioning freedoms x_1 and x_2, respectively. The masses are connected via spring elements having stiffness coefficients k_1 and k_2 (spring or flexible shaft). Additionally, to model the damping present in the system, a viscous damping, d_1 and d_2, are assumed to act on each of the masses in the mechanical system.

To obtain data, we work with the ECP model 210, rectilinear plant in Figure 2.2.

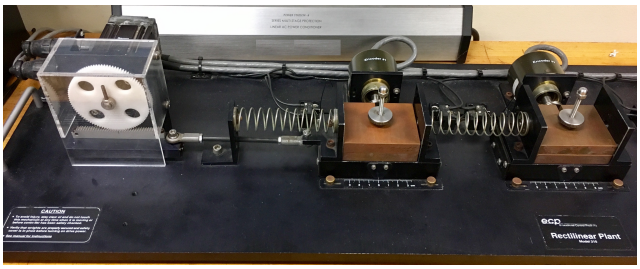


Figure 2.2: ECP Model 210.

2.2 Inverse Model Approach

For a given model, a straightforward approach is to check if the inverse is stable. We apply a step input signal the size of the output (once stable) to the inverse of the model, G^{-1}; this generates the ideal input.

In this case, the zeros are stable and an inversion can be performed on the model. With the inverse the ideal input can be computed, but there is no assurance that the computed ideal input is feasible, since we are working without any input constraints. The hardware may not have enough freedom to generate such an idealized input.

2.3 ZV and FIR filters

Two impulse responses can be superimposed such that the system moves forward without vibration after the input has ended. This is the two-impulse ZV input shaper. The amplitudes, A_i; and times, t_i; of the impulses are given by [23, 24]:

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \frac{1}{1+K} & \frac{K}{1+K} \\ 0 & 0.5T_d \end{bmatrix}, i = 1, 2, \tag{2.1}$$

where

$$K = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}, \tag{2.2}$$

T_d is the damped period of vibration, and ζ is the damping ratio. The convolution is performed between the original input and the amplitude of the first and second impulse, shifted by one-half of the damp vibration period. A constraint for the amplitude A_i of the impulse has to be met, $A_i \geq 0$ for $i = 1, \ldots, m$ and $\sum_{i=0}^{m} A_i = 1$.

The impulse convolution with a step input signal might not be the ideal approach for some systems. It creates smaller size steps until it reaches the full size step: a staircase command. Each impulse added will result in an output delay. Figure 2.3 shows a basic example of step command and impulse convolution as command shaping [25].

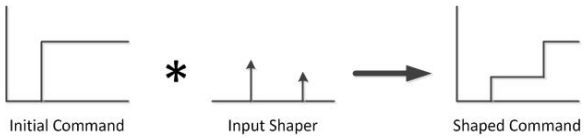


Figure 2.3: Input shaping a step to produce a staircase command.

Figure 2.4 shows the an improvement by utilizing the ZV method in regards of the overshoot and oscillation, while it does not add a delay.

In a different feedforward approach, the input signal and FIR filters are convoluted; this type of input shaper adds zeros to the system [3]. The poles in the z-plane are only at z=0 and are represented by [19]:

$$H(z) = \sum_{n=0}^{M} b_n \cdot z^{-n}, \tag{2.3}$$

where b_n are parameters of the filter. Similar to the ZV conditions, the FIR filter gain shall be 1.

Figure 2.5 shows there is no improvement for this system utilizing FIR filters. The 10^{th} and 20^{th} order fiters do not improve the output, only add a delay.

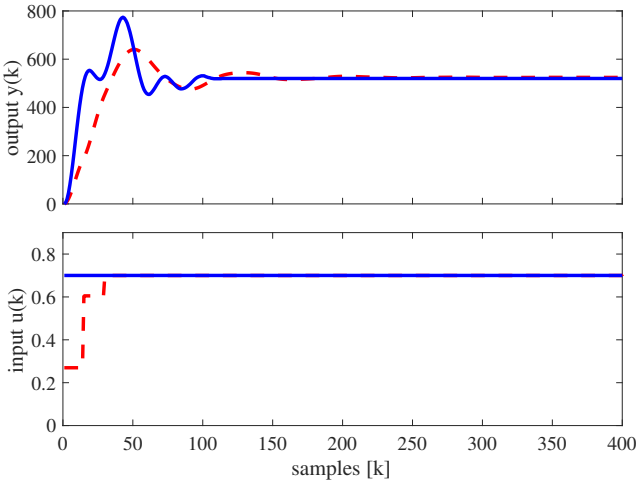


Figure 2.4: ZV for mass-spring-damper system. Top: output. Bottom: input. In each plot: data from mass-spring-damper experiment (solid blue line), data from ZV (red dotted line).

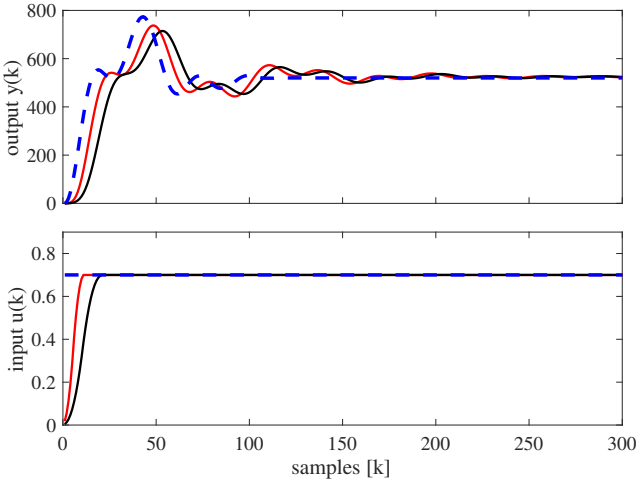


Figure 2.5: FIR filters for mass-spring-damper system. Top: output. Bottom: input. In each plot: data from mass-spring-damper experiment (dashed blue line), 10^{th} order FIR filter (red line), and 20^{th} order FIR filter (black line).

2.4 Convex Optimization

A convex optimization problems is LP, where the objective and all constraint functions are linear. The general formulation for this is [20]:

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad a_i^T x \leq b_i, \quad i = 1, ..., m, \end{aligned} \tag{2.4}$$

where x is the vector of optimization variables and a_i, b_i and c are parameters and constraint functions. The process is throughout described in Section 4.5.4.

Figure 2.6 schematically illustrates two different amplitude constraints [26]. In this example the first constraint during the initial response is a large one, due to the response initial oscillation. Once the response is settled, a second tight constraint suffices. The maximum and minimum constraints are defined as $\bar{u}$ and $\underline{y}$, respectively.

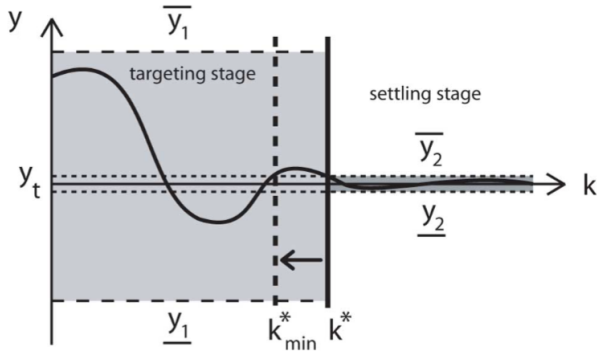


Figure 2.6: Definition of the output constraints.

k^* represents the samples to reach the steady state; if k^* is longer than the minimum necessary, the response will oscillate due to the freedom on time to reach the target. In order to find the minimum sample number k^*_{min}, the bisection method is used.

The bisection method consists of defining the problem as feasible, and start with the range $[k_{low}, k_{high}]$ which contains k^*. The convex feasibility problem it's solved at its midpoint $t = (k_{low} + k_{high})/2$, to determine whether the optimal value is in the lower or

upper half of the range, and update the range based on the result. This produces a new range, which contains the optimal value k^*, but has half the width of the initial one. This is repeated until the interval is within a valid maximum value [20].

Figure 2.7 shows the optimization for the mass-spring-damper system compared with the data from the experiment. The overshoot is completely removed and the response is steady at around the half of the samples than the original experiment.

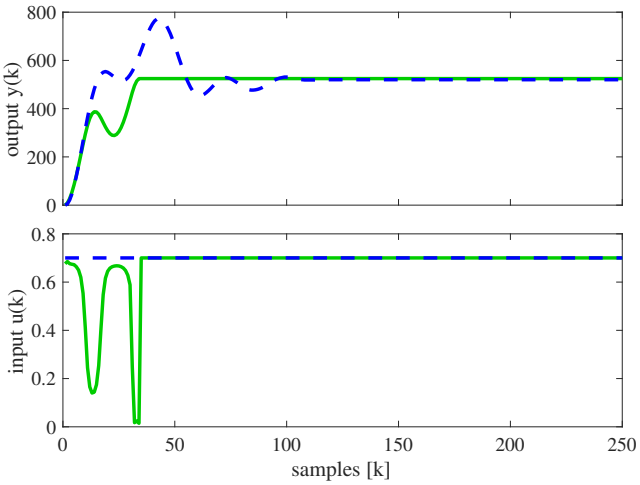


Figure 2.7: Optimization for mass-spring-damper system. Top: outputs. Bottom: inputs. In each plot: data from mass-spring-damper experiment (blue dashed line), and optimization (green solid line).

2.5 Mixed Integer Linear Programming

Since we have emphasized on the restrictions the input may be subject to based on the hardware, we explore further techniques such as MILP. Mixed integer linear programming solves LP problems in which at least one of the variables is restricted to integers [21]. Solutions use full design freedom on the input, but in the real system only input signals with particular constraints on the level/size can be applied. A valid approach could be rounding the values to the nearest integer, but this might affect the optimal solution. In those cases the MILP solution is given by [27].

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad a_i^T x \leq b_i, \quad i = 1, ..., m, \\ &\text{where} \quad x \in \mathbb{Z}. \end{aligned} \tag{2.5}$$

Figure 2.8 shows the results of the MILP technique applied to the mass-spring-damper system.

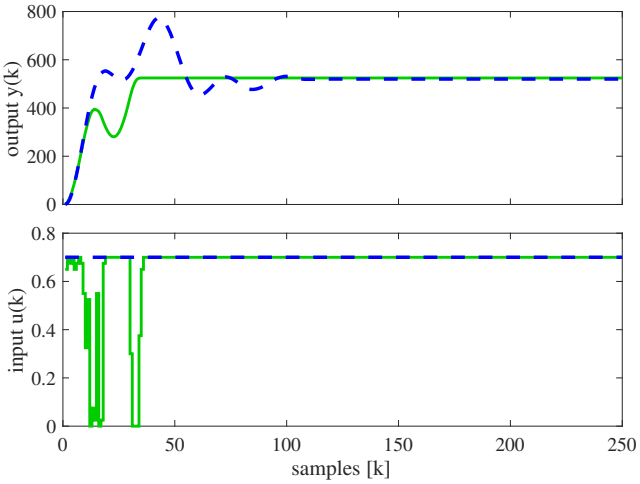


Figure 2.8: Computed fractional MILP and simulated response for mass-spring-damper system. Top: outputs. Bottom: inputs. In each plot: data from mass-spring-damper experiment (blue dashed line), and MILP (solid green line)

Figure 2.9 shows the MILP response of a binary type for the mass-spring-damper system.

To smooth the MILP response, a *fractional* MILP approach is performed. The problem is set as the MILP approach, but restricted to specific set of inputs $u(k) = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. The response is better since the input is not of binary type, and there is a smoother response from the initial shown in Figure 2.8. The response from fractional MILP is similar to the convex optimization response, with the advantage that the input values are restricted for hardware.

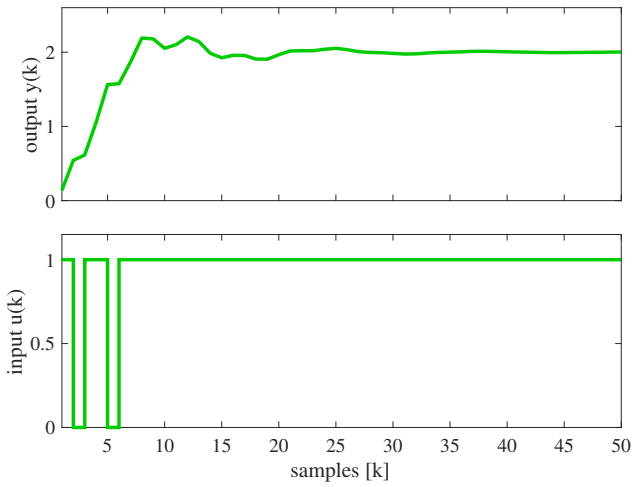


Figure 2.9: MILP for mass-spring-damper system. Top: output. Bottom: input.

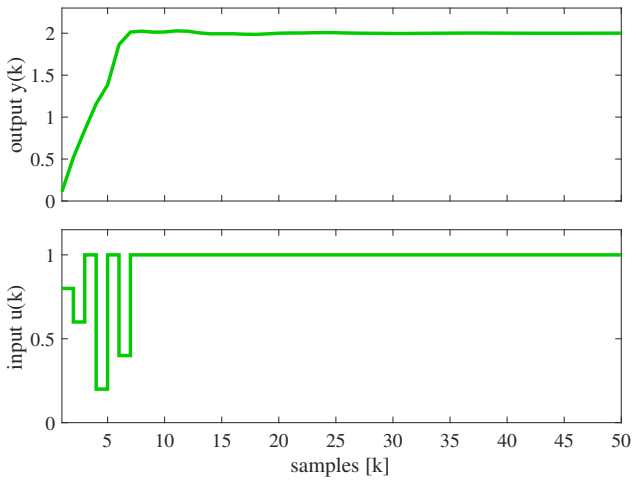


Figure 2.10: Fractional MILP for mass-spring-damper system. Top: output. Bottom: input.

2.6 Experimental Verification

In general, the convex optimization techniques give better results. In order to test the convex optimization technique, the experiment was performed on the ECP 210, from which the data to compute the model via system identification was obtained. The obtained input in Figure 2.7 is loaded as a trajectory (trj) file in the ECP. The results are shown in Figure A.10. The response accurately tracks the prediction.

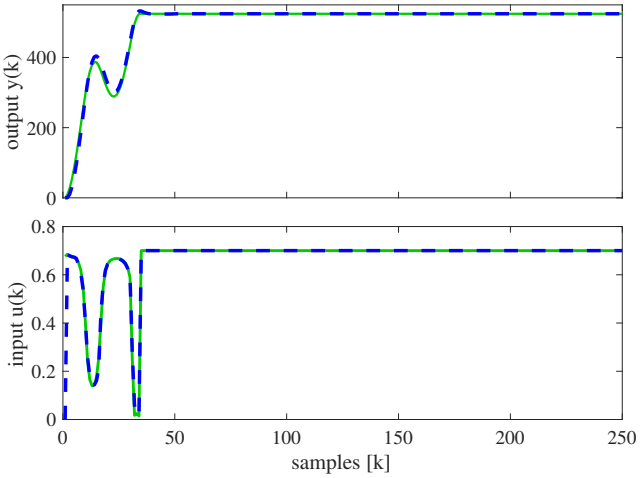


Figure 2.11: Computed optimization input and simulated response for two-mass system. Top: outputs. Bottom: inputs. In each plot: calculated optimization data (green solid line), and ECP test obtained data (blue dashed line).

Chapter 3

Open-loop input shaping process

For dynamic input shaping, knowledge of dynamics in the form of the state space model can be used to constrain the dynamic motion of the system. For writing the (linear) motion constraints, an approach similar to the one in [28] is followed to formulate the input design as a LP problem [20, 29, 30, 31].

3.1 Linear input and output constraints

Consider the nth order state space representation of a discrete-time model $G(q)$ with p inputs and m outputs given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{3.1}$$

sampled with a sampling time of Δ_T seconds and where $u(k) \in \mathbb{R}_{p\times 1}$, $y(k) \in \mathbb{R}_{m\times 1}$, $x(k) \in \mathbb{R}_{n\times 1}$. With zero initial conditions $x(0) = 0_{n\times 1}$ and $u(k) = 0_{p\times 1}$ for $k < 0$ are assumed. Designing a desired input $u(k) \in \mathbb{R}_{p\times 1}$ for (3.1) is the considered an open-loop input design and coined as an *input shaping problem*.

When designing a multi-variable input $u(k)$ to be applied to the open-loop system (3.1) or a multi-variable reference $r(k)$ to be applied to the closed-loop system, restriction will be imposed on the resulting input $u(k)$ and output $y(k)$. These restriction are:

- The p inputs in $u(k) \in \mathbb{R}_{p\times 1}$ are bounded between distinct time-varying lower amplitude bounds $\underline{u}(k) \in \mathbb{R}_{p\times 1}$ and time-varying amplitude upper bounds $\bar{u}(k) \in \mathbb{R}_{p\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{gathered} \underline{u}(k) \leq u(k) \leq \bar{u}(k), \;\; k = 0, 1, \ldots, N-1 \\ \text{where} \;\; \underline{u}, \bar{u} \in \mathbb{R}_{p\times 1} \end{gathered} \tag{3.2}$$

- The p inputs in $u(k) \in \mathbb{R}_{p\times 1}$ sampled at Δ_T seconds are rate limited between distinct time-varying lower rate bounds $\underline{d} \in \mathbb{R}_{p\times 1}$ and time-varying upper rate bounds $\bar{d} \in \mathbb{R}_{p\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{gathered} \underline{d}(k) \leq \tfrac{u(k)-u(k-1)}{\Delta_T} \leq \bar{d}(k), \;\; k = 0, 1, \ldots, N-1 \\ \text{where} \;\; \underline{d}, \bar{d} \in \mathbb{R}_{p\times 1} \;\; \text{and} \;\; u(-1) = 0 \end{gathered} \tag{3.3}$$

- The m outputs in $y(k) \in \mathbb{R}_{p\times 1}$ are bounded between distinct time-varying perturbation upper bounds $\underline{\varepsilon}(k) \in \mathbb{R}_{m\times 1}$ and time-varying perturbation lower bounds $\bar{\varepsilon}(k) \in \mathbb{R}_{m\times 1}$ around a time-varying desired output $y_{des}(k) \in \mathbb{R}_{m\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{gathered} \underline{\varepsilon}(k) \leq y(k) - y_{des}(k) \leq \bar{\varepsilon}(k), \;\; k = 0, 1, \ldots, N-1 \\ \text{where} \;\; \underline{\varepsilon}(k), \bar{\varepsilon}(k) \in \mathbb{R}_{m\times 1} \end{gathered} \tag{3.4}$$

- (For closed-loop) The m references in $r(k) \in \mathbb{R}_{m\times 1}$ are bounded between distinct time-varying lower amplitude bounds $\underline{r}(k) \in \mathbb{R}_{m\times 1}$ and time-varying amplitude upper

bounds $\bar{r}(k) \in \mathbb{R}_{r\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{aligned} &\underline{r}(k) \leq r(k) \leq \bar{r}(k), \quad k = 0, 1, \ldots, N-1, \\ &\text{where } \ \underline{r}(k), \bar{r}(k) \in \mathbb{R}_{m\times 1}. \end{aligned} \tag{3.5}$$

3.2 Open-loop input design

3.2.1 Feasibility for open-loop input design

Feasibility for open-loop input design is defined as the problem of finding feasible $p \times N$ values of the finite-time input signal

$$u(k) \in \mathbb{R}_{p\times 1}, \quad k = 0, 1, \ldots, N-1 \tag{3.6}$$

for the open-loop model given in (3.1), such that the linear conditions in (3.2), (3.3) and (3.4) hold. Feasibility of these conditions clearly depend on how fast and how close the output $y(k)$ needs to follow $y_{des}(k)$ within the bounds $\underline{\varepsilon}(k)$ and $\bar{\varepsilon}(k)$, but also on the fixed amplitude bounds $\underline{u}$, $\bar{u}$ and the fixed rate bounds $\underline{d}$, $\bar{d}$ imposed on the input $u(k)$. Any point $\hat{x}$ is said to be feasible it it satisfies the objective and all the constraints. The problem is said to be feasible if there exists at leas one feasible point. The set of all feasible points is called the feasible region, which implies the obtained solution is not unique. The choice of the constraints as well as the method used for optimization will reflect on the solutions obtained. Feasibility is checked by rewriting the linear conditions in (3.2), (3.3) and (3.4) into a linear matrix inequality (LMI) and check feasibility with a linear programming (LP) problem.

For vector notation purposes, the values of $u(k) \in \mathbb{R}_{p\times 1}, \quad k = 0, 1, \ldots, N-1$

contained in the matrix $U = \begin{bmatrix} u(0) & u(1) & \cdots & u(N-1) \end{bmatrix} \in \mathbb{R}_{p\times N}$ is first written as

$$\mathbf{u} = \text{vec}\{U\} = \begin{bmatrix} u(0)^T & u(1)^T & \cdots & u(N-1)^T \end{bmatrix}^T \in \mathbb{R}_{pN\times 1} \tag{3.7}$$

to obtain a vector $\mathbf{u}$ of desired multi-variable input values.

The first condition on the input $u(k)$ in (3.2) can be written as an LMI of the form $A_1\mathbf{u} \leq b_1$ where

$$A_1 = \begin{bmatrix} I_{pN} \\ -I_{pN} \end{bmatrix} \in \mathbb{R}_{2pN\times pN} \tag{3.8}$$

$$b_1 = \begin{bmatrix} \bar{\mathbf{u}} \\ -\underline{\mathbf{u}} \end{bmatrix} \in \mathbb{R}_{2pN\times 1}, \quad \begin{matrix} \bar{\mathbf{u}} = \text{vec}\left\{ \begin{bmatrix} \bar{u}(0) & \bar{u}(1) & \cdots & \bar{u}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN\times 1} \\ \underline{\mathbf{u}} = \text{vec}\left\{ \begin{bmatrix} \underline{u}(0) & \underline{u}(1) & \cdots & \underline{u}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN\times 1} \end{matrix} \tag{3.9}$$

with the definition of the vector $\mathbf{u}$ in (3.7). The second condition on the input $u(k)$ given in (3.3) can be written as an LMI of the form $A_2\mathbf{u} \leq b_2$, where

$$A_2 = \begin{bmatrix} E \\ -E \end{bmatrix} \in \mathbb{R}_{2pN\times pN}, \quad E = \begin{bmatrix} 0_{p\times p} & 0_{p\times p(N-1)} \\ -I_{p(N-1)} & 0_{p(N-1)\times p} \end{bmatrix} + I_{pN} \in \mathbb{R}_{pN\times pN} \tag{3.10}$$

$$b_2 = \begin{bmatrix} \bar{\mathbf{d}} \\ -\underline{\mathbf{d}} \end{bmatrix} \in \mathbb{R}_{2pN\times 1}, \quad \begin{matrix} \bar{\mathbf{d}} = \Delta_T \cdot \text{vec}\left\{ \begin{bmatrix} \bar{d}(0) & \bar{d}(1) & \cdots & \bar{d}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN\times 1} \\ \underline{\mathbf{d}} = \Delta_T \cdot \text{vec}\left\{ \begin{bmatrix} \underline{d}(0) & \underline{d}(1) & \cdots & \underline{d}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN\times 1} \end{matrix} \tag{3.11}$$

based on the definition of the vector $\mathbf{u}$ in (3.7).

For the condition (3.4) on the values of output $y(k)$ contained in $Y = \begin{bmatrix} y(0) & y(1) & \cdots & y(N-1) \end{bmatrix} \in \mathbb{R}_{m\times N}$ the same vector operation is performed

$$\mathbf{y} = \text{vec}\{Y\} = \begin{bmatrix} y(0)^T & y(1)^T & \cdots & y(N-1)^T \end{bmatrix}^T \in \mathbb{R}_{mN\times 1} \tag{3.12}$$

to obtain a vector $\mathbf{y}$ of desired multi-variable output values. The mapping of the input values $u(k)$ contained in $\mathbf{u}$ to the output values $y(k)$ contained in $\mathbf{y}$ is done with the linear state space description of the open-loop model in (3.1) via the convolution sum

$$y(k) = Du(k) + \sum_{m=1}^{k} CA^{m-1}Bu(k-m)$$

under the zero initial condition assumption. This results in

$$\mathbf{y} = \Phi\mathbf{u}, \quad \Phi = \begin{bmatrix} D & 0_{m\times p} & 0_{m\times p} & \cdots & \cdots & 0_{m\times p} \\ CB & D & 0_{m\times p} & \cdots & \cdots & 0_{m\times p} \\ CAB & CB & D & 0_{m\times p} & \cdots & 0_{m\times p} \\ \vdots & \vdots & \cdots & \ddots & \cdots & \vdots \\ CA^{N-3}B & CA^{N-2}B & \cdots & CB & D & 0_{m\times p} \\ CA^{N-2}B & CA^{N-1}B & \cdots & CAB & CB & D \end{bmatrix} \in \mathbb{R}_{mN\times pN} \tag{3.13}$$

as the linear matrix equality between $\mathbf{u}$ and $\mathbf{y}$. The condition (3.4) on the values of the output $y(k)$ can now be written in an LMI of the form $A_3\mathbf{u} \leq b_3$, where

$$A_3 = \begin{bmatrix} \Phi \\ -\Phi \end{bmatrix} \in \mathbb{R}_{2mN\times pN} \tag{3.14}$$

$$b_3 = \begin{bmatrix} \bar{\mathbf{y}} \\ -\underline{\mathbf{y}} \end{bmatrix} \in \mathbb{R}_{2mN\times 1}, \quad \begin{matrix} \bar{\mathbf{y}} = \mathrm{vec}\left\{ \begin{bmatrix} y_{des}(0) + \bar{\varepsilon}(0) & \cdots & y_{des}(N-1) + \bar{\varepsilon}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{mN\times 1} \\ \underline{\mathbf{y}} = \mathrm{vec}\left\{ \begin{bmatrix} y_{des}(0) - \underline{\varepsilon}(0) & \cdots & y_{des}(N-1) - \underline{\varepsilon}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{mN\times 1} \end{matrix} \tag{3.15}$$

With all three conditions covered as LMIs in the form $A_i\mathbf{u} \leq b_i,\ i = 1, 2, 3$, the

feasibility for an open-loop input design can be formulated as the question of

$$\exists \mathbf{u} \in \mathbb{R}^{pN\times 1}, \quad \text{subject to} \quad A\mathbf{u} \leq b, \quad \text{with}$$
$$A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \in \mathbb{R}_{4pN+2mN\times pN}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \in \mathbb{R}_{4pN+2mN\times 1}$$

The existence of $\mathbf{u}$ can be verified by adding a slack variable $\mathbf{z} \in \mathbb{R}_{4pN+2mN\times 1} \geq 0$ for all the linear constraints and solve the optimization

$$\min_{z,u} \sum \mathbf{z} \quad \text{subject to} \quad A\mathbf{u} + \mathbf{z} \leq b \quad \text{and} \quad \mathbf{z} \geq 0$$

as a solution for $\mathbf{z}$ and $\mathbf{u}$ with $\mathbf{z} = 0$ will amount to a feasible solution $\mathbf{u}$. With the definition of the full variable $\mathbf{x} = \begin{bmatrix} \mathbf{u}^T & \mathbf{z}^T \end{bmatrix}^T \in \mathbb{R}_{5pN+2mN\times 1}$ the above optimization is written as an LP problem

$$\min_{\mathbf{x}} f^T\mathbf{x}, \quad \text{subject to} \quad \begin{bmatrix} A & -I_{4pN+2mN} \\ 0_{4pN+2mN\times pN} & -I_{4pN+2mN} \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} b \\ 0_{4pN+2mN\times 1} \end{bmatrix} \quad \text{with}$$
$$f^T = \begin{bmatrix} 0_{1\times pN} & 1_{1\times 4pN+2mN} \end{bmatrix} \tag{3.16}$$

where the matrices $A_i,\ i = 1, 2, 3$ are defined in (3.8), (3.10) and (3.14) and the vectors $b_i,\ i = 1, 2, 3$ are defined in (3.9), (3.11) and (3.15) respectively. A feasible solution of the LP problem in (3.16) will lead to feasibility for the open-loop input design.

3.2.2 Optimal open-loop input design

Although the LP problem in (3.16) finds a feasible input design for the choices of the conditions formulated in (3.2), (3.3) and (3.4), the resulting input $u(k)$ is not optimal.

Only a feasible solution for both $\mathbf{u}$ and $\mathbf{z}$ is computed by minimizing the sum of the values of the slack variable $\mathbf{z}$. Depending on the optimization software used for solving the LP problem, different values of $\mathbf{u}$ may be found that may not be desirable. Example may include input sequences $u(k)$ that oscillate within the rate bounds (3.3) but do not influence the output constraints (3.4).

Additional uniqueness and an optimal value of the multi-variable input $u(k) \in \mathbb{R}_{p\times 1}$ for $k = 0, 1, \ldots, N$ can be enforced by also minimizing the finite-time variance of the rate-of-change of the input given by

$$\sigma_N^2 = \frac{1}{N}\sum_{k=0}^{N-1}\left(\frac{u(k)-u(k-1)}{\Delta_T}\right)^2 \tag{3.17}$$

to avoid penalizing the actual input values $u(k)$. The variance on the rate-of-change can be computed via the matrices used in the liner programming to verify that

$$\sigma_N^2 = \frac{1}{2}\mathbf{u}^T H\mathbf{u}, \quad H = \frac{2}{N\Delta_T^2}D^T D,$$

where D was defined in (3.10) for the LMI $A_2\mathbf{u} \leq b_2$ to satisfy the condition (3.3). Assuming the open-loop input design passed the feasibility test via the LP problem (3.16), the actual open-loop input design can be optimized via a QP problem.

$$\begin{gathered}\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T H\mathbf{u} + \mathbf{f^T u} \quad \text{subject to} \quad \mathbf{Au} \leq \mathbf{b}, \quad \text{with} \\ f^T = 0_{1\times pN},\end{gathered} \tag{3.18}$$

where the matrices $A_i,\ \ i = 1, 2, 3$ are defined in (3.8), (3.10) and (3.14) and the vectors $b_i,\ \ i = 1, 2, 3$ are defined in (3.9), (3.11) and (3.15) respectively. A feasibly solution to (3.18) now produces an input signal $u(k)$ that has minimized (3.17) under the conditions of (3.2), (3.3) on the input $u(k)$ and the condition (3.4) on the output $y(k)$.

Chapter 4

Multivariable Dynamic Input Shaping for Two-Axis Fast Steering Mirror

4.1 Chapter Abstract

A two-axis Fast Steering Mirror (FSM) is a commonly used mechanical component used in optical imaging and laser beam steering. This paper shows how inverse kinematic analysis and dynamic input shaping can be used to compute the two-axis input signals for the actuation of a FSM to be able to track a desired output trajectory. The approach is based on quasi-static kinematic analysis and dynamic modeling of a two-axis FSM multivariable motion from experimental step response data. It is shown how open-loop tracking can be improved by properly designed dynamic input shaping signals that take into account the inverse kinematics and dynamic response of the FSM.

4.2 Introduction

A Fast Steering Mirror (FSM) is a commonly used mechanical component in optical applications, such as astronomy, laser beam pointing systems, laser communication, opthalmology and laser cutting [32]. The analysis of how a laser propagates from the beam source via a mechanically adjustable mirror to a target is part of a multidisciplinary problem called "beam control" as described by [33]. Accuracy of beam steering is influenced by mechanical coupling, hysteresis and dynamics of the mirror itself [34], but also by atmospheric conditions in case of long-distance pointing [35]. Innovative approaches such as real-time or adaptive control of mirrors [36] can greatly improve the accuracy of beam steering, but require real-time measurements of the beam target location.

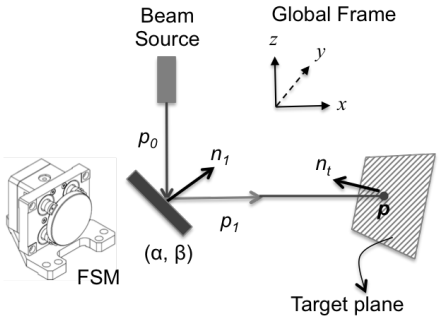


Figure 4.1: Beam and steering mirror system

Following the schematic description on Figure 4.1, this paper presents an open-loop approach to compute optimal steering signals of a FSM controlled by a Voice Coil Motor (VCM) on each axis to achieve a desired trajectory in a target plane. The VCM actuation of the FSM ensures minimal hysteresis effects, unlike the use of piezo-stacks for FSM actuation, discussed in [34]. Furthermore the proposed open-loop approach in this paper uses experimental data of the dynamic response of the FSM to formulate the optimal input shaping signals. Although detailed dynamic modeling of mirror dynamics is useful for analysis and design in high precision mirror systems [37], it is shown that a reliable

realization method presented in [38] can be used to directly formulate a linear multivariable dynamic model that captures the coupled dynamics of the two-axis VCM actuation of the FSM. Recent approaches of data-based input shaping have been recognized for input shaping of mechanical or MEMS devices, see e.g. [39], but often ignore the possible dynamic coupling effects in case of multi-axis actuation as seen in a FSM.

The input shaping for a two-axis FSM is broken down in two separable parts in this paper. The first part consists of the forward kinematics and dynamic analysis of the FSM. The forward kinematics uses the reflection matrix or mirror transformation matrix as defined in [40] to describe the quasi-static behavior of the beam reflection as a function of the two-axis rotation of FSM based on the work by [41] and [33]. Dynamic analysis of the FSM is based on step-response experiments that measure the dynamic response of the two-axis rotation of FSM and formulate a linear multivariable discrete-time dynamic model based on the realization algorithm described in [38].

The second part consists of the inverse kinematic analysis of the FSM, along with the computation of a set of dynamic voltage inputs to the VCMs used to steer the FSM. The inverse kinematics computes the desired quasi-static movement of the FSM from a desired target trajectory parametrization for beam forming. The set of dynamic voltage inputs to the VCMs of the FSM are solved by a Linear Programming (LP) problem similar to [42] but extended to the multivariable situation of the FSM. The LP problem uses the multivariable model to solve the input assignment under constraints to find a feasible solution for the input shaping of the FSM. Results are illustrated by the tracking of a square target reference to show the effectiveness of the proposed multivariable input shaping.

4.3 Mirror Beam-Pointing Kinematics

4.3.1 Rotated normal of the mirror

For the kinematic analysis we consider the steering mirror in Figure 4.2, whose rotation axes are located at the center of the mirror. The reflection problem of a beam by the mirror can be expressed as a matrix multiplication,

$$p_1 = Tp_0, \tag{4.1}$$

where T denotes the reflection matrix and p_0 and p_1 represent unit vectors in the directions of the incident beam and reflected beam, respectively. Following the approach of [41], we will write the reflection matrix T as a function of the rotation of the normal vector n of the mirror plane to accommodate inverse kinematic analysis.

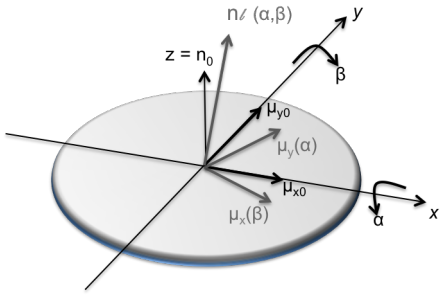


Figure 4.2: Steering mirror axes and normal

Without loss of generality, the local mirror frame is defined to be the x- and y-axes in the plane of the mirror and the normal n_0 is aligned with the a z-axis. The mirror can be rotated about both its x- and y-axes independently by the angles α and β, changing and the normal vector n of the mirror as follows. With the unit vectors

$$\mu_{x0} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \ \mu_{y0} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$

in the local mirror frame and the Euler transform (or rotation) matrices

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \; R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

based on the rotation angles α and β, the rotated unit vectors of the normal are given by

$$\mu_x(\beta) = R_y(\beta)\mu_{x0} = \begin{bmatrix} \cos\beta \\ 0 \\ -\sin\beta \end{bmatrix}, \tag{4.2}$$

$$\mu_y(\alpha) = R_x(\alpha)\mu_{y0} = \begin{bmatrix} 0 \\ \cos\alpha \\ \sin\alpha \end{bmatrix}. \tag{4.3}$$

As a result, the normal of the mirror $n_\ell(\alpha, \beta)$ in the local mirror frame is obtained with the cross product $n_\ell(\alpha, \beta) = \mu_x(\beta) \times \mu_y(\alpha)$ and leads to

$$n_\ell(\alpha, \beta) = \begin{bmatrix} \cos\alpha\sin\beta \\ -\cos\beta\sin\alpha \\ \cos\beta\cos\alpha \end{bmatrix}. \tag{4.4}$$

The normal $n_\ell(\alpha, \beta)$ of the mirror in the local mirror frame must be referenced to the global frame by considering the mounting angle of the mirror. The angle of rotation of the mirror relative to the global frame can again be described by a rotation around the x- and y- and z-axis of the global frame with rotation matrices $R_x(\phi)$, $R_y(\theta)$ and $R_z(\psi)$

defined by the matrices

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

respectively. As a result, the rotated normal of the mirror in global frame is given by

$$n(\phi, \theta, \psi, n_\ell(\alpha, \beta)) = R_x(\phi)R_y(\theta)R_z(\psi)n_\ell(\alpha, \beta) \tag{4.5}$$

where the normal $n(\cdot)$ in (4.5) is now also a function of the rotation angles (ϕ, θ, ψ) and the normal of the mirror n_ℓ in (4.4). With the mounting angles (ϕ, θ, ψ) all fixed, the rotation matrices can be characterized as constant, making

$$n(\alpha, \beta) = R\, n_\ell(\alpha, \beta) \tag{4.6}$$

with $n_\ell(\alpha, \beta)$ defined in (4.4) and $R = R_x(\phi)R_y(\theta)R_z(\psi)$.

4.3.2 Beam reflection

With the rotated normal $n(\alpha, \beta)$ given in (4.6), Figure 4.3 shows an incident beam p_0 and the reflected beam p_1. It should be noted that p_0 and $n(\alpha, \beta)$ span the beam plane and the reflected beam lies in the same beam plane, due to the assumption of a flat mirror.

With the notion of a perpendicular component $p_{1\perp}$ and parallel component $p_{1\parallel}$ [41], it is clear that $p_1 = p_{1\perp} + p_{1\parallel} = p_{0\perp} - p_{0\parallel}$ and

$$p_1 = p_0 - 2p_0 n(\alpha, \beta)n^T(\alpha, \beta),$$

which can be rewritten in the simplified matrix form $p_1 = T(n(\alpha, \beta))p_0$ as mentioned earlier

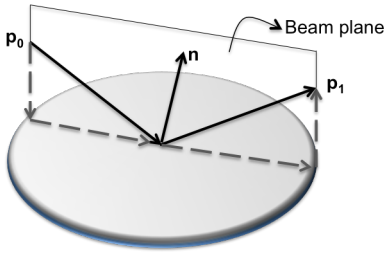


Figure 4.3: Incident and reflected beam decomposition

in (4.1). The analysis reveals that the reflection matrix $T(n(\alpha, \beta))$ is given by

$$T(n(\alpha, \beta)) = I_3 - 2n(\alpha, \beta)n^T(\alpha, \beta) \tag{4.7}$$

in terms of the normal vector $n(\alpha, \beta)$ of the mirror.

It should be noted that if the mirror is kept in a fixed position, the normal $n(\alpha, \beta)$ and the reflection matrix $T(n(\alpha, \beta))$ are constant, making the direction p_1 of the reflected beam only a function of the direction p_0 of the incident beam. Vice versa, keeping the incident beam direction fixed at p_0, the direction p_1 of the reflected beam can be adjusted by varying the rotation angle pair (α, β) of the mirror. In this case, the reflection matrix $T(n(\alpha, \beta))$ varies as function of the mirror rotation.

4.3.3 Beam intersection point

As the reflected beam p_1 leaves the mirror, it creates a target location on a subsequent target plane. This plane could be an imaging surface or another secondary mirror, as indicated in Figure 4.4.

Consider the reflected beam with a direction vector p_1, start point m_1, and a final point P_1 which is derived from the previous values. We have the segment $\overline{m_1 P_1}$ and

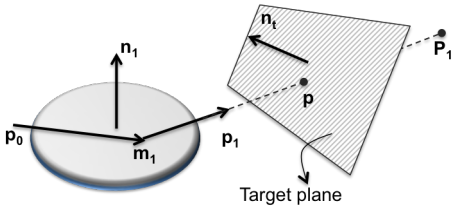


Figure 4.4: Reflected beam intersection with the target plane

following [43] we derive the intersection point.

$$P_1(m_1, p_1) = m_1 + kdp_1^T, \tag{4.8}$$

where k is a scale factor to assure the segment will be longer than the distance d to the plane. The distance d is a constant value known based on the opto-mechanical layout.

The intersection p with the plane is derived with the parametric line equation

$$p(m_1, P_1, n_t) = m_1 + s \times (P_1 - m_1), \tag{4.9}$$

where n_t is the normal of the subsequent plane and s is the intersection parameter scalar obtained as

$$s = \frac{n_t \cdot (m_1 - p_t)}{n_t \cdot (P_1 - m_1)}. \tag{4.10}$$

4.4 Mirror rotation dynamics

4.4.1 Step response excitation

A fast steering mirror (FSM) typically consists of a mirror, polished and with coatings such as protected gold, aluminum or silver, [33]. Precision movement of the FSM is done via embedded actuators capable of rotating the mirror closely along its two perpendicular axes that typically intersect on the center of the mirror [44, 32, 34]. Similar

to such a typical design of a FSM, this paper uses the the two inch FSM depicted in Figure 4.5, which can be rotated around x- and y-axes with a voltage driven voice coil motor actuation with an input range of $\pm 10V$.

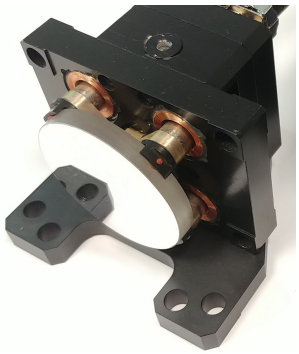


Figure 4.5: FSM OIM202, courtesy of Optics in Motion LLC.

For understanding the rotation dynamics of the FSM, one would typically have to model the dynamics of the FSM actuation system. Such a dynamic model must take into account the rotational inertia of the mirror, vibrations of the mirror/actuation system and possible dynamic coupling of vibrations due to off-centered actuation of the mirror positioning system. However, such dynamic information becomes readily available by performing experiments that reveal the coupled vibrations and static displacement of the mirror/actuation system in the form of step response excitation experiments.

Discrete-time $k = 1, 2, \ldots$ data of mirror angle rotations $\alpha(k)$ and $\beta(k)$ due to the step response excitation of the VCMs of the FSM depicted in Figure 4.5 are displayed in Figure 4.6. During the experiments, independent Voltage step signals (block wave form signals) $V_\alpha(k)$ and $V_\beta(k)$ are applied to each of the VCM of the FSM, whereas multiple step response measurements are used for averaging and additional measurement noise reduction. It is worthwhile to note both the small, but non-negligible static and dynamic coupling between the mirror angle rotations $\alpha(k)$ and $\beta(k)$.

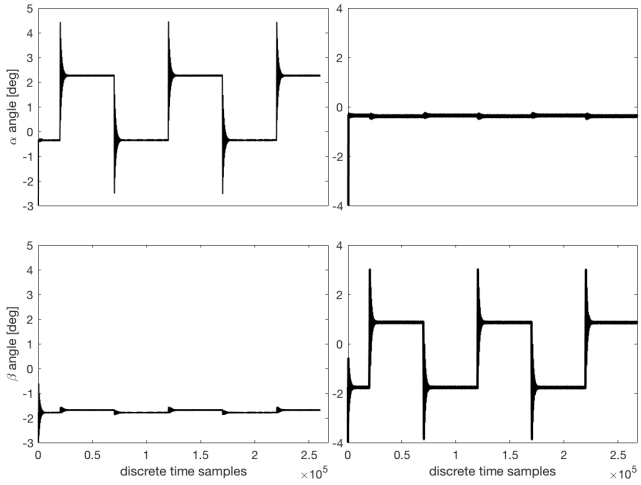


Figure 4.6: Mirror step-response data. Left figures are $\alpha(t)$ and $\beta(t)$ angle rotation response to excitation of voltage $V_\alpha(t)$ of the voice coil motor (VCM) designed for α rotation. Right figures are the same angle rotation response to excitation of voltage $V_\beta(t)$ of the voice coil motor (VCM) designed for β rotation.

4.4.2 Step response realization

The process from step-response data to a linear dynamic model is accomplished with the step-response realization algorithm as presented in [38]. The realization algorithm uses the averaged step response data in Figure 4.6 to formulate a block Hankel matrix from which a fourth order multivariable discrete-time state-space model

$$x(k+1) = Ax(k) + B \begin{bmatrix} V_\alpha(k) \\ V_\beta(k) \end{bmatrix}, \quad \begin{bmatrix} \alpha(k) \\ \beta(k) \end{bmatrix} = Cx(k) \tag{4.11}$$

is derived via a singular value decomposition. More details of the procedure can be found in [38], as for this paper it suffices to conclude that the state-space model in (4.11) is able to model the static and dynamic coupling between the rotation angle mirror angle rotations $\alpha(t)$ and $\beta(t)$. A confirmation of the quality of the dynamic model is given in Figure 4.7 that presents a visual comparison of the measured step response data and the simulated

step response produced by the fourth order state space model in (4.11).

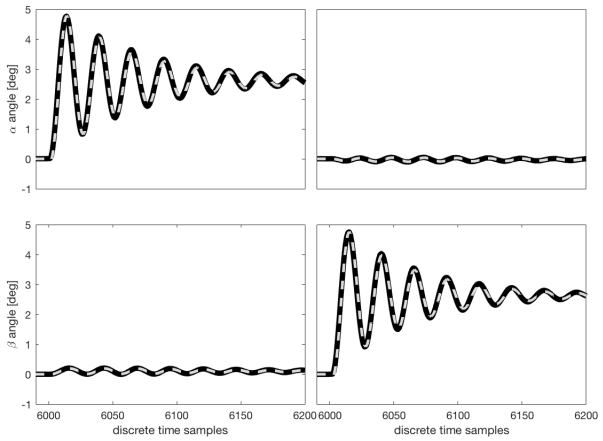


Figure 4.7: Comparison between measured multivariable step response data (solid lines) and simulated multivariable step response data (dashed lines) obtained by fourth order state space model in (4.11) obtained, in turn, by the step response realization algorithm.

4.5 Mirror input shaping

4.5.1 Target parametrization

To illustrate the effectiveness of dynamic input shaping for a FSM, a target trajectory with fast dynamic transitions in the motion of the FSM is chosen. Without loss of generality we may assume a target surface aligned with the (y, z)-plane of the global coordinates and formulate fast dynamic transitions in form of a square trajectory as indicated in Figure 4.8

The trajectory is parametrized by a parameter ϕ by following the sequence from Figure 4.8 according to

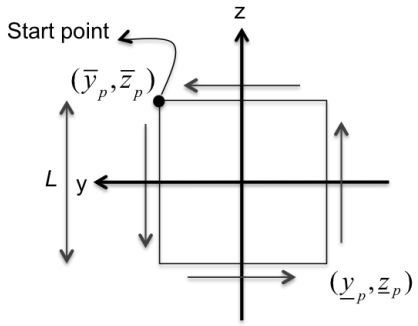


Figure 4.8: Trajectory of the square of size L in target surface aligned in (y, z)-plane.

$$\begin{aligned}
(y_p(\phi), z_p(\phi))_1 &= (\overline{y}_p, \overline{z}_p - \phi) \\
(y_p(\phi), z_p(\phi))_2 &= (\overline{y}_p - \phi, \underline{z}_p) \\
(y_p(\phi), z_p(\phi))_3 &= (\underline{y}_p, \underline{z}_p + \phi) \\
(y_p(\phi), z_p(\phi))_4 &= (\underline{y}_p + \phi, \overline{z}_p) \\
\text{for} \quad 0 \leq \phi \leq L
\end{aligned} \tag{4.12}$$

From the starting point $p = (\overline{y}_p(\phi), \overline{z}_p(\phi))$, we move through the axes by increments of ϕ, until we close the figure in the same starting point.

4.5.2 Inverse kinematics

Given a target trajectory $(y_p(\phi), z_p(\phi))$ parametrized in terms of ϕ by (4.12), the values for the desired mirror rotation pair $(\alpha(\phi), \beta(\phi))$ can be found by inverse kinematic analysis of the mirror rotation. In the inverse kinematic analysis, the mirror normal vector $n(\alpha, \beta)$ dependent reflection matrix $T(n(\alpha, \beta))$ given in (4.7) can be used to compute the desired mirror rotation pair $(\alpha(\phi), \beta(\phi))$.

To start the inverse kinematics process, we normalize the target p via

$$p_1(\phi) = \frac{p}{\|p\|}, \text{ where } p = (y_p(\phi), z_p(\phi))$$

and obtain the inverse direction vector p_1, towards the mirror. Having the direction of the incoming source beam p_0, we can obtain the normal of mirror $n(\phi)$ in the global coordinates as a function of the parameter ϕ via

$$n(\phi) = \frac{p_1(\phi) - p_0}{\|p_1(\phi) - p_0\|} \tag{4.13}$$

and the normal of the mirror in local coordinates given by

$$n_\ell(\alpha(\phi), \beta(\phi)) = R_y^{-1}(45^\circ) n_1(\phi).$$

Finally, the values of desired mirror rotation pair $(\alpha(\phi),$
$\beta(\phi))$ are found by

$$\alpha(\phi) = \tan^{-1}(\frac{-n_\ell^y(\phi)}{n_\ell^z(\phi)}), \quad \beta(\phi) = \tan^{-1}(\frac{n_\ell^x(\phi)}{n_\ell^z(\phi)}) \tag{4.14}$$

by using (4.4), where n_ℓ^x indicates the x-coordinate, n_ℓ^y indicates the y-coordinate and n_ℓ^z indicates the z-coordinate of $n_\ell(\phi)$. Using the square shape as a target, we follow the inverse kinematics process to compute the angles sets $(\alpha(\phi), \beta(\phi))$ shown in Figure 4.9, to reach the target as seen on Figure 4.8.

4.5.3 Quasi-static input shaping

Reaching the desired target with the computed mirror angle pair $(\alpha(\phi(k)), \beta(\phi(k))$ obtained from the inverse kinematic analysis in (4.14) presents a challenge, as the actual FSM is a dynamic system. This challenge becomes apparent when a $(\alpha(\phi(k)), \beta(\phi(k))$ is required over a short time period $k = 1, 2, \ldots, N-1$ of $N = 300$ samples as depicted in Figure 4.9 and leading to the target shape depicted in Figure 4.10. It is clear that the dynamic behavior of the FSM interferes with the quasi-static analysis of the inverse

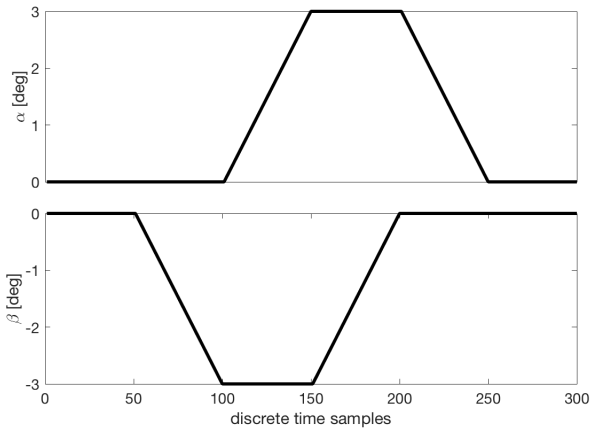


Figure 4.9: Discrete-time mirror angle pair $\alpha(\phi(k))$ (top) and $\beta(\phi(k))$ (bottom) to obtain the desired target square of size L over a short time period of 300 samples.

kinematics.

4.5.4 Dynamic input shaping

To address the coupled static and dynamic motion of the FSM, input shaping is required. For the dynamic input shaping, knowledge of the dynamic model obtained by step response experiments and given by the state space model in (4.11) can be used to constrain the dynamic motion of the FSM. For writing the (linear) motion constraints we follow an approach similar to [28], but impose additional constraints on the desired static and dynamic motion of the FSM. This process is thoroughly detailed in Chapter 3.

The first constraint is to enforce the resulting output $y(k) = \begin{bmatrix}\alpha(k) & \beta(k)\end{bmatrix}^T$ to be close to the desired output $y(\phi(k)) = \begin{bmatrix}\alpha(\phi(k)) & \beta(\phi(k))\end{bmatrix}^T$ obtained from the inverse kinematic analysis in (4.14) via $y(k) \leq y(\phi(k)) + \varepsilon(k)$ and $y(k) \geq y(\phi(k)) - \varepsilon(k)$. Due to the dynamic behavior modeled by the state space model in (4.11) and under the assumption that initial state *x(1)=0*, such a constraint on the output $y(k)$ can be converted to the input $u(k) = \begin{bmatrix}V_\alpha(k) & V_\beta(k)\end{bmatrix}^T$ for input shaping at each time index $k = 1, 2, \ldots, N$ via the

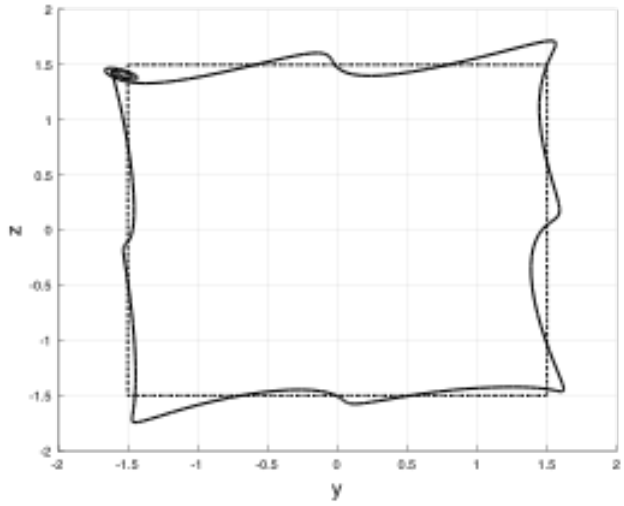


Figure 4.10: Target shape produced by FSM due to VCM actuation scaled by computed quasi-static mirror angle pair $(\alpha(\phi(k)), \beta(\phi(k))$ of Figure 4.9.

matrix relation $\mathbf{y} = \Phi\mathbf{u}$.

The next constraints impose constraints on the volatility and size of the resulting input signal $u(k) = \begin{bmatrix} V_\alpha(k) & V_\beta(k) \end{bmatrix}^T$. To avoid volatility in the voltage/input signals, constraints on the rate of change $|V_\alpha(k)(k) - V_\alpha(k)(k-1)| \leq d(k)$, $|V_\beta(k) - V_\beta(k-1)| \leq d(k)$ are implemented to avoid large values in the voltage/input signals during input shaping. As also described in Chapter 3, the input shaping problem under the linear constraints can be written in a LP problem.

Computation of a feasible solution to the LP problem of 3.16 using the target desired mirror rotation pair $y(\phi(k)) = \begin{bmatrix} \alpha(\phi(k)) & \beta(\phi(k)) \end{bmatrix}^T$ depicted earlier in Figure 4.9 now leads to much better dynamic performance of the FSM in tracking the desired square trajectory. The results are summarized in Figure 4.11 and it can be seen that despite the short period of $N = 300$ samples, the dynamics of the FSM is correctly handled by the input shaping routine.

It is also worthwhile to inspect the computed input shape $u(k) = \begin{bmatrix} V_\alpha(k) & V_\beta(k) \end{bmatrix}^T$ used as input for each of the VCM of the FSM. The input voltage signals are summarized

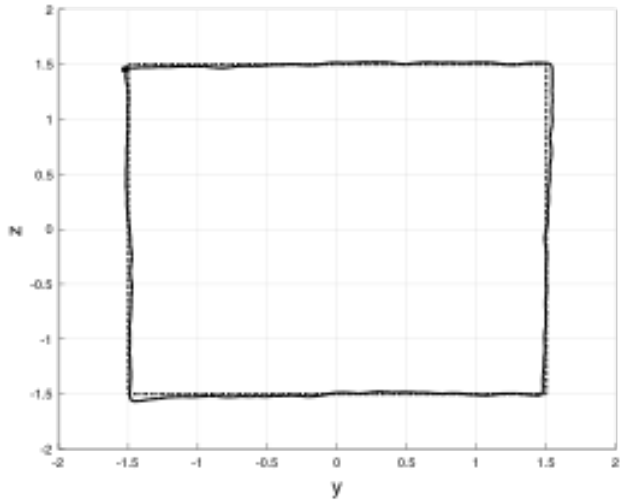


Figure 4.11: Target shape produced by FSM due to VCM actuation via input shaping via Linear Programming problem 3.16.

in Figure 4.12 and it is interesting to observe that the small changes in the input signal anticipate the both the static changes and vibrations of the FSM to reduce tracking errors and vibrations in the beam pointing.

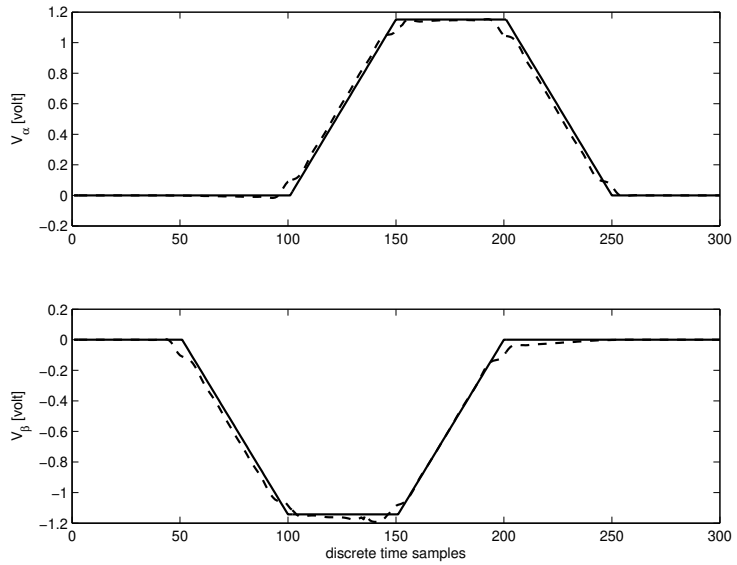


Figure 4.12: Result of input shaping for $V_\alpha(k)$ (top) and $V_\beta(k)$ (bottom) with Comparison between the quasi-static input due to inverse kinematics only (solid lines) and the dynamic shaped input (dashed lines).

4.6 Conclusions and Future Research

Kinematic analysis and dynamic modeling based on step response experiments is shown to be an effective way to model both the quasi-static behavior of beam steering and the dynamic coupling of the motion system of a fast steering mirror (FSM). In case of a single FSM, inverse kinematic analysis can be used to compute desired mirror rotation angles. In addition, it has been shown that step response experiments can be used to formulate a multivariable model that captures both the static and dynamic coupling of a two-axis motion system of a FSM.

The work in this paper shows how desired mirror rotation angles and the multivariable model can be used to formulate an input shaping design that is solved via a standard Linear Programming problem. Experimental results from a commercial FSM are used to validate the approach and a comparison is made between standard quasi-static input and dynamically shaped input to show the effectiveness of the input shaping. Future work

involves adding feed-forward control to the input shaping, as well as creating an optical layout with feedback outside of the FSM, such as a position sensing device to measure and correct any error due to input shaping.

4.7 Acknowledgement

The authors acknowledge and thank Don Henderson from Optics in Motion LLC and Patrick Newton and Thomas Morrison for their support.

This chapter, in part, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Multivariable Dynamic Input Shaping for Two-Axis Fast Steering Mirror." *IFAC-PapersOnLine*, Volume 53, Issue 2, 2020, pp. 8644-8649. The dissertation author is the first author of this article.

Chapter 5

Dynamic Input Shaping and Image Motion Compensation for a Dual-Mirror System

5.1 Chapter Abstract

This paper shows that the task of trajectory planning for a dual-mirror optical pointing system with mehcanical vibrations greatly benefits from carefully designed dynamic input signals, especially when high bandwidth throughput is required. The optical pointing system consists of two Fast Steering Mirrors (FSM) for which dynamically coupled input signals are designed, while adhering to mechanical and input signal constraints. The proposed procedure consists of three steps and starts from dual-input vibration data to formulate a dynamic model and then use the model to formulate an input shaped signal via a convex optimization. First, step-response data is used to estimate a dynamic model of each FSM via a realization algorithm. Second, via (inverse) kinematic analysis the planned trajectories for the dual-mirrors are determined. Third, a linear programming problem is

used to compute the dynamic input signal for each of the FSMs, with one of the mirrors acting as an image motion compensation device, that guarantees tracking of a planned trajectory within a specified accuracy and the operating constraints of the FSMs. The end result is a dynamically coupled set of planned input signals that improve the overall tracking of the dual-mirror optical pointing system, as demonstrated on the FSMs exercise included in this paper.

5.2 Introduction

5.2.1 Motivation and method

The motivation of the study in this paper is based upon a two voice coil motor (VCM) actuated FSM with a flexure suspension, organized in the dual-mirror configuration of Figure B.1.

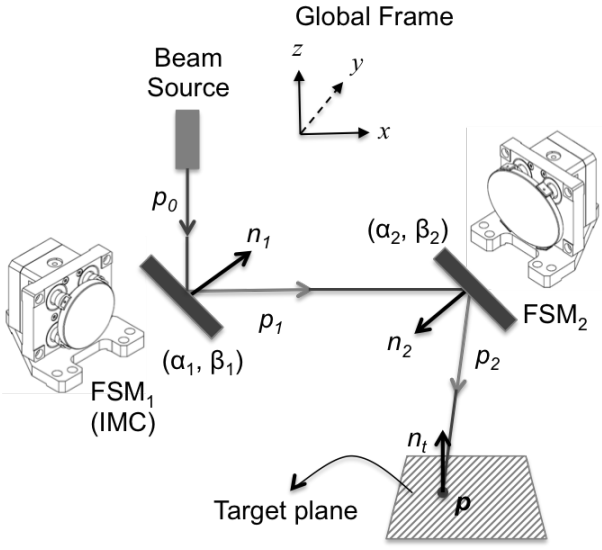


Figure 5.1: Mechanical layout of a dual fast steering mirror system with FSM_1, FSM_2 and a beam source. The orientation of the mirrors is indicated by the normal vectors n_1, n_2 and the direction of the beam path is represented by the vectors p_0, p_1 and p_2.

To illustrate the effectiveness of dynamic input shaping for FSMs, a beam target trajectory problem with fast dynamic transitions in the motion is chosen. The target is in form of a square trajectory, as illustrated in Figure 4.8. The beam source will reflect in FSM_1 towards FSM_2, which will attept to follow the trajectory. FSM_1 then will act as a compensator for the error on positioning from FSM_2 to the target. (α_1, β_1) and (α_2, β_2) are the position in degrees of each of the mirror axes. Each angle is commanded via voltage input $(cmd_{\alpha 1}, cmd_{\beta 1})$ and $(cmd_{\alpha 2}, cmd_{\beta 2})$, respectively.

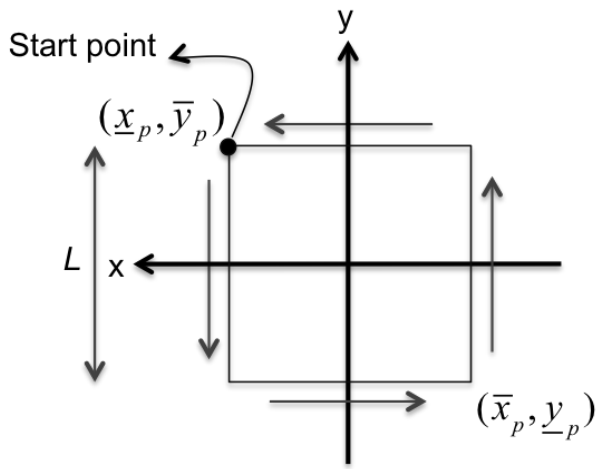


Figure 5.2: Beam desired trajectory: a square of size L in the target plane aligned in (x, y)-plane.

The VCMs in each FSM are mounted in a push/pull configuration to generate torque to rotate the mirror against the flexure suspension. Before formally stating the contributions of this paper, it is worthwhile to illustrate the mechanical vibrations and (dynamic/static) cross coupling between the rotational axes of the FSM, as summarized in the step response measurement of Figure 5.3. It can be seen that application of a step-wise voltage input signal on one of the VCMs in the FSM results not only in mechanical vibrations of the angular rotation of the FSM around one of its main orthogonal axis; it also creates a dynamic coupling in the form of vibrations around the other orthogonal axis. It is clear

that the motion system of a FSM is inherently dynamic and considering only the static kinematic relationships in a dual-mirror system is not enough for accurate precision.

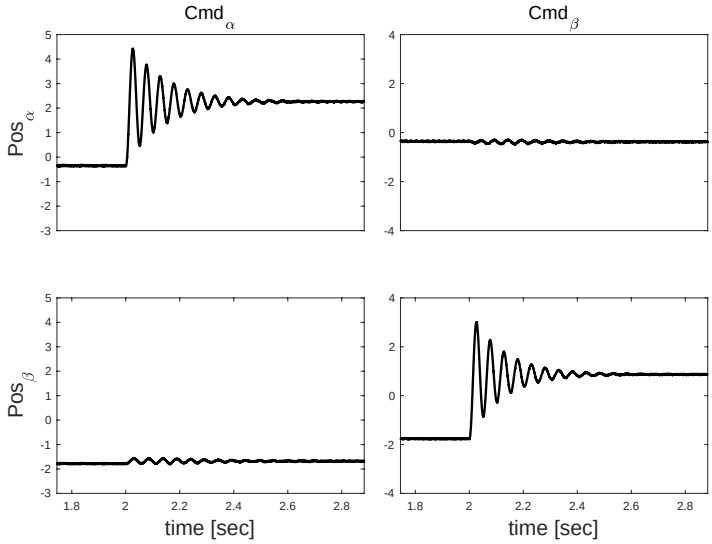


Figure 5.3: Illustration of angular vibrations and cross-couplig in the output, due to step-wise voltage excitation of a VCM actuated FSM with a flexure suspension for each rotational axis.

The contribution of this paper is to use step-response data of FSMs (as illustrated in Figure 5.3) to develop a method for dynamic trajectory planning of a dual-mirror optical pointing system (as illustrated in Figure B.1) that guarantees that optical signals are projected within a pre-specified accuracy on a target plane. The mirror dynamics is not capable of following the trajectory for fast transition of the desired target, Figure 5.4.

A step for this analysis is the kinematics analysis; this refers to the geometrical motion of the mirrors without consideration of the forces (dynamics). The mirror is commanded to move within a range of degrees, which reflects as the final position of the beam in the target plane in $x - y$ coordinates. The desired target is defined as coordinates in the target plane, thus to compute the angles required fo trace the target, the inverse kinematics process shall be performed to obtain the angles for each axis of both mirrors. This process is explained in Section 5.3.

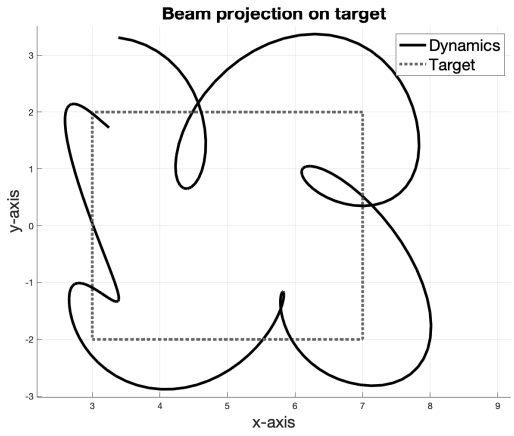


Figure 5.4: Comparison of the target and the dynamics of the system.

The proposed solution to address the task of dual-mirror trajectory planning is a combination of dynamic modeling and multivariable optimal input shaping that takes into account both the dynamics and the kinematic coupling between the FSMs, as well as constraints on the input signal that can be applied to the FSMs. The novelty of the proposed procedure goes from multivariable step-response to feasible multivariable input shape, while working with convex optimization to compute the dynamic input signal that guarantees tracking of a planned trajectory within a specified accuracy and the operating constraints. It is shown how a multivariable step-response data with dynamic coupling can be used for coupled input shaping. In short, the approach outlined in this paper designs a coupled input shaped for a coupled multivariable dynamic system.

The method of data-based IS for a dual-mirror system proposed in this paper consists of three basic steps.

- Step-response data is used to estimate a dynamic model of each FSM via a realization algorithm. This first step is used to directly formulate a linear multivariable discrete-time dynamic model for each FSM that captures the coupled dynamics of the two-axis actuation based on the realization algorithm.

- Via (inverse) kinematic analysis the planned trajectories for the dual-mirrors are determined. The forward kinematics describes the kinematic behavior of the beam reflection as a function of the two-axis rotation of the FSM, extended to the dual-mirrors system.

- The problem of dynamic IS based on the planned trajectories for the dual-mirrors is solved by a linear program (LP) problem extended to the multivariable dynamic model of the FSMs.

The LP problem solves the multivariable IS for each FSM under constraints on size and rate-of-change on the input signal to find a feasible solution that guarantees a certain accuracy for the planned trajectory of each FSM. The LP result for one FSM shows the room for improvement if a second mirror is to be used as an image motion compensation (IMC) device.

5.2.2 Literature overview

The concept of image motion stabilization via dynamic beam control covers an area of multidisciplinary contributions in optics, atmospherics, control theory, vibration analysis and lasers [33]. Examples of beam controls in military and commercial applications can be found in [45, 46] and include surveillance, missile guidance, communications, astronomy and lithography. The optical devices used in beam controls may vary according to the specific application and include fixed mirrors, optical beam-splitter, position sensing detectors, focusing lenses and fast steering mirrors.

A fast steering mirror (FSM) is especially useful in image motion stabilization as it can be used to alter the optical pathway for beam controls [47, 48]. A FSM is typically an electro-mechanical positioning device allowing rotational movement on either one or two orthogonal rotational axes in the plane of the mirror. Multiple commercial FSMs exist

and may include motion systems based on piezoelectric mirrors, galvanometer mirrors and voice coil actuated mirrors [44]. Although useful for optical pathway planning, the motion system of a FSM will exhibit dynamic phenomena that includes mechanical vibrations and (dynamic/static) cross coupling between the rotational axes of the FSM. The resulting challenge of dynamic pathway planning becomes compounded when two FSMs are used in tandem to increase motion range in a so-called dual-mirror configuration [49, 50] as also illustrated in Figure 5.1.

It should be recognized that the notion of input shaping (IS) is not new [1] and there are new tools and techniques to solve the IS problem [7, 8]. The input shaping concept is commonly used to describe the convolution of a reference command with a series of impulses designed to eliminate unwanted vibration, which implies a short delay [1]. In this paper, input shaping is formulated as an open-loop problem, with linear constraints on input and output signals imposed to formulate a linear programming problem to find optimal input profiles. Recent approaches of data-based IS have been developed for mechanical or MEMS devices [39], but often ignore the possible dynamic coupling effects in case of multi-axis actuation as seen in a FSM. Different approaches are used when solving a coupled dynamic problem, from crafting a lower cross-coupling device to designing a controller to minimize the error originated from the cross-coupling [51]. Although a precise decoupled mechanical design of the FSMs provides the opportunity of performing two independent IS problems, realistically a multi-input, multi-output approach is required [47]. In some exercises the system model formulation is based in a physical approach [52], while some use convex optimization approach for multi-input systems [53].

For system identification methods the algorithms in [54, 38] are referred. For forward kinematics analysis, including beam reflection and instersection point, and inverse kinematics [40, 43, 41, 33] are referred. The LP analysis is performed as in [55] with references to [30, 31]. IMC plays an important role in a variety of applications such

as satellites, telescopes and tracking/pointing systems, where the optical configurations deployed require at least two optical devices (i.e. a gimbal and a mirror) to achieve a common goal, such as accurately tracking a target [45, 56].

5.3 Beam-pointing kinematics

5.3.1 Beam reflection

For the kinematic analysis, the rotation axes of a FSM are assumed to cross at a center point m of the mirror. The reflection problem can be formulated as a matrix multiplication $p_1 = T(n)p_0$, where $T(n)$ denotes the reflection matrix and p_0 and p_1 represent unit vectors in the directions of the incident beam and reflected beam, respectively. The reflection matrix $T(n) = \boldsymbol{I_3} - 2nn^T$, where $\boldsymbol{I_3}$ represents an identity matrix of size 3, is a function of the rotation of the normal vector n of the mirror plane [41]. This section summarizes the expression of $T(n)$ as a function of the mirror rotation.

As a general approach, the local mirror frame is defined as the x- and y-axes in the plane of the mirror, and the local normal vector n_ℓ of the mirror plane is aligned with the z-axis. The mirror can be rotated about both its axes independently by the angles α and β, changing the local normal vector $n_\ell(\alpha, \beta)$ of the mirror according to

$$n_\ell(\alpha, \beta) = \begin{bmatrix} \cos\alpha \sin\beta \\ -\cos\beta \sin\alpha \\ \cos\beta \cos\alpha \end{bmatrix} \tag{5.1}$$

In addition, the local normal vector $n_\ell(\alpha, \beta)$ in the local frame of the mirror must be referenced to the global frame. As a result, the rotated normal n of the mirror in global frame depends on the mounting (rotation) angles (ϕ, θ, ψ) of the normal of the mirror

$n_\ell(0,0)$ relative to the global frame. Since the mounting angles (ϕ, θ, ψ) of the mirror can be assumed to be fixed, the associated rotation matrices are constant, creating a global normal vector $n(\alpha, \beta) = R\, n_\ell(\alpha, \beta)$, where R is the product of the rotation matrices $R = R_x(\phi)R_y(\theta)R_z(\psi)$.

Consistent with the dual-mirror configuration displayed in Figure 5.1, the notation $n_1(\alpha_1, \beta_1)$ and $n_2(\alpha_2, \beta_2)$ is adopted to indicate the global normal vectors of the first FSM_1 and the second FSM_2 respectively. Similarly, $n_{\ell,1}(\alpha_1, \beta_1)$ and $n_{\ell,2}(\alpha_2, \beta_2)$ will be used for the local normal vectors. Assuming FSM_1 in Figure 5.1 is a flat mirror, it should be noted that p_0 and $n_1(\alpha_1, \beta_1)$, span the beam plane and the reflected beam p_1 lies in the same beam plane. The resulting reflection matrix $T_1(n(\alpha_1, \beta_1))$ for FSM_1 now depends on α_1 and β_1 and given by $T_1(n_1(\alpha_1, \beta_1)) = I_3 - 2n_1(\alpha_1, \beta_1)n_1^T(\alpha_1, \beta_1)$ in terms of the global normal vector $n_1(\alpha_1, \beta_1)$ of FSM_1.

5.3.2 Beam intersection point

Continuing the kinematic analysis of the dual-mirror system depicted in Figure 5.1, the reflected beam $p_1 = T_1(n_1(\alpha_1, \beta_1))p_0$ creates a location $p \in \mathbb{R}_{3\times 1}$ on a subsequent target plane via FSM_2. This section summarizes the computation of the point p as function of the angular rotations of the two FSMs in the dual-mirror system.

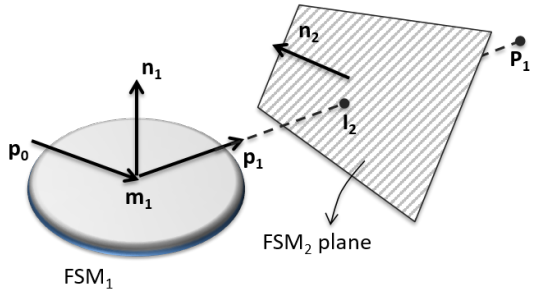


Figure 5.5: Intersection of reflected beam, p_1, with FMS_2 plane at point I_2.

Figure 5.5 illustrates the reflected beam vector p_1, starting at point m_1 is used to

describe a segment $\overline{m_1 P_1}$ with final point $P_1(m_1, p_1) = m_1 + k d_1 p_1^T$, where k is a scaling of the (unit) vector p_1 to assure the segment will be longer than the distance d_1 between FSM_1 and FSM_2. The mirror plane of FSM_2 can be described by a set of points I for which $(I - m_1) \cdot n_2(\alpha_2, \beta_2) = 0$. The resulting intersection point I_2 of the segment $\overline{m_1 P_1}$ and the surface of FSM_2 can be found with the parametric line equation

$$I_2 = m_1 + s \times (P_1 - m_1), \quad s = \frac{n_2(\alpha_2, \beta_2) \cdot (m_1 - m_2)}{n_2(\alpha_2, \beta_2) \cdot (P_1 - m_1)} \tag{5.2}$$

where s is the intersection parameter scalar [43]. Assuming the inner product $(P_1 - m_1) \cdot n_2(\alpha_2, \beta_2) \neq 0$, i.e. the beam vector p_1 is not parallel with the surface of FSM_2.

Following the same derivation, the relation of the reflected beam $p_2 = T_2(n_2(\alpha_2, \beta_2))p_1$ of FSM_2 to the point of intersection p on the target plane with the normal vector n_t is obtained. The final point of segment $\overline{I_2 P_2}$ is $P_2(m_2, p_2) = I_2 + k d_2 p_2^T$, where k is a scaling of the (unit) vector p_2 to assure the segment will be longer than the distance d_2 between FSM_2 and the target plane. Assuming again $(P_2 - I_2) \cdot n_t \neq 0$, e.g. the reflected beam vector p_2 is not parallel with the target plane, the end result is a target point

$$p = I_2 + s \times (P_2 - I_2), \quad s = \frac{n_t \cdot (I_2 - m_t)}{n_t \cdot (P_2 - I_2)} \tag{5.3}$$

where m_t is a (central) point on the target plane, n_t is (unit) normal vector of the target plane and I_2 is the intersection point of FSM_2 given in (5.2). To be complete, $p_2 = T_2(n_2(\alpha_2, \beta_2))p_1$ and $p_1 = T_1(n_1(\alpha_1, \beta_1))p_0$, making the target point p basically a function of the rotations (α_1, β_1) of FSM_1 and the rotations (α_2, β_2) of FSM_2.

5.3.3 Target parametrization

To illustrate the effectiveness of dynamic input shaping for FSMs, a target trajectory with fast dynamic transitions in the motion is chosen. Similar to the kinematic analysis presented, the target plane is assumed to be aligned with the (x, y)-plane of the global coordinates and formulate fast dynamic transitions in form of a square trajectory as indicated in Figure 4.8. The trajectory is parametrized by ϕ by following the square in Figure 4.8 according to

$$\begin{aligned}
(x_p(\phi), y_p(\phi))_1 &= (\underline{x}_p, \overline{y}_p - \phi) \\
(x_p(\phi), y_p(\phi))_2 &= (\underline{x}_p + \phi, \underline{y}_p) \\
(x_p(\phi), y_p(\phi))_3 &= (\overline{x}_p, \underline{y}_p + \phi) \\
(x_p(\phi), y_p(\phi))_4 &= (\overline{x}_p - \phi, \overline{y}_p) \\
\text{for } \ 0 \leq \phi \leq L, \ &\text{where } L = 4 \text{ inches.}
\end{aligned} \tag{5.4}$$

From the starting point $p = (\underline{x}_p(\phi), \overline{y}_p(\phi))$, the trajectory is followed along the different axes of the square by increments of ϕ, ending in the same starting point.

5.3.4 Inverse kinematics

Given the target trajectory $(x_p(\phi), y_p(\phi))$ parametrized by (4.12), the values for the desired mirror rotation pairs (α_1, β_1) and (α_2, β_2) can be computed as a function of ϕ by inverse kinematic analysis of the dual-mirror system. Typically in dual-mirror systems, one mirror is smaller with a reduced angular range and higher bandwidth, and the second mirror is bigger with a wide angular range and smaller bandwidth. In the dual-mirror system of Figure B.1, FSM_2 may act as a larger/coarse tracker while FSM_1 is responsible for the fine tracking of an optical target on the target plane. As such, FSM_1 can be used to compensate for any tracking errors or range limitations due to FSM_2 and FSM_1 acts as

the image motion compensation device. Following this notion of IMC, inverse kinematics is performed by first setting $(\alpha_1, \beta_1) = (0, 0)$ and computing (α_2, β_2). Any errors created by the dynamic motion of FSM_2 will then be compensated via designing a dynamic input for FSM_1.

The inverse kinematics process starts by normalizing the beam reflection vector $p_2(\phi) = p/\|p\|$, where $p = (x_p(\phi), y_p(\phi))$ is parametrized by (4.12). With $(\alpha_1, \beta_1) = (0, 0)$, the beam reflection vector p_1 is known and the global and local normal vector of FSM_2 are computed via

$$\begin{aligned} n_2(\phi) = &\quad \frac{p_2(\phi) - p_1}{\|p_2(\phi) - p_1\|} \\ n_{\ell,2}(\phi) = &\quad R_y^{-1}(-135^\circ) n_2(\phi) \end{aligned} \tag{5.5}$$

Subsequently, the values of desired mirror rotation pair $(\alpha_2(\phi), \beta_2(\phi))$ are found by

$$\alpha_2(\phi) = \tan^{-1}\left(\frac{-n_{\ell,2}^y(\phi)}{n_{\ell,2}^z(\phi)}\right), \quad \beta_2(\phi) = \tan^{-1}\left(\frac{n_{\ell,2}^x(\phi)}{n_{\ell,2}^z(\phi)}\right) \tag{5.6}$$

where $n_{\ell,2}^x$, $n_{\ell,2}^y$ and $n_{\ell,2}^z$ respectively denote the x-, y- and z-coordinates of $n_{\ell,2}(\phi)$ of FSM_2 defined in (5.5).

Figure 5.6 shows the computed angles through the inverse kinematics process to reach the square shape target from Figure 4.8. In this analysis, N is the number of points along the parametrized trajectory, where N is chosen to be $N = 200$.

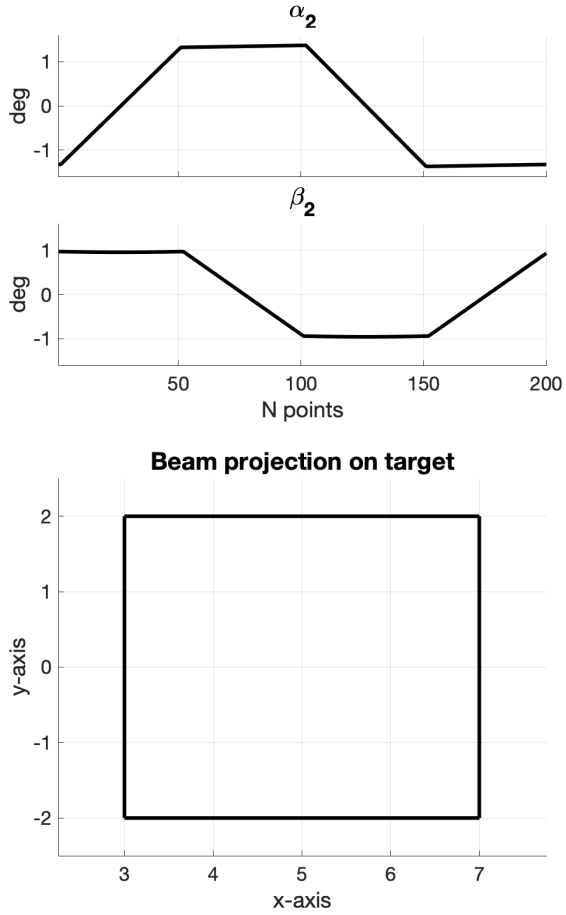


Figure 5.6: Kinematic angles of FSM_2 to obtain the desired target along the square trajectory of size L, with N points and one mirror, FSM_1 is fixed $(\alpha_1, \beta_1) = (0,0)$. Top figure: FSM_2 $(\alpha_2(\phi), \beta_2(\phi)$. Bottom figure: beam projection on the target plane.

5.4 Mirror rotation dynamics

5.4.1 Step-response excitation

Typically a FSM consists of a polished mirror with coatings such as protected gold, aluminum or silver [33]. Precision movement of the FSM is done via embedded actuators capable of rotating the mirror closely along its two perpendicular axes that typically intersect on the center of the mirror [44, 32, 34]. The work in this paper utilizes the two FSMs depicted in Figure 5.7, which can be rotated around x- and y-axes with a voltage

driven voice coil motor (VCM) actuation with an input range of $\pm 10V$.

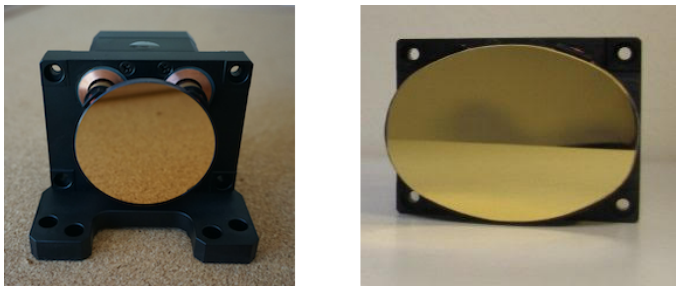


Figure 5.7: OIM202 FSM models, courtesy of Optics in Motion LLC. The mirror on the left is a 2" mirror, OIM202. The mirror on the right is a 2.1" x 3.0" mirror, OIM202.3.

A FSM may be modeled as a double integrator, or as lightly damped second order system [49]. However, such models may only approximate the dynamic behavior and assume a negligible coupling or decoupled model [47, 57]. Furthermore, input shaping is a model-based feedforward compensation and model errors should be reduced as much as possible since it cannot compensate for errors due to model uncertainties [58, 59, 18]. Simple step-response excitation experiments, as illustrated earlier in Figure 5.3, can provide detailed information on dynamic behavior and dynamic coupling of the FSM.

Figure 5.8 and Figure 5.9 show the data of mirror angle rotations due to the step-response excitation of the VCMs of the FSMs. Step experiments are obtained by applying independent voltage step (block wave form signals) $V_\alpha(k)$ and $V_\beta(k)$ to each of the VCMs of the FSMs through the module analog I/O module IO104 from Speedgoat GmbH with Simulink Real-Time™, which measures the angles position as voltage. Multiple step-response measurements are used for averaging and noise reduction. It should be noted that the FSMs from OIM do include a feedback controller to help dampen oscillations but is bypassed to operate in open-loop and to illustrate IS for such a lightly damped mechanical system.

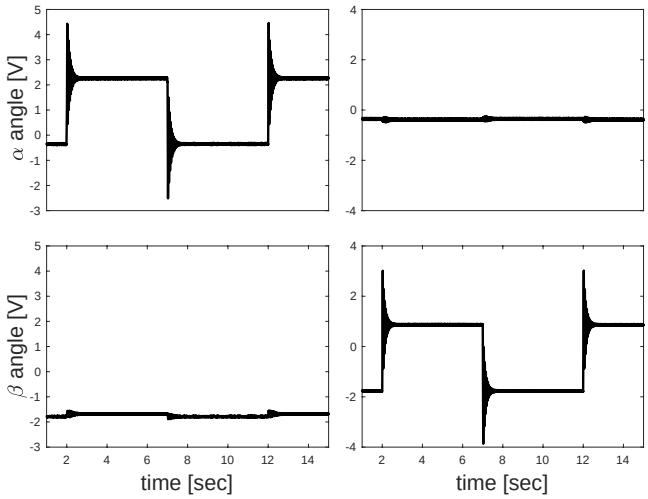


Figure 5.8: OIM202, FSM_1, step-response data. Left figures depict $\alpha(t)$ and $\beta(t)$ angle rotation response measured in voltage to excitation of voltage $V_\alpha(t)$ of the VCM_α rotation. Right figures depict the same angle rotation response to excitation of voltage $V_\beta(t)$ of the VCM_β rotation.

5.4.2 Step-response realization

The process of estimating a multivariable discrete-time model from step-response data is accomplished with a step-response realization algorithm [38, 54]. In particular for a FSM that exhibits (dynamic) coupling between the mirror axes rotation, a multivariable model can be used to model this coupling. For each FSM, the realization algorithm produces a fourth order multivariable discrete-time state-space model

$$x(k+1) = Ax(k) + B \begin{bmatrix} V_\alpha(k) \\ V_\beta(k) \end{bmatrix}, \quad \begin{bmatrix} \alpha(k) \\ \beta(k) \end{bmatrix} = Cx(k). \tag{5.7}$$

where the state matrices A, B and C will be used for model-based IS for each FSM.

Figure 5.10 and Figure 5.11 present a visual comparison of the measured step-response data and the simulated step-response produced by the fourth order state space model for each FSM. It can be seen that the state-space model in (5.7) is able to model

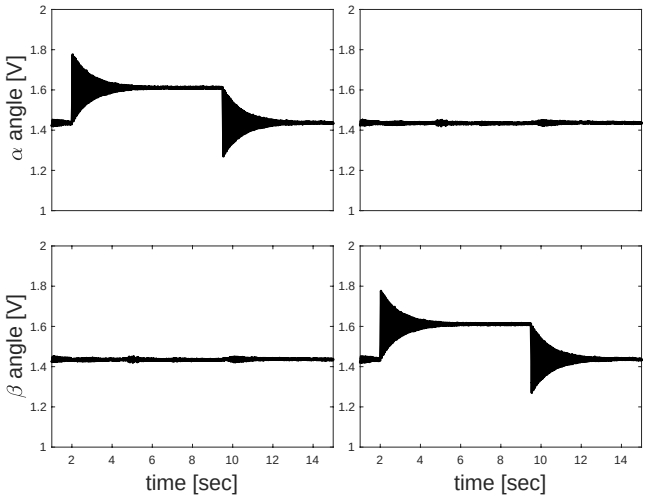


Figure 5.9: OIM202.3, FSM_2, step-response data. Left figures depict $\alpha(t)$ and $\beta(t)$ angle rotation response measured in voltage to excitation of voltage $V_\alpha(t)$ of the VCM_α rotation. Right figures depict the same angle rotation response to excitation of voltage $V_\beta(t)$ of the VCM_β rotation.

the static and dynamic coupling between the mirror angle rotations $\alpha(t)$ and $\beta(t)$.

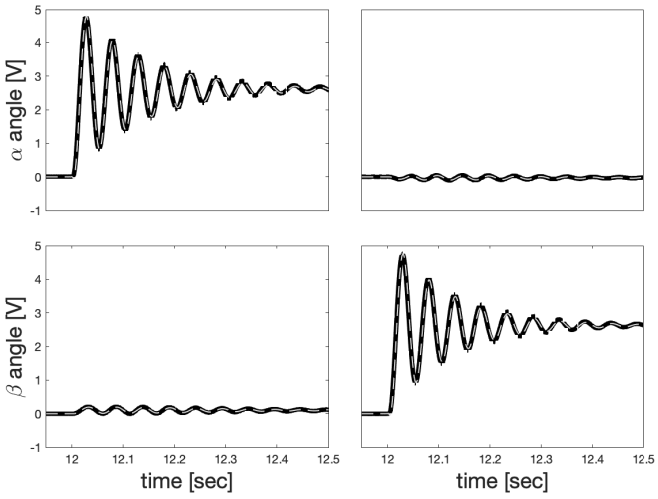


Figure 5.10: Comparison between measured step-response data (solid lines) and simulated step-response data (dashed lines) obtained by fourth order state space model in (5.7) for FSM_1.

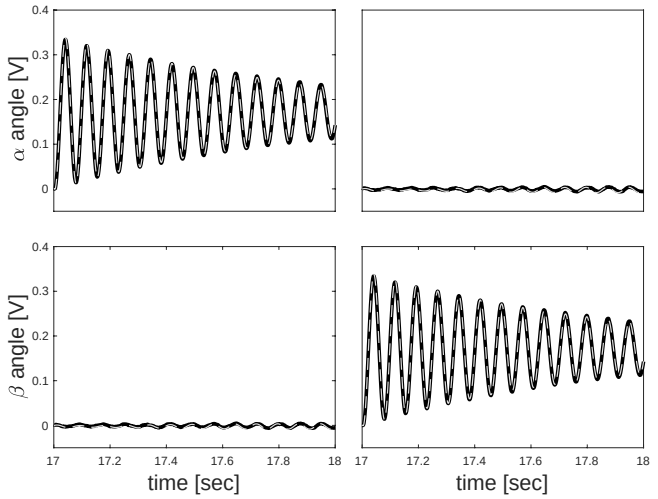


Figure 5.11: Comparison between measured step-response data (solid lines) and simulated step-response data (dashed lines) obtained by fourth order state space model in (5.7) for FSM_2.

5.5 Mirror input shaping

5.5.1 Application of quasi-static input

The angle pair $(\alpha_2(\phi), \beta_2(\phi))$ computed in (5.6) and depicted in Figure 5.6 are made discrete-time dependent angles $(\alpha_2(k), \beta_2(k))$ via $\phi = \phi(k)$ over a time interval $t_k = k\Delta_t,\ k = 1, 2, \ldots, N$. Tracking such a time-dependent angle pair $(\alpha_2(k), \beta_2(k))$ presents a challenge for the flexible FSM_2, especially when Δ_t is small and the time interval $N\Delta_t$ is short. With $\Delta_t = 2$ msec, $N\Delta_t = 400$ msec and voltage signals are applied to FSM_2 that are simply scaled proportionally to the quasi-static values of $(\alpha_2(k), \beta_2(k))$, this challenge becomes apparent and is analiyzed in [60].

5.5.2 Dynamic input shaping

For dynamic input shaping, knowledge of dynamics in the form of the state space model in (5.7) obtained by the step-response experiments and the realization algorithm, can be used to constrain the dynamic motion of FSM_2. For writing the (linear) motion constraints, an approach similar to the one in [28] is followed to formulate the input design as a LP problem [20, 29, 30, 31] as described in Chapter 3.

The first constraint is to enforce the resulting output $y(k) = \begin{bmatrix} \alpha(k) & \beta(k) \end{bmatrix}^T$ to be close to the desired output $y(\phi) = \begin{bmatrix} \alpha(\phi) & \beta(\phi) \end{bmatrix}^T$ in (5.6) via the requirement $|y(k) - y(\phi)| \leq \varepsilon(k)$, $k = 1, 2 \ldots, N$. Such a constraint on the output $y(k)$ can be converted to the input $u(k) = \begin{bmatrix} V_\alpha(k) & V_\beta(k) \end{bmatrix}^T$ for input shaping via the matrix relation $\mathbf{y} = \Phi\mathbf{u}$. The second constraint is a limit on the rate-of-change $|u(k) - u(k-1)| \leq d(k)\Delta_t$, to reduce volatility in the voltage input signals $u(k)$. Such a rate-of-change constraint is implemented via $E\mathbf{u} \leq \mathbf{d}$ as described in Chapter 3. The last constraint limits the size of the input signal via $\underline{\mathbf{u}} \leq \mathbf{u} \leq \underline{\mathbf{u}}$, to avoid large values during input shaping.

The feasibility of a multivariable input signal $\mathbf{u} = \begin{bmatrix} u(1) & u(2) & \cdots & u(N) \end{bmatrix}^T$ that satisfies these constraints can be written as a LP problem. The LP problem proposed in 3.16 is solved with the simplex method, which is very common for linear objective functions.

Finding a feasible solution to the LP problem 3.16 using the desired angle pair $(\alpha_2(k), \beta_2(k))$ computed in (5.6) now leads to the dynamic response of FSM_2 summarized in Figure 5.12. It can be observed that the dynamics of FSM_2 is correctly handled by the input shaping, despite the short time interval of $N\Delta_t = 400$ msec. The offset of the angles $(\alpha_2(k), \beta_2(k))$ leads to a small offset on the target.

It is worthwhile to notice the result obtained in Figure 5.12 is following the mechanical constraints of the FSM_2. The LP problem offers the advantage to compute IS, while allowing the adjust to several constraints. The constraints are given by the mechanical specifications of the FSM and the desired result for a specific application; the rate of change $\mathbf{d}$ and the size of the input signal $\bar{\mathbf{u}}$ and $\underline{\mathbf{u}}$ are attached to the mechanical specs of the mirror, while the margin of error ε is defined by the desired response of the application.

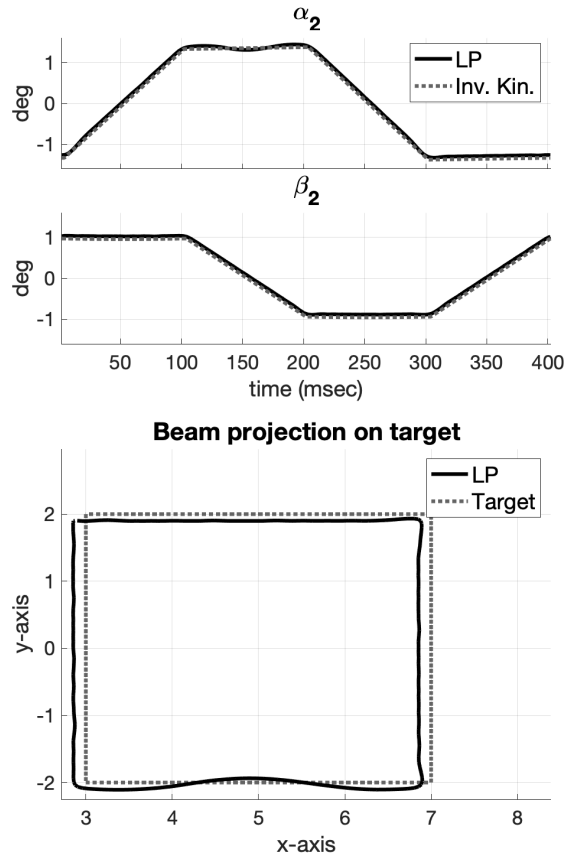


Figure 5.12: Dynamic response of FSM_2 with dynamic input shaping. Top figure: comparison of discrete-time dynamic angle pair $\alpha_2(k), \beta_2(k)$ (solid lines) and desired inverse kinematic angle pair $\alpha_2(\phi), \beta_2(\phi)$ (dashed lines). Bottom figure: comparison of beam projection on the target plane when FSM_1 is fixed at $(\alpha_1, \beta_1) = (0,0)$ for same $(\alpha_2(k), \beta_2(k))$ (solid line) and kinematic angles $(\alpha_2(\phi), \beta_2(\phi))$ (dashed line).

5.6 Image motion compensation

Although input shaping for FSM_2 greatly improves angle tracking, small errors remain on the target plane due to the limitations on the applicable accuracy ε and the voltage input constraints in the LP problem of (3.16).

A basic IMC concept illustrated in [56], consists of a *fine* device working with a *coarse* device. FSM_2 is considered the coarse device, which generates a response with a residual error. The fine device is considered the IMC device, represented by FSM_1 in this

exercise. The residual FSM_2 error signal is used to define the motion of the IMC device. A redundant device such as FSM_1 augments the performance of FSM_2 by reducing the error. The target trajectory defined in Figure 5.2 by (5.4) is not fully reached by FSM_2. The definition of a new trajectory for FSM_1, $(x_1(\phi), y_1(\phi))$, is the difference of the desired target and the obtained result. Figure 5.13 illustrates the definition of the target for FSM_1 as the residual FSM_2 error signal.

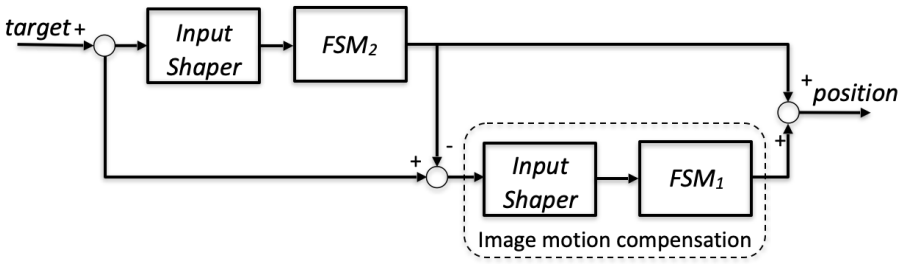


Figure 5.13: Illustration of subsequent input shaping for low bandwidth FSM_2 and high bandwidth FSM_1.

Following the angles computation process from Section 5.3.4, the new angles (α_1, β_1) to compensate for the error can be estimated. Computation of the desired discrete-time varying angles $\alpha_1(k), \beta_1(k)$ can be done via inverse kinematics and followed again by a computation of the input signal $\mathbf{u} = \begin{bmatrix} u(1) & u(2) & \cdots & u(N) \end{bmatrix}^T$ for the multivariable input $u(k) = \begin{bmatrix} V_\alpha(k) & V_\beta(k) \end{bmatrix}^T$ of FSM_1. These computations lead to the dynamic response of the combination of FSM_1 and FSM_2 summarized in Figure 5.14.

The IMC implemented by FSM_1, following the limitations of the position accuracy of FSM_2, is capable of a fast change in the direction of the beam due to its high angular resolution and bandwidth. Keeping FSM_1 initially centered, e.g. $(\alpha_1, \beta_1) = (0, 0)$ and computing the position angles of FSM_2 first, provides FSM_1 with the capability to rotate to its full capacity on its two axis for error correcting purposes. The control process can be summarized as follows:

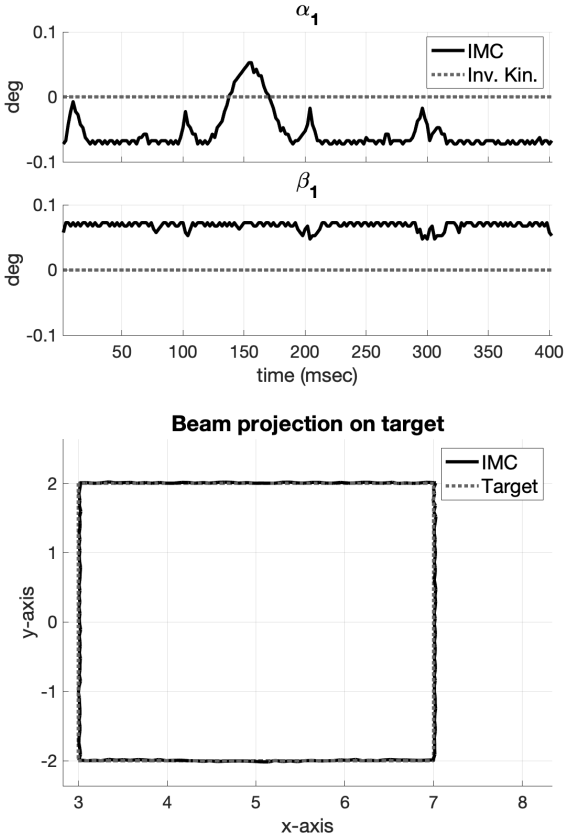


Figure 5.14: Combined dynamic response of FSM_2 with dynamic input shaping and FSM_1 with dynamic input shaping for target error correction. Top figure: comparison of discrete-time dynamic angle pair $\alpha_1(k), \beta_1(k)$ (solid lines) and initial zero angle pair $(\alpha_1, \beta_2) = (0, 0)$ (dashed lines). Bottom figure: comparison of beam projection on the target plane with FSM_1 moving with $(\alpha_1(k), \beta(k))$ and FSM_2 moving with $(\alpha_2(k), \beta_2(k)$ (solid line) and kinematic angles $(\alpha_2(\phi), \beta_2(\phi)$ (dashed line).

1. Define the beam projection target and compute the angles via inverse kinematics;

2. given the desired angles, perform input shaping;

3. apply the input shaped to one mirror (FSM_2) and compute the residual error, which is the target for the IMC device (FSM_1);

4. compute the angles for the IMC device (FSM_1) via inverse kinematics;

5. compute the input shaped for the IMC.

At the end of this process, the shaped inputs for both mirrors are obtained. In practice, the result position is measured with a laser beam profiler (e.g. camera), then the error between the desired target and the result position is computed.

The improvement in tracking on the target plane is summarized in the measured Root Mean Square (RMS) values in Table 5.1. The FSM_2 and IMC input shaping result in a $10\times$ improvement over the FSM_2 input shaping only. A significant improvement is achieved compared to the application of a quasi-static input obtained from inverse kinematic analysis alone.

Table 5.1: RMS of tracking errors for different approaches.

RMS error (in)	x-axis	y-axis
Quasi-static input	0.5361	0.7419
FSM_2 Input Shaping	0.1410	0.0904
FSM_2 and IMC Input Shaping	0.0162	0.0095

5.7 Conclusions

Tracking performance of a target in a dual-mirror system is significantly improved by an open-loop approach in which kinematic analysis and modeling of dynamics is combined with dynamic input shaping. Step-response experiments combined with a realization algorithm provide a linear discrete-time state space model that provides information on motion dynamics and motion coupling of a fast steering mirror (FSM). Using (inverse) kinematic analysis, the desired rotation angles of the main FSM can be computed while the state-space model can be used to formulate the dynamic input signal design via a feasibility check of a linear programming problem. Any remaining target tracking errors can be picked up by the second FSM acting as an image motion compensation device by again input shaping based on FSM dynamics and desired rotation angles.

The end result is a ten-fold reduction of the RMS of the tracking error, while

maintaining constraints on rate-of-change and size of the voltage signals applied to the FSMs in the dual-mirror system. The approach is illustrated here only for the open-loop steering of the dual-mirror system to illustrate the benefit of input shaping of a highly resonant and coupled mechanical motion system. By its nature, the model-based feedforward approach cannot compensate for errors due to model uncertainties, therefore, future work will include a closed-loop based reference signal shaping in which a feedback controller can be used to reduce the effect of model uncertainty.

5.8 Acknowledgement

This chapter, in part, has been submitted for publication, A. Dautt-Silva and R. A. de Callafon (2018) "Reference signal shaping for closed-loop systems with causality constraints." The dissertation author is the first author of this article.

Chapter 6

Reference signal shaping for closed-loop systems with causality constraints

6.1 Chapter Abstract

A reference signal shaping problem formulated as a convex optimization problem is presented for the design of the reference signal in a closed-loop discrete-time linear-time-invariant system, with the purpose that internal control signals and system output are bounded within constraints. A causal solution endures the reference profiles changes only after a system output is required to change. The proposed solution allows us to compute a causal or noncausal reference profile, by adding a time-dependent signal constraint. Feasibility and existence of a reference profile is verified with a linear programming (LP) problem, while an optimal reference profile for the closed-loop system is obtained via a quadratic program (QP) problem. A mass-spring-damper system paired with a PID controller is the illustrative example for closed-loop reference shaping. To evaluate the

proposed design, the closed-system response for both causal and noncausal reference profiles are compared.

6.2 Introduction

In linear time-invariant (LTI) control systems, input shaping (IS) is a technique originally used for defining a shaped command input to eliminate or reduce unwanted system vibration. The input profile brings the system from an initial point to a desired target without the unwanted vibration [1]. Input shaping is usually formulated as an open-loop problem, which can be reformulated as a convex optimization problem by imposing linear constraints on the input and output signals [20, 29, 61]. Commonly, input shaping is used in open-loop for multiple-input multiple-output (MIMO) systems in continuous time and discrete time [2, 3, 4]. With the input shaping schemes being inherently open-loop, uncertainties in the open-loop model can lead to system performance degradation.

Feedback control and the resulting closed-loop system compensate for the uncertainties and disturbances [62, 17]. As such, input shaping can be formulated for the reference signal in closed-loop MIMO discrete-time systems subject to linear constraints on amplitude and rate-of-change as followed in [26, 63]. Some research has suggested placing the input shaper as part of the closed-loop system [64, 65], while a different approach for closed-loop control is extending the system by placing the input shaper outside of the loop [66].

Approaching the IS problem as a general convex optimization problem, the desired response can be further optimized by defining constraint functions and limits or bounds for these constraints [20, 29, 67]. All constraints are combined in a linear matrix inequality (LMI), which can be solved as either a linear program (LP) problem or as a quadratic program (QP) problem. If a solution is found via LP, which follows the constraints, this might not be a desirable input profile if it results in an aggressive motion [23]. Furthermore,

an LP input profile might not be unique and if very different sets of input profiles are obtained by slightly adjusting the constraints, choosing a solution might be challenging [68, 69]. As an example, considering the variance of the rate-of-change in the obtained input profile provides additional uniqueness and an optimal value; the objective function is then redefined as quadratic and we can get an optimal solution via QP.

The problem of computing a reference trajectory that satisfies certain constraints on output and input signals, formulated as an LP or QP problem, has been studied [63, 70]. Unfortunately the results for LP and QP usually lead to a noncausal shaped input obtained from the desired output and its bounds. This might not be the ideal case in many applications, where the use of feedback controllers that do not explicitly require future data to be known is a requirement or and advantage (i.e., causal controllers) [71]. We consider the computation of both causal and noncausal reference signals. The term *causal* is used to refer to shaped signals (input or reference) which are allowed to change only after the output actually requires it. Likewise, in reference shaping, the reference signal should change at the time when the output is required to change, not before.

In the classical approach as open-loop input shaping, the input shaping technique proves its accuracy in an open-loop system, but by its nature it cannot compensate for errors due to model uncertainty or disturbance signals. Input shaping techniques can be combined with any type of feedback controller [65] and show significant improvement. In this paper we focus on the design of a reference signal for a closed-loop system and design an out-of-the-loop reference shaper, which will be referred to as a *closed-loop reference shaping* (CLRS) problem. For the design of the reference input we considered time-variant amplitude and rate-of-change bounds of the closed-loop signal. The CLRS is first verified for feasibility as an LP problem and then restructured as a QP problem to obtain a smoother input profile. Furthermore, this paper presents the possibility of causality, a characteristic usually neglected. These constraints are time-dependent to allow both causal and noncausal

reference profiles to be computed.

The paper is structured as follows. Section 6.3 is the definition of the open and closed-loop systems. Section 6.4 describes the definition of input and output constraints for the system. Section 6.5 discusses the feasibility of the problem by restating the linear constraints from section 6.4 in terms of linear matrix inequalities, which are used to define the convex optimization problem [72]. Feasibility is verified with an LP problem and the input profile is optimized via a QP problem. In section 6.6 we consider a mass-spring-damper system in closed-loop with an existing PID controller (with gains initially defined) as a closed-loop reference shaping problem. The time-dependent constraint added to the reference shaper enables the causality results. Both the causal and noncausal reference profiles applied to the closed-loop system are compared.

6.3 Defining the System: Open-loop and closed-loop models

Given the fact that the continuous-time system will be excited with a Zero-Order-Hold signal updated at regular discrete-time intervals, motivates the use of a ZOH discrete-time equivalent model representation of the continuous-time system. Consider the nth order state space representation of a discrete-time model $G(q)$ with p inputs and m outputs given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{6.1}$$

sampled with a sampling time of Δ_T seconds and where $u(k) \in \mathbb{R}_{p\times 1}$, $y(k) \in \mathbb{R}_{m\times 1}$, $x(k) \in \mathbb{R}_{n\times 1}$. Without loss of generality, zero initial conditions $x(0) = 0_{n\times 1}$ and $u(k) = 0_{p\times 1}$ for $k < 0$ are assumed. Designing a desired input $u(k) \in \mathbb{R}_{p\times 1}$ for (6.1) is considered an open-loop input design and coined as an *input shaping problem*.

To address robustness of input shaping and, in particular, robustness to possible variability in the dynamics or DC-gain of of the model $G(q)$, input shaping will be done for a closed-loop system. The presence of an additional feedback controller can help reduce the effects of variability in the dynamics of $G(q)$, especially in the case of DC-gain variations that would require different steady-state value of the input $u(k)$. For a closed-loop reference design, consider an additional n_cth order negative feedback discrete-time controller $C(q)$ with m inputs, p outputs and a sampling time of Δ_T seconds used to compute the input $u(k)$ via a reference signal $r(k)$ according to

$$u(k) = C(q)[r(k) - y(k)] \ .$$

As a result, the closed-loop system

$$\begin{bmatrix} u(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} C(q) \\ C(q)G(q) \end{bmatrix} (I_{p\times p} + C(q)G(q))^{-1} r(k) \tag{6.2}$$

with the feedback interconnection well posed, will have a state space model

$$\begin{aligned} z(k+1) &= A_{cl} z(k) + B_{cl} r(k) \\ \begin{bmatrix} u(k) \\ y(k) \end{bmatrix} &= \begin{bmatrix} C_u \\ C_y \end{bmatrix} z(k) + \begin{bmatrix} D_u \\ D_y \end{bmatrix} r(k) \ , \end{aligned} \tag{6.3}$$

where $z(k) \in \mathbb{R}_{n+n_c\times 1}$ contains both the model and the controller state and $r(k) \in \mathbb{R}_{m\times 1}$. For the purpose of the analysis presented here, the closed-loop output matrix C_{cl} and feed-through matrix D_{cl} are split in the contributions that produce the control signal $u(k)$ and the output signal $y(k)$. Again, without loss of generality, zero initial conditions $z(0) = 0_{n\times 1}$ and $r(k) = 0_{p\times 1}$ for $k < 0$ are assumed. Designing a desired reference $r(k) \in \mathbb{R}_{pm\times 1}$ for (6.3) will be considered a closed-loop reference design referred as a *closed-loop reference*

shaping problem.

6.4 Linear input and output constraints

When designing a multi-variable input $u(k)$ to be applied to the open-loop system (6.1) or a multi-variable reference $r(k)$ to be applied to the closed-loop system, restriction will be imposed on the resulting internal control signal (input) $u(k)$ and system output $y(k)$. These restrictions are:

- The p inputs in $u(k) \in \mathbb{R}_{p\times 1}$ are bounded between distinct time-varying lower amplitude bounds $\underline{u}(k) \in \mathbb{R}_{p\times 1}$ and time-varying upper amplitude bounds $\bar{u}(k) \in \mathbb{R}_{p\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{gathered} \underline{u}(k) \leq u(k) \leq \bar{u}(k),\ k = 0, 1, \ldots, N-1 \\ \text{where}\ \ \underline{u}, \bar{u} \in \mathbb{R}_{p\times 1} \end{gathered} \tag{6.4}$$

- The p inputs in $u(k) \in \mathbb{R}_{p\times 1}$ sampled at Δ_T seconds are rate limited between distinct time-varying lower rate bounds $\underline{d} \in \mathbb{R}_{p\times 1}$ and time-varying upper rate bounds $\bar{d} \in \mathbb{R}_{p\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{gathered} \underline{d}(k) \leq \tfrac{u(k)-u(k-1)}{\Delta_T} \leq \bar{d}(k),\ k = 0, 1, \ldots, N-1 \\ \text{where } \underline{d}, \bar{d} \in \mathbb{R}_{p\times 1} \text{ and } u(-1) = 0 \end{gathered} \tag{6.5}$$

- The m outputs in $y(k) \in \mathbb{R}_{p\times 1}$ are bounded between distinct time-varying lower perturbation bounds $\underline{\varepsilon}(k) \in \mathbb{R}_{m\times 1}$ and time-varying upper perturbation bounds $\bar{\varepsilon}(k) \in \mathbb{R}_{m\times 1}$ around a time-varying desired output $y_{des}(k) \in \mathbb{R}_{m\times 1}$ for each value of

$k = 0, 1, \ldots, N-1$, e.g.

$$\begin{gathered} \underline{\varepsilon}(k) \leq y(k) - y_{des}(k) \leq \bar{\varepsilon}(k), k = 0, 1, \ldots, N-1 \\ \text{where } \ \underline{\varepsilon}(k), \bar{\varepsilon}(k) \in \mathbb{R}_{m\times 1} \end{gathered} \tag{6.6}$$

- The m references in $r(k) \in \mathbb{R}_{m\times 1}$ are bounded between distinct time-varying lower amplitude bounds $\underline{r}(k) \in \mathbb{R}_{m\times 1}$ and time-varying upper amplitude bounds $\bar{r}(k) \in \mathbb{R}_{r\times 1}$ for each value of $k = 0, 1, \ldots, N-1$, e.g.

$$\begin{aligned} &\underline{r}(k) \leq r(k) \leq \bar{r}(k), \quad k = 0, 1, \ldots, N-1, \\ &\text{where } \ \underline{r}(k), \bar{r}(k) \in \mathbb{R}_{m\times 1}. \end{aligned} \tag{6.7}$$

In the above constraints, N is the number of samples, for which the bounds and rates are defined. The first two conditions on the input $u(k)$ are needed to satisfy basic amplitude- and rate-limit constraints. The condition on the output $y(k)$ defines a performance condition in terms of desired output profile $y_{des}(k)$ and allowable perturbations around it. Rate limits on the output can also be easily included, but not included here as rate limits on the input $u(k)$ and the dynamics of $G(q)$ already impose rate limits. Fixed amplitude- and rate-limit constraints on the input $u(k)$ are typical for actuation of a dynamic system (with or without control) to avoid excessive values on the input in terms of size and rate-of-change. The last condition imposed time-variant amplitude bounds on the reference signal to allow both causal and noncausal solution for the reference signal shaping. However such fixed constraints will allow $u(k)$ to change before $y_{des}(k)$ changes for planning purposes. Requiring $u(k)$ to change only after $y_{des}(k)$ changes, e.g. $\bar{u}(k) = \underline{u}(k) = 0$ before $y_{des}(k)$ changes, can be used to enforce causal solutions. Such causal solutions are therefore easily found by using time-varying amplitude- and rate-limit constraints on the input $u(k)$, to enforce $u(k)$ to change only after $y_{des}(k)$ changes. Both solutions will be illustrated here.

6.5 Closed-loop reference design

6.5.1 Feasibility

Feasibility for closed-loop reference design is defined as the problem of finding feasible $m \times N$ values of the finite-time reference signal

$$r(k) \in \mathbb{R}_{m\times 1}, \quad k = 0, 1, \ldots, N-1 \tag{6.8}$$

for the closed-loop model given in (6.3), such that the linear conditions (6.4), (6.5) hold for the resulting reference profile signal $r(k)$, while also satisfying the linear conditions (6.6) for the resulting output $y(k)$ of the closed-loop system.

Feasibility of these conditions will depend on the closed-loop dynamics created by the feedback of $G(q)$ and the controller $C(q)$, but also also on the specified desired output $y_{des}(k)$ and its bounds $\underline{\varepsilon}(k)$ and $\bar{\varepsilon}(k)$. The fixed amplitude bounds $\underline{u}$, $\bar{u}$ and the fixed rate bounds $\underline{d}$, $\bar{d}$ imposed on the input $u(k)$ are now a combination of the design of the reference $r(k)$ and the control signals generated by the controller $C(q)$. Any point $\hat{x}$ is said to be feasible if it satisfies the objective and all the constraints. The problem is said to be feasible if there exists at least one feasible point [20, 29]. The set of all feasible points is called the feasible region, which implies the obtained solution is not unique. The choice of the constraints as well as the method used for optimization will reflect on the solutions obtained. Feasibility is verified by rewriting the linear conditions in (6.4), (6.5) and (6.6) into a linear matrix inequality and find a solution via a linear programming problem.

For vector notation purposes, the values of $u(k) \in \mathbb{R}_{p\times 1},\ k = 0, 1, \ldots, N-1$ contained in the matrix

$$U = \begin{bmatrix} u(0) & u(1) & \cdots & u(N-1) \end{bmatrix} \in \mathbb{R}_{p\times N}$$

is first written as

$$\mathbf{u} = \text{vec}\{U\} = \left[\begin{array}{cccc} u(0)^T & u(1)^T & \cdots & u(N-1)^T \end{array} \right]^T \in \mathbb{R}_{pN\times 1} \tag{6.9}$$

to obtain a vector $\mathbf{u}$ of desired multi-variable input values. In addition, consider the notation I_n to denote a square identity matrix of size $n \times n$, 1_n to denote a $n \times 1$ vector with values 1 and $0_{n\times m}$ to denote a zero matrix of size $n \times m$, along with the notion of the Kronecker product $\otimes$.

For the condition (6.6) on the values of output $y(k)$ contained in

$$Y = \left[\begin{array}{cccc} y(0) & y(1) & \cdots & y(N-1) \end{array} \right] \in \mathbb{R}_{m\times N}$$

the same vector operation is performed

$$\mathbf{y} = \text{vec}\{Y\} = \left[\begin{array}{cccc} y(0)^T & y(1)^T & \cdots & y(N-1)^T \end{array} \right]^T \in \mathbb{R}_{mN\times 1} \tag{6.10}$$

to obtain a vector $\mathbf{y}$ of desired multi-variable output values.

Similar to the vectorization of $u(k)$ to $\mathbf{u} \in \mathbb{R}_{pN\times 1}$ in (6.9) and $y(k)$ to $\mathbf{y} \in \mathbb{R}_{mN\times 1}$ in (6.10), the values of the multi-variable reference $r(k) \in \mathbb{R}_{m\times 1}, \quad k = 0, 1, \ldots, N-1$ are vectorized via

$$\mathbf{r} = \left[\begin{array}{cccc} r(0)^T & r(1)^T & \cdots & r(N-1)^T \end{array} \right]^T \in \mathbb{R}_{mN\times 1} \tag{6.11}$$

to obtain a vector $\mathbf{r}$ of desired multi-variable reference values. To implement the constraints on the resulting control signal $u(k)$ and resulting model output $y(k)$, first the mapping from $\mathbf{r}$ to $\mathbf{u}$ and $\mathbf{y}$ must be established. The linear mapping is done with the linear state

space description of the closed-loop model in (6.3) via the convolution sums

$$\begin{aligned} u(k) &= D_u r(k) + \sum_{m=1}^{k} C_u A_{cl}^{m-1} B_{cl} r(k-m) \\ y(k) &= D_y r(k) + \sum_{m=1}^{k} C_y A_{cl}^{m-1} B_{cl} r(k-m) \end{aligned}$$

under the zero initial condition assumption. This results in

$$\mathbf{u} = \Phi_u \mathbf{r}, \quad \text{where}$$

$$\Phi_u = \begin{bmatrix} D_u & \cdots & 0_{p\times m} \\ C_u B_{cl} & \cdots & 0_{p\times m} \\ C_u A_{cl} B_{cl} & \cdots & 0_{p\times m} \\ \vdots & \cdots & \vdots \\ C_u A_{cl}^{N-3} B_{cl} & D_u & 0_{p\times m} \\ C_u A_{cl}^{N-2} B_{cl} & C_u B_{cl} & D_u \end{bmatrix} \in \mathbb{R}_{pN\times mN} \tag{6.12}$$

as the linear matrix equality between $\mathbf{r}$ and $\mathbf{u}$ and

$$\mathbf{y} = \Phi_y \mathbf{r}, \quad \text{where}$$

$$\Phi_y = \begin{bmatrix} D_y & \cdots & 0_{m\times m} \\ C_y B_{cl} & \cdots & 0_{m\times m} \\ C_y A_{cl} B_{cl} & \cdots & 0_{m\times m} \\ \vdots & \cdots & \vdots \\ C_y A_{cl}^{N-3} B_{cl} & D_y & 0_{m\times m} \\ C_y A_{cl}^{N-2} B_{cl} & C_y B_{cl} & D_y \end{bmatrix} \in \mathbb{R}_{mN\times mN} \tag{6.13}$$

as the linear matrix equality between $\mathbf{r}$ and $\mathbf{y}$.

The above linear matrix equalities can now be used to translate the closed-loop input $u(k)$ and output $y(k)$ constraints to the reference signal $r(k)$. With the definition of Φ_u in (6.12), the first condition on the input $u(k)$ in (6.4) can be written as an LMI of the form $G_1\mathbf{r} \leq b_1$, where

$$G_1 = \begin{bmatrix} \Phi_u \\ -\Phi_u \end{bmatrix} \in \mathbb{R}_{2pN \times mN} \ , \quad \text{and} \tag{6.14}$$

$$\begin{aligned} b_1 &= \begin{bmatrix} \bar{\mathbf{u}} \\ -\underline{\mathbf{u}} \end{bmatrix} \in \mathbb{R}_{2pN \times 1} \ , \\ \bar{\mathbf{u}} &= \text{vec}\left\{ \begin{bmatrix} \bar{u}(0) & \cdots & \bar{u}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN \times 1} \\ \underline{\mathbf{u}} &= \text{vec}\left\{ \begin{bmatrix} \underline{u}(0) & \cdots & \underline{u}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN \times 1} \end{aligned} \tag{6.15}$$

with the definition of the vector $\mathbf{u}$ in (6.9). Similarly, the second condition on the input $u(k)$ given in (6.5) can be written as an LMI of the form $G_2\mathbf{r} \leq b_2$, where

$$\begin{aligned} G_2 &= \begin{bmatrix} D\Phi_u \\ -D\Phi_u \end{bmatrix} \in \mathbb{R}_{2pN \times mN} \ , \\ D &= \begin{bmatrix} 0_{p \times p} & 0_{p \times p(N-1)} \\ -I_{p(N-1)} & 0_{p(N-1) \times p} \end{bmatrix} + I_{pN} \in \mathbb{R}_{pN \times pN} \end{aligned} \tag{6.16}$$

$$\begin{aligned} \text{and} \quad b_2 &= \begin{bmatrix} \bar{\mathbf{d}} \\ -\underline{\mathbf{d}} \end{bmatrix} \in \mathbb{R}_{2pN \times 1} \ , \\ \bar{\mathbf{d}} &= \Delta_T \cdot \text{vec}\left\{ \begin{bmatrix} \bar{d}(0) & \cdots & \bar{d}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN \times 1} \\ \underline{\mathbf{d}} &= \Delta_T \cdot \text{vec}\left\{ \begin{bmatrix} \underline{d}(0) & \cdots & \underline{d}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{pN \times 1} \end{aligned} \ . \tag{6.17}$$

It is worth noting that in case fixed values $\bar{u}(k) = \bar{u}$ and $\underline{u}(k) = \underline{u}$ are used, then the

expressions for $\bar{\mathbf{u}}$ and $\underline{\mathbf{u}}$ in (6.15) simplify to $\bar{\mathbf{u}} = 1_N \otimes \bar{u}$ and $\underline{\mathbf{u}} = 1_N \otimes \underline{u}$. Similar for fixed values $\bar{d}(k) = \bar{d}$ and $\underline{d}(k) = \underline{d}$ in (6.17), where $\bar{\mathbf{d}} = 1_N \otimes \bar{d}\Delta_T$ and $\underline{\mathbf{d}} = 1_N \otimes \underline{d}\Delta_T$, [73]. Finally, the condition (6.6) on the values of output $y(k)$ can be written as an LMI of the form $G_3\mathbf{r} \leq b_3$, where

$$G_3 = \begin{bmatrix} \Phi_y \\ -\Phi_y \end{bmatrix} \in \mathbb{R}_{2mN \times mN}, \quad \text{and} \tag{6.18}$$

$$\begin{aligned} b_3 &= \begin{bmatrix} \bar{\mathbf{y}} \\ -\underline{\mathbf{y}} \end{bmatrix} \in \mathbb{R}_{2mN \times 1}\ , \\ \bar{\mathbf{y}} &= \text{vec}\left\{ \begin{bmatrix} y_{des}(0) + \bar{\varepsilon}(0) & \cdots \\ y_{des}(N-1) + \bar{\varepsilon}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{mN \times 1} \\ \underline{\mathbf{y}} &= \text{vec}\left\{ \begin{bmatrix} y_{des}(0) - \underline{\varepsilon}(0) & \cdots \\ y_{des}(N-1) - \underline{\varepsilon}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{mN \times 1} \end{aligned} \tag{6.19}$$

for which the matrix Φ_y is defined in (6.13).

Similar to the input amplitude bounds used in open-loop input shaping, reference amplitude bounds (6.7) lead to an LMI of the from $G_4\mathbf{r} \leq b_4$ where

$$G_4 = \begin{bmatrix} I_{mN} \\ -I_{mN} \end{bmatrix} \in \mathbb{R}_{2mN \times mN}, \quad \text{and} \tag{6.20}$$

$$\begin{aligned} b_4 &= \begin{bmatrix} \bar{\mathbf{r}} \\ -\underline{\mathbf{r}} \end{bmatrix} \in \mathbb{R}_{2mN \times 1}\ , \\ \bar{\mathbf{r}} &= \Delta_T \cdot \text{vec}\left\{ \begin{bmatrix} \bar{r}(0) & \cdots & \bar{r}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{mN \times 1} \\ \underline{\mathbf{r}} &= \Delta_T \cdot \text{vec}\left\{ \begin{bmatrix} \underline{r}(0) & \cdots & \underline{r}(N-1) \end{bmatrix} \right\} \in \mathbb{R}_{mN \times 1} \end{aligned} \tag{6.21}$$

with the definition of the vector $\mathbf{r}$ in (6.11).

With all four conditions covered as LMIs in the form of $G_i\mathbf{r} \leq b_i,\ i = 1, \ldots, 4$, the feasibility for a closed-loop reference design can be formulated as the question of

$$\begin{gathered}\exists \mathbf{r} \in \mathbb{R}^{mN\times 1}, \quad \text{subject to} \quad G\mathbf{r} \leq b, \quad \text{with} \\ G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix} \in \mathbb{R}_{4pN+4mN\times mN}, \\ b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \in \mathbb{R}_{4pN+4mN\times 1}.\end{gathered} \tag{6.22}$$

The existence of $\mathbf{r}$ can be easily checked by a common transformation to replace the general *inequality* constraint by a pair of constraints: a general *equality* constraint containing an extra (slack) variable, and a *nonnegativity* bound on the slack variable [29]. Adding a slack variable $\mathbf{z} \in \mathbb{R}_{4pN+4mN\times 1} \geq 0$ to cover all the linear constraints and solve the optimization

$$\min_{z,r} \sum \mathbf{z} \quad \text{subject to} \quad G\mathbf{r} \leq b + \mathbf{z} \quad \text{and} \quad \mathbf{z} \geq 0$$

as a solution for $\mathbf{z}$ and $\mathbf{r}$ with $\mathbf{z} = 0$ will amount to a feasible solution $\mathbf{r}$. With the definition of the full variable $\mathbf{x} = \begin{bmatrix} \mathbf{r}^T & \mathbf{z}^T \end{bmatrix}^T \in \mathbb{R}_{4pN+5mN\times 1}$, the above optimization is written as a

standard linear programming problem

$$\min_{\mathbf{x}} f^T\mathbf{x}, \quad \text{subject to}$$
$$\begin{bmatrix} G & -I_{4pN+4mN} \\ 0_{4pN+4mN\times mN} & -I_{4pN+4mN} \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} b \\ 0_{4pN+4mN\times 1} \end{bmatrix} \tag{6.23}$$
$$\text{with} \quad f^T = \begin{bmatrix} 0_{1\times mN} & 1_{1\times 4pN+4mN} \end{bmatrix},$$

where matrices G and b are defined in (6.22). A feasible solution to (6.23) now produces a reference signal $r(k)$ that satisfies the feasibility for the closed-loop reference design. The actual reference signal $r(k),\ k = 0, 1, \ldots, N-1$ can be extracted from $\mathbf{r}$ contained in $\mathbf{x} = \begin{bmatrix} \mathbf{r}^T & \mathbf{z}^T \end{bmatrix}^T$ via the inverse operation of (6.11).

6.5.2 Optimal reference

Although the LP problem in (6.23) does give a quick answer to whether or not a feasible reference design exists based on the choices of the conditions formulated in (6.4), (6.5) and (6.6), the resulting reference $r(k)$ is not unique. Again, solutions for $r(k)$ may result in closed-loop input sequences in which $r(k)$ oscillates within the rate bounds (6.7) but do not influence the input or output constraints as the oscillations of $r(k)$ are blocked by any possible (right half plane) transmission zeros of the controller $C(q)$ [74].

Additional uniqueness and an optimal value of the multi-variable reference $r(k) \in \mathbb{R}_{m\times 1}$ for $k = 0, 1, \ldots, N$ can be enforced by also minimizing the finite-time variance of the rate-of-change of the reference given by

$$\mu_N^2 = \frac{1}{N}\sum_{k=0}^{N-1}\left(\frac{r(k) - r(k-1)}{\Delta_T}\right)^2 \tag{6.24}$$

to avoid penalizing the absolute values of $r(k)$. This will now ensure that the resulting reference signal $r(k)$ will be as smooth as possible. It should be noted that a smooth reference $r(k)$ will also lead to a smooth input $u(k)$, provided the controller $C(q)$ of the closed-loop system (6.2) is well designed. In addition, during the reference shaping, the rate constraints on the input $u(k)$ with the matrices G_2 in (6.16) and b_2 (6.17), ensure an admissible control output $u(k)$ is obtained. The value of μ_N^2 in (6.24) can be related to the design variable $\mathbf{r}$ via

$$\begin{gathered} \mu_N^2 = \tfrac{1}{2}\mathbf{r}^T H \mathbf{r}, \quad H = \tfrac{2}{N\Delta_T^2} E^T E \ , \\ E = \begin{bmatrix} 0_{m\times m} & 0_{m\times m(N-1)} \\ -I_{m(N-1)} & 0_{m(N-1)\times m} \end{bmatrix} + I_{mN} \in \mathbb{R}_{mN\times mN} \end{gathered} \tag{6.25}$$

Assuming the closed-loop reference design passed the feasibility test via the LP problem (6.23), the actual closed-loop reference design can be optimized via a quadratic programming problem,

$$\begin{gathered} \min_{\mathbf{u}} \frac{1}{2}\mathbf{r}^T H \mathbf{r} + \mathbf{f^T u} \ \text{ subject to } \ \mathbf{Gr} \leq \mathbf{b}, \ \text{with} \\ f^T = 0_{1\times pN}, \quad H = \frac{2}{N\Delta_T^2} E^T E \end{gathered} \tag{6.26}$$

where E is defined in (6.25) and the matrices G and b are defined in (6.22). A feasibly solution to (6.26) now produces a reference signal $r(k)$ that has minimized the finite-time variance (6.24) on the rate-of-change of the reference $r(k)$, while satisfying the conditions of (6.4), (6.5) on the input $u(k)$ and the condition (6.6) on the output $y(k)$.

6.6 Experimental simulation

Consider the discrete-time ZOH equivalent of the SISO continuous-time system

$$G(s) = \frac{1}{ms^2 + ds + k} \ ,$$

modeling a single mass m, spring k, damper d mechanical system; with a force $u(k)$ as an input and a position $y(k)$ as an output. This example is chosen for its simplicity, while retaining the intuition and importance of open-loop input shaping and closed-loop reference shaping for a dynamic system with a single resonance mode. For numerical evaluation we consider a nominal model with $m = 1$, $k = 100$ and $d = 2$ with a sampling time $\Delta_T = 0.05$ leading to the ZOH equivalent

$$G(q) = \frac{1}{100} \cdot \frac{0.1185q + 0.1145}{q^2 - 1.627q + 0.9048} \ , \tag{6.27}$$

with a discrete-time zero located at -0.9669 inside the unit disc. It is worth noting that $G(s)$ and $G(q)$ have a static gain of $1/100$, which will be used to create a simple quasi-static input $u_q(k)$ given by $u_q(k) = y_{des}(k)/100$ that would ignore the dynamics of $G(q)$ and serves as a baseline comparison for dynamic input shaping.

For the closed-loop reference shaping we consider a normalized $y_{des}(k)$ of an even number of N samples. The $y_{des}(k)$ is created by the filtering of a constant and equal acceleration and deceleration over $N_s < N$ samples in the middle of the interval of the N samples, where N_s is the number of samples where the transitions of desired values are defined. The filtering to obtain the desired position $y_{des}(k)$ is a discrete-time ZOH equivalent of a double integrator. For a sampling time Δ_T, such a normalized output $y_{des}(k)$ can be characterized by

$$y_{des}(k) = \Delta_T^2 \tfrac{q+1}{(q-1)(q-1)} a_{des}(k),$$

$$a_{des}(k) = \begin{cases} 0, & k = 0, \ldots, N/2 - N_s \\ c, & k = N/2 - N_s + 1, \ldots, N/2 \\ -c, & k = N/2 + 1, \ldots, N/2 + N_s \\ 0, & k = N/2 + N_s + 1, \cdots, N - 1 \end{cases}$$

where c is chosen such that $y_{des}(k)$ is normalized to $y_{des}(0) = 0$ and $y_{des}(N-1) = 1$. In the numerical study included here, the values of $N = 40$ and $N_s = 4$ are chosen for a relatively short transition of $y_{des}(k)$ from 0 to 1 in just $2N_s = 8$ samples to illustrate the dynamic effects of $G(q)$ in both the input and reference planning.

For the closed-loop reference shaping we consider the control of $G(q)$ with digital PID controller specified by

$$C(q) = 100 \cdot \frac{2.8q^2 - 4.8q + 2.6}{q^2 - q} \ , \tag{6.28}$$

which will be used in the feedback connection (6.2). The PID controller provides some damping of the resonance mode of $G(q)$, but more importantly, it provides steady-state output tracking. The property of steady-state output tracking is used to create a simple quasi-static reference $r_q(k)$ given by $r_q(k) = y_{des}(k)$ that would ignore the dynamics of $G(q)$ and serves as a baseline comparison for dynamic reference shaping.

For the noncausal reference shaping, the time-invariant input amplitude bounds and input rate limits are set to: $\bar{u} = -\underline{u} = 110$ and $\bar{d} = -\underline{d} = 500$. For closed-loop reference shaping, reference amplitude bounds and set $\bar{r} = -\underline{r} = 1.1$. Results for the (nominal) closed-loop reference shaping are summarized in Figure 6.1 where it can be observed that the open-loop shaped reference guarantees that the output remains within the bounds after the transition period. In addition, the control signal $u(k)$ satisfies the imposed constraints of (6.4) and (6.5). In comparison with a quasi-static reference $r(k) = y_{des}(k)$, the shaped reference signal does a better job of ensuring the output settles within the imposed margins around the desired output $y_{des}(k)$ in the presence of a feedback controller $C(q)$, while satisfying the constraints on the size and rate of change of the input $u(k)$.

It is clear again that the reference shaping does lead to a noncausal input $r(k)$ that changes *before* the desired transition takes place in $y_{des}(k)$. Causality can be enforced by

imposing time-varying amplitude bounds $\bar{r}(k)$ and $\underline{r}(k)$ given in (6.7) to ensure only a causal solution is found. An example is illustrated in the reference shaping of Figure 6.2 where the reference $r(k)$ is bounded to not change **before** the transition occurs in the desired output $y_{des}(k)$ at $t = k\Delta_t = 0.8$ seconds. The resulting causal reference shape $r(k)$ is a signal pattern for the closed-loop system to ensure the output $y(k)$ is placed within its desired bounds around $y_{des}(k)$. The shape of $r(k)$ is not only determined by $y_{des}(k)$ and the imposed margins, but also dependent on the closed-loop dynamics and the rate-of-change bounds imposed on $u(k)$.

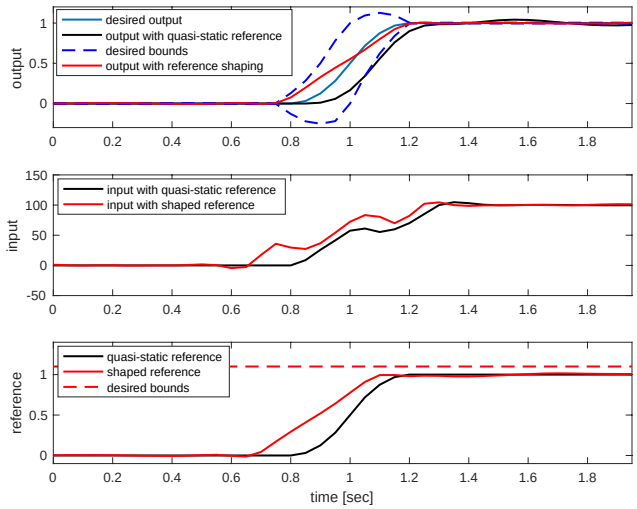


Figure 6.1: Comparison between noncausal closed-loop reference shaping for the model/controller $G(q)$, $C(q)$ given in (6.27), (6.28) and a quasi-static input $r(k) = y_{des}(k)$ for $N = 40$ and $N_s = 4$.

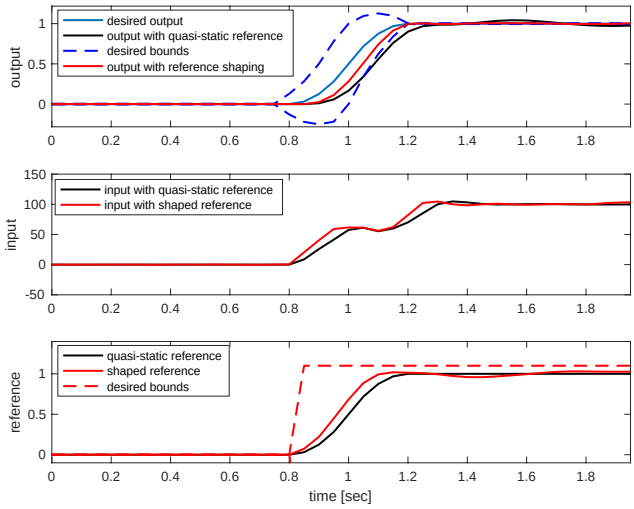


Figure 6.2: Comparison between causal closed-loop reference shaping for the model/controller $G(q)$, $C(q)$ given in (6.27), (6.28) and a quasi-static input $r(k) = y_{des}(k)$ for $N = 40$ and $N_s = 4$ via time varying bounds on the reference $r(k)$ to allow $r(k)$ to change only after $t = 0.8$ sec.

6.7 Conclusions

A closed-loop reference shaping problem has been described in this paper and solved as a QP problem. First, feasibility was verified via LP and an optimal reference profile was computed formulating the problem as QP. It is shown that the shaped reference signal maintains the system response within the desired bounds, but it is also noticeable how the reference shaping leads to a noncausal signal profile, which might not be desirable or feasible for certain systems. A time-dependent constraint added to the convex optimization problem formulation allows us to obtain a reference for a causal response. The mass-spring-damper system in closed-loop with a PID controller illustrates how the reference shaping can enforce causality, via a time-dependent bound signal. The convex optimization problem formulation with a time-dependent constraint computes a valid reference shaping for systems where causal profiles are required. Reference shaping working with a closed-loop system is a safe way to compensate for errors due to model uncertainty or disturbance signals; for which

reference shaping, by its open-loop nature, cannot compensate. As part of future research, robustness to the reference shaping problem should be considered, to ensure the reference profile can be applied to an existing closed-loop system with uncertainties.

6.8 Acknowledgement

This chapter, in full, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Reference signal shaping for closed-loop systems with causality constraints." *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1838-1843. The dissertation author is the first author of this article.

Chapter 7

Robust Input Shaping Design for Uncertain Second-order Dynamics Systems

7.1 Chapter Abstract

Models of dynamics systems with resonance modes might include uncertainty on the exact numerical value of the damping and natural frequency. These models are referred to as parametric uncertainty models. Input shaping (IS) design is often computed for a nominal model, rejecting any possible variations in the parameters. Input shaping that is robust for a parametric uncertainty on the resonance mode over a range is required. The process defined in this paper considers the problem of robust input shaping, where a single input signal is designed for a range of uncertainty in the parameters of a dynamic model. Robust input shaping is achieved by considering the extreme cases in the parameter variations, and designing a robust input signal for which constraints on both input and output signals are satisfied. The approach is tested on a second order model with 10%

variation in the natural frequency and 10% variation on the damping, which illustrates the ability to design a robust input signal.

7.2 Introduction

Working with a physical system, the search for a model (plant) could be computed mainly via physical equations or system identification when input/output data is available. Models which closely describe physical systems are difficult or impossible to precisely characterize; when modeling a physical system via the physical equations, the parameters might not have the exact values. Common problems in system identification include: limited data, data containing a degree of uncertainty, unmodeled dynamics (since any dynamical model will neglect some physical phenomena) and unknown initial conditions [75],[76]. A model could be thoroughly detailed, but it is never an exact representation of the real physical system and in many applications noise is clearly present [75]. Having said that, modeling errors shall be considered as they might affect the performance of the control system [11].

In linear time-invariant (LTI) control systems, the concept of input shaping (IS) is not new; it is a technique originally used for defining a shaped command input to eliminate or reduce unwanted system vibration [1]. New tools have emerged to solve the IS problem [7, 8]. Commonly, input shaping is used in open-loop for multiple-input multiple-output (MIMO) systems in continuous time and discrete time [2], [3], [4]. With the input shaping schemes being inherently open-loop, uncertainties in the open-loop model can lead to system performance degradation [12].

For the most common methods of input shaping, e.g. zero vibration (ZV), zero vibration derivative (ZVD) and extra insensitive (EI); the primary measure of performance is the residual vibration. The fact that input shaping does not take into account parametric

uncertainty has been approached by different methods where further measures are taken into account, such as weighting the ranges of system parameter values according to the expected modeling errors [13, 14]; allowing the shaper to contain negative impulses [15]; a probabilistic approach of the uncertain model parameters with a polynomial chaos approach [16]; design of closed loop input shapers [17]; just to mention a few approaches. These methods show great improvement, while most of them are sill based on the ZV, ZVD and EI methods.

The approach to robust input shaping in this paper differs from the traditional approaches. The robust input shaping is formulated as a convex optimization problem by imposing simultaneous linear constraints on the input and output signals of multiple models that represent the extreme cases of the parametric uncertainties. By approaching the robust input shaping problem as a general convex optimization problem, the desired response can be further optimized by defining constraint functions and limits or bounds for these constraints [20], [29]. All constraints are combined in a linear matrix inequality (LMI) [72] which can be solved as either a linear program (LP) problem or as a quadratic program (QP) problem. In this paper robust input shaping is formulated as an open-loop problem with linear constraints on input and output signals, imposed to formulate an LP problem to verify feasibility and a QP problem to compute the optimal input profiles.

Robust control consists of the control methods utilized to operate in the uncertain parameters of a model; the predominant idea is to compensate for the model uncertainty via feedback [75, 77, 78]. Feedback would be omitted in most control systems if there were no disturbance and uncertainties [10]. Since this IS problem is approached as an open-loop problem, the feedback solution is not suitable for this method. Input shaping has proved effective on different types of systems, however IS methods neglect taking into account the modeling errors and parameter uncertainty. In real systems, there is some knowledge of the model uncertainties; taking this knowledge into consideration leads to a better shaped

input [13]. Since the uncertainties on the model parameters might affect the performance of the output to the input shaping design, the design must consider these uncertainties in order to be robust.

An approach to uncertainty is illustrated in [79], where the analysis is based in an state-space model and the relationship between the parameter perturbation range and the associated eigenvalue/eigenvector. In this paper, a direct parametric approach is utilized for the robust input shaping, which will be illustrated on a second-order system. The parametric uncertainty is modeled as a normalized variation weighted by a perturbation matrix. The perturbation matrix allows the normalized parametric uncertainty to be distributed over the different parameters in a discrete-time model that is used for the robust input shaping design. The robust input shaping design illustrated consists of the application of the input shaping problem to obtain a shaped input signal, not only for the nominal model, but also for a number of extreme models.

The extreme models are defined as the models for which the response is at the extremes of the bounds for the response. As a first approach, the following conjecture of the extreme models is made; the extreme values in the normalized variation of the parametric uncertainty are used to define the extreme models for which the input signal must be designed. The robust input shaping design is computed for these models and the outputs for all the models within the uncertainty range are evaluated. If any output is outside the bounds, the worst case is considered an extreme model.

The contribution of this paper is to propose a method to analyze the model uncertainty through the parameters in the equations of the discrete-time model, and to compute a single robust input shaping signal that is able to satisfy constraints on both the open-loop input and the desired open-loop output over the range of the parametric uncertainty.

The paper is structured as follows. In section 7.3 the open-loop model and the notation for model uncertainty are described, as well as the computation of a perturbation

matrix which reflects the uncertainties in the model. Section 7.4 elaborates on the input shaping process, the constraints, feasibility and optimal solution. Section 7.5 illustrates the previous definitions with a simulation example.

7.3 Defining the parametric uncertainty model

Consider the nth order state space representation of a discrete-time model $G(q)$ given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned} \tag{7.1}$$

sampled with a sampling time of Δ_T seconds. Zero initial conditions $x(0) = 0$ and $u(0) = 0$ are assumed. Open-loop input shaping is now defined as the design of an input $u(k)$ for (7.1) over a finite time interval $k = 1, \ldots, N$ so that the output $y(k)$ and the resulting shaped input $u(k)$ satisfy certain (time-varying) constraints. As the input design problem is mostly an input/output based design approach, we will move to an input/output dynamic representation of (7.1), but the state space matrices A, B, C and D will be used in the formulation of the constraints for the sake of generality.

Consider the system

$$G(q, \theta) = \frac{B(q, \theta)}{A(q, \theta)} \ , \tag{7.2}$$

where θ is a vector of uncertain parameters and $B(q, \theta)$, $A(q, \theta)$ are polynomials in q with coefficients which depend on θ.

It is assumed that the set of model $\mathcal{G}$ is represented by a nominal model $G(q, \hat{\theta})$ with a nominal parameter $\hat{\theta}$ and parameter uncertainty modeled by a finite number Q of independent perturbations.

To address parametric uncertainty for a robust input shaping design, the set $\mathcal{G}$ of discrete-time models $G(q, \theta) \in \mathcal{G}$ shall be considered. The uncertainty on the parameter θ

constitutes the set of models for which the robust input signal must comply.

$$\mathcal{G}(q,\theta) = \frac{b_1 q^{-1} + b_2 q^{-2} + \cdots + b_m q^{-m}}{1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_n q^{-n}} \ .$$

An uncertain polynomial $a(q,\theta)$

$$a(q,\theta) = \sum_{i=1}^{n} a_i(\theta) q^i \ , \tag{7.3}$$

is used to define an interval polynomial family, with uncertainty θ. N_c is the number of coefficients in the polynomial that reflect uncertainty. $\hat{\theta}$ represents the nominal value of the coefficients in G. The uncertainty on the parameters θ of the model are captured by a linear parametric variation

$$\theta = \hat{\theta} + \hat{P} \cdot \delta_k, \quad |\delta_k| \leq 1, \quad k = 1, \ldots, Q \ , \tag{7.4}$$

where $\hat{P}$ denotes an additive perturbation matrix. $\hat{P}$ is used to distribute the normalized parametric uncertainty δ_k over the parameters θ of the model.

For a scalar uncertainty the above description simplifies to $\theta = \hat{\theta} + \hat{P}\delta$, where $\hat{P}$ is a perturbation vector.

For n variations of each δ, a new θ is computed with all the variations in the model. Both are represented as $\Delta = [\delta_1, \delta_2 \ldots \delta_n] \in \mathbb{R}_{n^Q \times Q}$ and $\theta = [\theta_1, \theta_2, \ldots, \theta_n] \in \mathbb{R}_{N_c \times n^Q}$, respectively. It should be noted that if the perturbation matrix $\hat{P}$ is not know, but the set of Δ and θ are known, the perturbation matrix $\hat{P}$ in the linear parametric variation (7.4)

can be computed via a least squares (LS) solution.

$$[\theta_1, \theta_2, \ldots, \theta_n] = \hat{\theta} + P\Delta^T + \varepsilon(P) \ ,$$

$$\text{Defining } \ \Theta = \ [\theta_1 - \hat{\theta} \quad \theta_2 - \hat{\theta} \quad \ldots \quad \theta_n - \hat{\theta}] \ , \text{ we have}$$

$$\Theta = P\Delta^T + \varepsilon(P).$$

Minimizing the LS error $\text{Tr}\,\{\varepsilon(P)^T\varepsilon(P)\}$ leads to the solution

$$\hat{P} = \Theta\Delta[\Delta^T\Delta]^{-1} \ \in \mathbb{R}_{N_c \times Q}. \tag{7.5}$$

With the $\hat{P}$ matrix, the variation of the model can be estimated directly with the parameters in the transfer function. This provides a tool for analysis of the uncertainties. In Section 7.5 the perturbation matrix computation is exemplified.

7.4 Robust Input Shaping

7.4.1 Linear input and output constraints

The robust input shaping performance of an uncertain discrete-time model is achieved by designing a single open-loop shaped input signal $u(k)$ for m extreme discrete-time models. Utilizing the same shaped input signal for the computation of the m outputs $y(k)$ of each model generates outputs within the imposed constraints. Since the purpose of this process is to obtain one signal only for different models, for this particular exercise the number of inputs $p = 1$. The following restrictions will be imposed on the control (input) signal $u(k)$ and system outputs $y(k)$:

- The single input $u(k)$ is bounded between distinct time-varying lower and upper

amplitude bounds, which satisfies the amplitude constraints, i.e.

$$\begin{gathered} \underline{u}(k) \leq u(k) \leq \bar{u}(k),\ k = 0, 1, \ldots, N-1 \\ \text{where}\ \ \underline{u}, \bar{u} \in \mathbb{R}_{p\times 1} \end{gathered} \tag{7.6}$$

- The single inputs $u(k)$ sampled at Δ_T seconds is rate limited between distinct time-varying lower and upper rate bounds, which satisfies the rate-limit constraints, i.e.

$$\begin{gathered} \underline{d}(k) \leq \tfrac{u(k)-u(k-1)}{\Delta_T} \leq \bar{d}(k), \\ k = 0, 1, \ldots, N-1 \\ \text{where } \underline{d}, \bar{d} \in \mathbb{R}_{p\times 1} \text{ and } u(-1) = 0 \end{gathered} \tag{7.7}$$

- The m outputs in $y(k)$ are bounded between a time-varying lower and upper bounds around the same time-varying desired output profile $y_{des}(k)$, which defines the allowable perturbations for the outputs $y(k)$, i.e.

$$\begin{gathered} \underline{\varepsilon}(k) \leq y(k) - y_{des}(k) \leq \bar{\varepsilon}(k), \\ k = 0, 1, \ldots, N-1 \\ \text{where}\ \ \underline{\varepsilon}(k), \bar{\varepsilon}(k) \in \mathbb{R}_{m\times 1} \end{gathered} \tag{7.8}$$

7.4.2 Feasibility for open-loop input design

Feasibility for open-loop input design is defined as the problem of finding feasible $p \times N$ values of the finite-time input signal,

$$u(k) \in \mathbb{R}_{p\times 1}, \quad k = 0, 1, \ldots, N-1 \tag{7.9}$$

for the open-loop model given in (7.1), such that the linear conditions hold. Any point $\hat{x}$ is said to be feasible if it satisfies the objective function and all the constraints [20, 29].

Linear conditions in (7.6), (7.7) and (7.8) are rewritten into a linear matrix inequality (LMI) for feasibility and solved as a QP problem.

For vector notation purposes, the values of $u(k) \in \mathbb{R}_{p\times 1}, \quad k = 0, 1, \ldots, N-1$ contained in the matrix

$$U = \left[\; u(0), u(1), \ldots, u(N-1) \; \right] \in \mathbb{R}_{p\times N}$$

is first written as

$$\mathbf{u} = \text{vec}\{U\} = \left[\; u(0)^T, \ldots, u(N-1)^T \; \right]^T \in \mathbb{R}_{pN\times 1}$$

to obtain a vector $\mathbf{u}$ of desired multi-variable input values.

The first condition on the input $u(k)$ in (7.6) can be rewritten as an LMI of the form $A_1\mathbf{u} \leq b_1$ where

$$A_1 = \begin{bmatrix} I_{pN} \\ -I_{pN} \end{bmatrix}, \quad b_1 = \begin{bmatrix} \bar{\mathbf{u}} \\ -\underline{\mathbf{u}} \end{bmatrix}. \tag{7.10}$$
$$\in \mathbb{R}_{2pN\times pN} \qquad \in \mathbb{R}_{2pN\times 1}$$

The second condition on the input $u(k)$ given in (7.7) can be written as an LMI of the form $A_2\mathbf{u} \leq b_2$, where

$$A_2 = \begin{bmatrix} E \\ -E \end{bmatrix}, \quad b_2 = \begin{bmatrix} \bar{\mathbf{d}} \\ -\underline{\mathbf{d}} \end{bmatrix}, \quad \text{with}$$
$$\in \mathbb{R}_{2pN\times pN}, \qquad \in \mathbb{R}_{2pN\times 1}, \tag{7.11}$$
$$E = \begin{bmatrix} 0_{p\times p} & 0_{p\times p(N-1)} \\ -I_{p(N-1)} & 0_{p(N-1)\times p} \end{bmatrix} + I_{pN} \in \mathbb{R}_{pN\times pN}$$

with $\bar{\mathbf{d}} = \Delta_T\{[\bar{d}(0) \; \bar{d}(1) \; \cdots \; \bar{d}(N-1)]\} \in \mathbb{R}_{N\times 1}$ and $\underline{\mathbf{d}} = \Delta_T\{[\underline{d}(0) \; \underline{d}(1) \; \cdots \; \underline{d}(N-1)]\} \in$

$\mathbb{R}_{N\times 1}$.

The mapping of the input values $u(k)$ contained in $\mathbf{u}$ to the output values $y(k)$ contained in $\mathbf{y}$ is done with the linear state space description of the open-loop model in (7.1) via the convolution sum

$$y(k) = Du(k) + \sum_{m=1}^{k} CA^{m-1}Bu(k-m)$$

This results in $\mathbf{y} = \Phi\mathbf{u}$, with $\Phi \in \mathbb{R}_{mN\times N}$. The condition (7.8) on the values of the output $y(k)$ can be written in an LMI of the form $A_3\mathbf{u} \leq b_3$, where

$$\underset{\in \mathbb{R}_{2mN\times N}}{A_3 = \begin{bmatrix} \Phi \\ -\Phi \end{bmatrix}}, \quad \underset{\in \mathbb{R}_{2mN\times 1}}{b_3 = \begin{bmatrix} \bar{\mathbf{y}} \\ -\underline{\mathbf{y}} \end{bmatrix}}, \tag{7.12}$$

with $\bar{y} = \text{vec}\{[y_{des}(0) + \bar{\epsilon}(0), y_{des}(1) + \bar{\epsilon}(1), \ldots, y_{des}(N-1) + \bar{\epsilon}(N-1)]\} \in \mathbb{R}_{mN\times 1}$ and $\underline{y} = \text{vec}\{[y_{des}(0) - \underline{\epsilon}(0), y_{des}(1) - \underline{\epsilon}(1), \ldots, y_{des}(N-1) - \underline{\epsilon}(N-1)]\} \in \mathbb{R}_{mN\times 1}$.

With all three conditions covered as LMIs, the feasibility for an open-loop input design can be formulated as

$$\begin{gathered} \exists \mathbf{u} \in \mathbb{R}^{pN\times 1}, \quad \text{subject to} \quad A\mathbf{u} \leq b, \quad \text{with} \\ A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \in \mathbb{R}_{4pN+2mN\times pN}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \in \mathbb{R}_{4pN+2mN\times 1} \end{gathered} \tag{7.13}$$

7.4.3 Optimal open-loop input design

The problem in (7.13) can have additional uniqueness and an optimal value of the multi-variable input $u(k) \in \mathbb{R}_{p\times 1}$ for $k = 0, 1, \ldots, N$ by also minimizing the finite-time

variance of the rate-of-change of the input given by

$$\sigma_N^2 = \frac{1}{N}\sum_{k=0}^{N-1}\left(\frac{u(k)-u(k-1)}{\Delta_T}\right)^2 \tag{7.14}$$

to avoid penalizing the actual input values $u(k)$. The decision for miniization on σ_N^2 is based on the benefit of a smoother rate-of-change for the input. The variance on the rate-of-change can be computed via the matrices defined as it is easy to verify that

$$\sigma_N^2 = \frac{1}{2}\mathbf{u}^T H\mathbf{u}, \quad \text{with} \quad H = \frac{2}{N\Delta_T^2}E^T E.$$

The actual open-loop input design can be optimized via the QP problem,

$$\begin{gathered}\min_{\mathbf{u}} \frac{1}{2}\mathbf{u}^T H\mathbf{u} + \mathbf{f^T u} \quad \text{subject to} \quad \mathbf{Au} \leq \mathbf{b}, \quad \text{with} \\ f^T = 0_{1\times N}, \quad H = \frac{2}{N\Delta_T^2}E^T E.\end{gathered} \tag{7.15}$$

A feasibly solution to (7.15) produces an input signal $u(k)$ that has minimized (7.14) under the constraints in (7.6), (7.7) and (7.8).

7.5 Simulation Study

7.5.1 Definition of discrete-time model uncertainty

The second-order model is used for the simplified dynamic analysis and the capability of an oscillatory response to a step input while generating results that are easier to interpret. Consider the continuous-time second-order resonant system given by

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\ , \tag{7.16}$$

modeling a mass-spring-damper mechanical system, with a force $u(k)$ as an input and a position $y(k)$ as an output, for which a 10% variability in the natural frequency ω_n and a 10% variability in damping ζ are present. The variability of the natural frequency ω_n and damping ζ is an example of parametric uncertainty.

Perturbations in ω_n in the continuous-time model $G(s)$ will typically map non-linearly to the parameters θ of the discrete-time model $G(q,\theta)$. However, a linear approximation with (7.4) can be shown to provide an excellent way to capture the uncertainty. For illustration purposes, consider a nominal continuous-time model $G(s)$ with $K = 1/100$, $\omega_n = 10$ and $\zeta = 0.2$ to find a nominal discrete-time model $G(q,\hat{\theta})$ with a sampling time $\Delta_T = 0.05$. A Zero-Order-Hold discrete-time equivalent of the system in (7.16),

$$G(q,\theta) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}, \quad \hat{\theta} = \begin{bmatrix} b_1 & b_2 & a_1 & a_2 \end{bmatrix}^T .$$

7.5.2 Input Shaping for the nominal model

For the input shaping is considered a normalized $y_{des}(k)$ of an even number of N samples. The $y_{des}(k)$ is created by the filtering of a constant and equal acceleration and deceleration over $N_s < N/2$ samples in the middle of the interval of the N samples, where N_s is the number of samples where the transitions of desired values are defined. The filtering to obtain the desired position $y_{des}(k)$ is a discrete-time ZOH equivalent of a double integrator. For a sampling time Δ_T, such a normalized output $y_{des}(k)$ can be characterized

by

$$y_{des}(k) = \Delta_T^2 \frac{q+1}{(q-1)(q-1)} a_{des}(k),$$

$$a_{des}(k) = \begin{cases} 0, & k = 0, \ldots, N/2 - N_s \\ c, & k = N/2 - N_s + 1, \ldots, N/2 \\ -c, & k = N/2 + 1, \ldots, N/2 + N_s \\ 0, & k = N/2 + N_s + 1, \cdots, N-1 \end{cases}$$

where c is chosen such that $y_{des}(k)$ is normalized to $y_{des}(0) = 0$ and $y_{des}(N-1) = 1$. In the numerical study included here, the values of $N = 40$ and $N_s = 4$ are chosen for a relatively short transition of $y_{des}(k)$ from 0 to 1 in just $2N_s = 8$ samples to illustrate the dynamic effects of $G(q)$ in the input planning.

The property of steady-state output tracking is used to create a simple quasi-static input given by $u_q(k) = y_{des}(k)$ that would ignore the dynamics of $G(q)$ and serves as a baseline comparison for dynamic input shaping. For the input shaping, the time-invariant causality is enforced by imposing time-variant amplitude bounds $\bar{u}(k)$ and $\underline{u}(k)$ [80]. Results for the (nominal) open-loop input shaping are summarized in Figure 7.1 where it can be observed that the open-loop shaped input guarantees that the output remains within the (narrow) bounds after the transition period.

7.5.3 Computation of the perturbation matrix

Given by the nominal parameter $\hat{\theta}$ from G

$$\hat{\theta} = \begin{bmatrix} 0.0011 \\ 0.0011 \\ -1.5968 \\ 0.8187 \end{bmatrix} \tag{7.17}$$

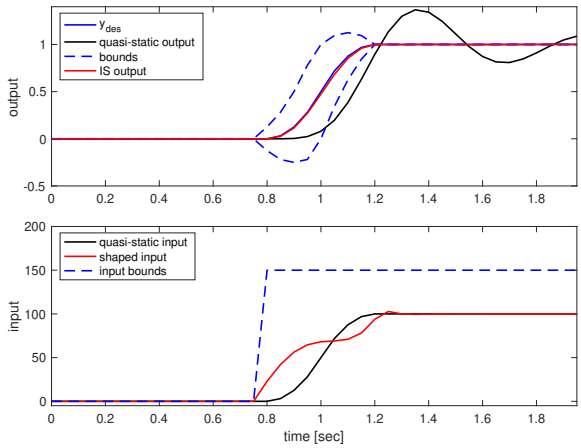


Figure 7.1: Comparison between the quasi-static input and input shaping for the nominal model G. Desired output y_{des} (solid blue line), bounds (dashed blue line), quasi-static signals (solid black line), input shaping signals (solid red line).

Combining $n = 100$ linearly spaced values for each, $\delta_1(k)$ and $\delta_2(k)$, in the range from -0.1 to 0.1 to perturb $\omega_n = 10 \cdot (1 + \delta_1(k))$ and $\zeta = 0.2 \cdot (1 + \delta_2(k))$, the result is $\Delta \in \mathbb{R}_{100^2 \times 2}$. Computing the resulting parameters ZOH discrete-time model parameter $\theta(k)$ organized in the 4×100 matrix

$$\begin{aligned} \theta_{\omega_n} &= \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_{100} \end{bmatrix} \in \mathbb{R}_{4\times 100} \ , \\ \theta_{\zeta} &= \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_{100} \end{bmatrix} \in \mathbb{R}_{4\times 100} \ , \end{aligned} \tag{7.18}$$

can be illustrated that the (non-linear) parameter perturbations in θ can be approximated with (7.4) using a single $|\delta| \leq 1$ and a perturbation matrix P. A least squares approximation is utilized to compute P as shown in (7.5),

$$P = \begin{bmatrix} 0.002 & 0 \\ 0.002 & 0 \\ 0.0577 & 0.0142 \\ -0.0164 & -0.0164 \end{bmatrix} . \tag{7.19}$$

Note that P indicates all parameters in θ are perturbed with the proper scaling to

capture the perturbation of the frequency ω_n and damping ζ. It is worthwhile to note that the values $\delta_1 = \pm 1$ in (7.4) will indicate the extreme values of the 10% frequency and 10% damping variations.

7.5.4 Extreme models and input shaping

It is not defined what the extreme models are, so as an initial conjecture, the extreme models G_1, G_2, G_3 and G_4 are defined from the extreme values of the natural frequency ω_n and damping ζ. This is reflected as the extreme values of P, computed with $\delta = \pm 1$, in order to define the extreme models, as per (7.4). An illustration of shaped input for the nominal model G, applied to the extreme models is given in Figure 7.2. The response of the extreme models to the nominal shaped input, is clearly not within the constraints ($\bar{y}$, $\underline{y}$) defined for the output (7.8).

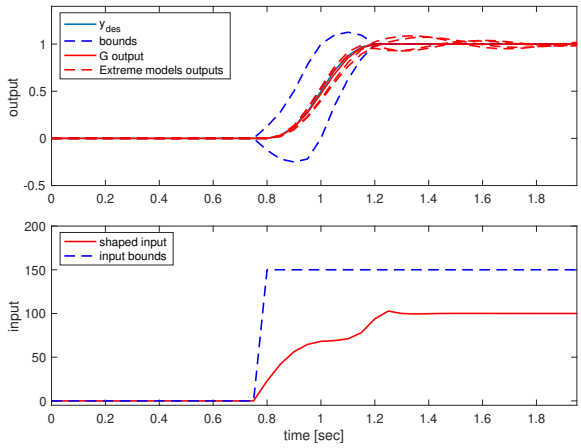


Figure 7.2: Comparison between input shaping for the nominal model G applied to the extreme model G_1, G_2, G_3 and G_4. Desired output y_{des} (solid blue line), bounds (dashed blue line), input shaping signals (solid red line), extreme models (dashed red line).

The input shaping process is computed a second time considering only the extreme models. Adding the nominal model G to the computation will not improve the shaped input signal, since it is a model covered by the extremes. The new input shaping design

signal is computed for a new $G = [G_1\ G_2\ G_3\ G_4]$, which illustrates the fact of the process for robust input shaping is the same as the process for input shaping, for one, two or more models. The linear input/output constraints and desired output remain the same, while having a different transfer function. To estimate the shaped input for the extreme models the constraints on the output are wider in order to allow the output of the model to fit, but it is not as wide as their initial response on Figure 7.2. Figure 7.3 illustrates the shaped input computed for the extreme models and the response. It can be appreciated in Figure 7.3 the extreme models fit within the new constraints.

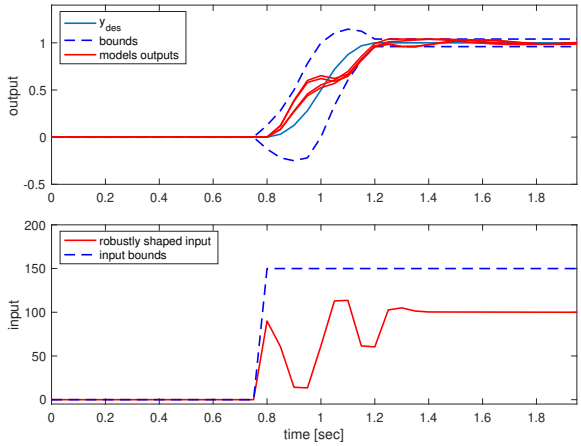


Figure 7.3: Input shaping computed for the extreme models G_1, G_2, G_3 and G_4. Desired output y_{des} (solid blue line), bounds (dashed blue line), input shaping signals (solid red lines).

In order to test the shaped input signal for the extreme models, an evaluation of 100 randomly variations of δ, (within $|\delta| \leq 1$) reflected on the perturbation matrix P is performed and utilized to create the equivalent θ for 100 models within the range. In Figure 7.4, the output from the 100 linearly spaced variations of δ and the respective perturbed models is illustrated. It can be seen on the plot that the output for each perturbed model fit within the constraints.

The results in Figure 7.4 prove the conjecture defined at the beginning of this section

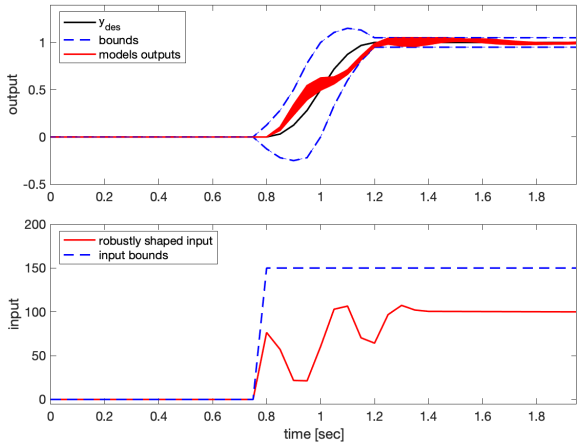


Figure 7.4: Input shaping response to 100 linearly spaced variations of $|\delta| \leq 1$ from (7.4). Desired output y_{des} (solid blue line), bounds (dashed blue line), input shaping signals (solid red line).

as correct; defining the extreme models G_1, G_2, G_3 and G_4 from the extreme values of the δ for ω_n and ζ. That might not be the case for all models. If the results were different, e.g. a model G_x has a value out of the constraint bounds, then robust input shaping would be computed one more time with a new extreme. The new definition for the extreme models would be including the new extreme as $G = [G_1\ G_2\ G_3\ G_4\ G_x]$. It is possible that an initial conjecture of extreme values might not be the best approach, so based on the results, the extreme models can be replaced. For the new computation, the linear input/output constraints and desired output remains the same, with a new transfer function.

To explore further the results and to make a more accurate evaluation of the results, an evaluation of the distance from the output of each model to the upper and lower constraint bounds is computed, (7.20).

$$\begin{aligned} \bar{x}_d &= (y(k) - \bar{y}(k)), \quad k = 1, 2, \ldots 100 \\ \underline{x}_d &= (\underline{y}(k) - y(k)), \quad k = 1, 2, \ldots 100 \end{aligned} \tag{7.20}$$

where $\bar{x}$ and $\underline{x}$ represent the distance of each model to the top and bottom constraint bounds.

Being $y(k)$ the response of a model, the distance to the (upper and lower) constraint bound shall be $\bar{x}_d \leq 0$ and $\underline{x}_d \leq 0$, where 0 indicates the response is at the boundaries (this is still valid), but a distance > 0, indicates the response $y(k)$ is out of the constraints bounds, thus is not valid. All cases where $\bar{x}_d > 0$ and $\underline{x}_d > 0$ indicate the models are out of bounds constraints. Finding the maximum distance

$$\begin{aligned} \bar{X}_d = \max_k(\bar{x}_d), \qquad \underline{X}_d = \max_k(\underline{x}_d), \\ X = \max\left(\bar{X}_d, \underline{X}_d\right) , \end{aligned} \tag{7.21}$$

$\bar{X}_d$ and and $\underline{X}_d$ indicate the extreme models with respect to the upper bound and lower bound, respectively, and X indicates the overall extreme model.

Figure 7.5 shows that the distance of all the models is within the constraint bounds. For this example, the extreme models $\bar{x}_d$ and $\bar{x}_d$ is for the extreme models, which match the extreme values of $\delta_1 = \pm 1$ and $\delta_2 = \pm 1$.

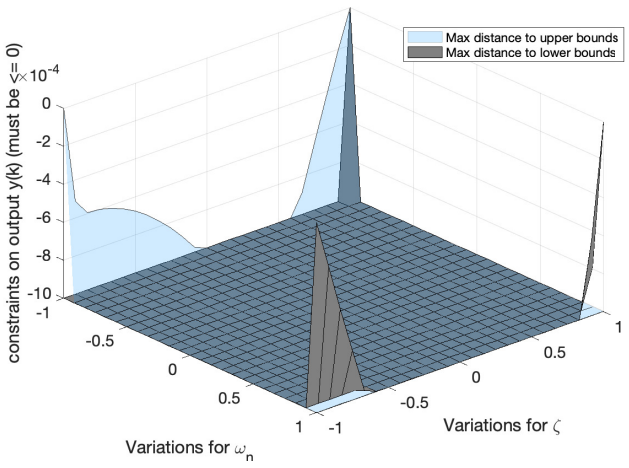


Figure 7.5: Distance to upper and lower bounds for each model as illustrated on (7.20).

7.6 Conclusions

The notion of input shaping for a linear dynamic discrete-time model can be formulated as an open-loop input design that can be solved via a linear or quadratic program problem with linear constraints on the input and output signals. Although input design is powerful for motion planning, the design of the shaped input and its input/output constraints may not be robust to parametric uncertainties in the model. This paper shows how the tools of input design can be extended to robust input design by application of the same convex optimization tools to a multi-model input design, in which the models represent the extreme cases of the variations in the dynamics due to the parametric uncertainties. Under the conjecture that extreme models represent the worst-case scenario for input shaping, constraints on the single robust input shaping and the multiple outputs of the extreme models can be enforced to find a robust input shape that can handle parametric uncertainties. The example illustrates the development of a non-intuitive robust input shape that can handle the possible dynamic variations in the oscillatory dynamics of the model. Future work may approach the analysis of uncertainties in the input-shaping problem, by restructuring the problem as a close-loop system.

7.7 Acknowledgement

This chapter, in part, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Robust Input Shaping Design for Uncertain Second-order Dynamics Systems." *2022 CCTA Conference on Control Technology and Applications.* The dissertation author is the first author of this article.

Chapter 8

Robust Input Shaping for Second-order Dynamic Systems with Parameters Variations

8.1 Chapter Abstract

Models of dynamic systems computed from physical equations or system identification include parametric variability on the numerical values of the parameters. For second-order systems, damping and natural frequency are often parameters with a nominal value and a variability range. These models are referred to as parametric uncertain models. Input shaping (IS) is commonly computed as an open-loop exercise for a nominal model, where the possible variations in the parameters are neglected. Parametric uncertain systems require a computation of a robust input signal to work for the set of models derived from the range of the variable parameters. This paper proposes an approach to robust input shaping based on the extreme models, which are derived from the extreme values of the variable parameters via a finite impulse response and a perturbation matrix. The robust

input signal is designed with constraints on both input and output signals and solved as a quadratic program (QP) problem. To verify if the input signal is robust, the outputs for the set of models within the parameter variability are evaluated to find any output signal that violates the constraints; this is proposed as a linear program (LP) problem. A numerical example for the computation of a robust input signal for a second-order system with parametric variability in the natural frequency and damping illustrates the ability of the design.

8.2 Introduction

A model can be described as the mathematical representation, from inputs to outputs, of a system. Models are not an exact match with real systems, and their quality depends on how close their responses are to those of the true plants; a set of models is required to compensate for the deficiencies. To sum up, a model that duplicates the true physical plant behavior can never be computed. It is at this point where the term uncertainty appears, referring to the differences between the models and the real system [10].

Uncertainty can be described in the form of parameter variations and neglected dynamics. Parametric uncertainty in dynamical systems occurs when the parameters describing the systems are known only up to certain extent [81]; a typical example is the variation of a mechanical component, such as a spring where the initial constant value wears out over time. The effects of parametric variability in the system can be evaluated based on a range for the parameters; these calculations reveal the effects of the parametric variability on the system. Approaches for uncertainty go from analysis based in an state-space model and the relationship between the parameter perturbation range and the associated eigenvalue/eigenvector [79] to analysis through the parameters in the

transfer function of the model [82].

In general terms, a variability margin on a model is required for assessing the quality of the description of the dynamic system, aiming to the design of a robust response [83]. Common problems in model representation include: limited data, data containing a degree of uncertainty, unmodeled dynamics (since any dynamical model will neglect some physical phenomena), noise, and unknown initial conditions [75, 76].

In linear time-invariant (LTI) systems, the concept of input shaping (IS) is a technique originally implemented for defining a shaped command input to eliminate or reduce unwanted system vibration [1]. New tools have emerged to solve the input shaping problem [7, 8]. As a common approach, input shaping is utilized in open-loop for multiple-input multiple-output (MIMO) systems in continuous time and discrete time [2, 3, 4]. With the input shaping proposal being inherently open-loop, uncertainties in the open-loop model can lead to system performance degradation [12].

For the most popular methods of input shaping, e.g. zero vibration (ZV), zero vibration derivative (ZVD) and extra insensitive (EI), the evaluation of performance is the residual vibration. The fact that input shaping does not take into account parametric variability has been approached by different methods where further constraints are implemented, such as: weighting the ranges of system parameter values according to the expected modeling errors [13, 14]; allowing the shaper to contain negative impulses [15]; a probabilistic approach of the uncertain model parameters with a polynomial chaos approach [16]; design of closed loop input shapers [17]; just to mention a few approaches. These methods show great improvement, however, most of them are sill based on the ZV, ZVD and EI methods.

Robustness in a system is defined as its qualification to generate the desired output over disturbances and uncertainties. The approach to robust input shaping in this paper differs from most traditional ones. The robust input shaping is formulated as a convex

optimization problem by imposing simultaneous linear constraints on the input and output signals of multiple models that represent the extreme cases of the parametric variability. By approaching the robust input shaping design as a general convex optimization problem, the desired response can be further optimized by defining constraint functions and limits or bounds for these constraints [20, 29]. All constraints are combined in a linear matrix inequality (LMI) [72] which can be solved as either a linear program (LP) problem or as a quadratic program (QP) problem.

Since this input shaping problem is approached as an open-loop problem, the classic robust control solutions are not suitable for this method. Input shaping has proved effective on different types of systems, however, input shaping design methods neglect the modeling errors and parameter variability. In models of real systems, there is some knowledge of the model uncertainties; taking this knowledge into consideration leads to a better shaped input signal [13], since the variability of the model parameters might affect the performance of the output to the input shaping design.

8.2.1 Problem formulation

In this paper, the approach to uncertainty is through parametric variability, that is, the variation of the nominal parameters in the model within a range. The modeling of uncertainty on a linear parametric approach is assumed. The parametric variability is reflected in the form of an impulse response and modeled as a normalized variation weighted by a perturbation matrix. The perturbation matrix allows the normalized parametric variability to be distributed over the different parameters in a discrete-time model that is used for the robust input shaping design.

Robust input shaping is formulated as an open-loop problem with linear constraints on input and output signals; and then solved as a convex optimization problem via QP. The robust input shaping design illustrated consists of the application of the input shaping

problem to obtain a shaped input signal, not only for the nominal model, but also for a number of extreme models. The robustness in the uncertain model is solved by iterating between the definition of extreme models and the computation of outputs outside the bounds. As a first approach, the following conjecture of the extreme models is made: the extreme values in the normalized variation of the parametric variability are used to define the extreme models for which the input signal must be designed; and the robust input signal is computed for these models. As a second step, validation of the robust input signal is solved as an LP problem designed to find if an output that lays outside of the bounds exists. If any output is outside the defined bounds, then the model for that output is considered an extreme model.

The contribution of this paper is a method to analyze the model uncertainty through the impulse response of the variation of the parameters in the transfer function of the discrete-time model, and to compute a single robust input signal that is able to satisfy constraints on both the open-loop input and the desired open-loop output throughout the range of the parametric variability. The input shaping process solved via a QP problem and verified for robustness as an LP problem.

The paper is structured as follows: the model with variable parameters is defined in Section 8.3, along with the representation of the parametric variation and perturbation matrix. Section 8.4 and Section 8.5 elaborate on the robust input shaping process, the linear constraints and the computation and validation of the robust input signal. Section 8.6 illustrates the steps to the robust input signal with a numerical example of a second-order model with two parameters variation.

8.2.2 Notation

The notation defined in Table 8.1 is valid for all sections, considering a multi-input multi-output (MIMO) system. Additional variables are introduced and defined in some

sections.

Table 8.1: Notation.

Symbol	Description
m	number of outputs
p	number of inputs
Q	number of uncertainties
N	number of points
G	model
$\mathbf{u} \in \mathbb{R}_{pN\times 1}$	inputs
$\mathbf{y} \in \mathbb{R}_{mN\times 1}$	outputs
$\underline{y} \in \mathbb{R}_{mN\times 1}$	output lower bounds
$\bar{y} \in \mathbb{R}_{mN\times 1}$	output upper bounds
$\underline{u} \in \mathbb{R}_{pN\times 1}$	input lower bounds
$\bar{u} \in \mathbb{R}_{pN\times 1}$	input upper bounds
$u \in \mathbb{R}_{pN\times 1}$	input signal
$\hat{P} \in \mathbb{R}_{mN\times pQ}$	perturbation matrix
$\theta \in \mathbb{R}_{mN\times pN}$	impulse response coefficients
$\hat{\theta} \in \mathbb{R}_{mN\times pN}$	nominal model
$\delta \in \mathbb{R}_{1\times Q}$	uncertainty vector

8.3 Defining the parametric variability

8.3.1 The Markov parameters

Consider a linear time-invariant continuous-time (CT) causal model $G(s)$, with parametric uncertainty modeled by a finite number Q of independent perturbations. Suppose for each variable parameter $\delta_1, \delta_2, \ldots, \delta_Q$ there is a range described by $\delta_k \in [\delta_k^-, \delta_k^+]$. It is assumed that the parametric variability is represented by a set of models $\mathcal{G}$ with a nominal model $G(\hat{\theta})$, where the nominal parameter is defined as $\hat{\theta}$.

A Zero-Order-Hold discrete-time equivalent of the model $G(s) \to G(q)$ reflects the uncertainty from the original model in the parameters of the polynomial $G(q) = \frac{B(q)}{A(q)}$. Applying an impulse signal as the input, the output signal is a characterization of the

system [84], thus the output of a system can be computed with the impulse response representation. The outputs to impulse signal is the impulse response and are the *Markov parameters* used to describe the system.

$$G(q) = \sum_{k=0}^{\infty} g(k)q^{-k}, \quad \text{assuming} \quad g(k) \approx 0 \quad \text{for} \quad k \geq N,$$
$$G(q) \approx \sum_{k=0}^{N-1} g(k)q^{-k}. \tag{8.1}$$

The parametric variability reflected as the Markov parameters in θ, denotes the *Markov parameter vector*

$$\theta = \begin{bmatrix} \theta(0) & \theta(1) & \theta(2) & \dots & \theta(N-1) \end{bmatrix}. \tag{8.2}$$

With a sequence of the impulse response coefficients θ, the output for the system can be computed as a finite convolution sum by [85]

$$y(\tau) = \sum_{k=0}^{N-1} \theta(k)u(\tau - k), \tag{8.3}$$

where u is the input and y the output. This system might be represented by a matrix formulation as an alternative to the summation notation. Writing out this equation, a linear system of equations $y = \Phi(\theta)u$ is obtained, where Φ is a Toeplitz matrix with the Markov parameters vector. If the input sequence u is of length N_u and the operator signal θ is of length N, the output y is of length $L = N + N_u - 1$ and $\in \mathbb{R}_{L-1 \times N_u}$.

The matrix formulation of the system in (8.3) is

$$
\begin{aligned}
y =& \Phi(\theta)u, \quad \text{where} \\
\Phi(\theta) =& \begin{bmatrix} \theta(0) & 0 & \dots & 0 & 0 \\ \theta(1) & \theta(0) & \dots & 0 & 0 \\ \theta(2) & \theta(1) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \theta(1) & \theta(0) \end{bmatrix}, \\
y =& \begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \vdots \\ y(L-1) \end{bmatrix}, \quad u = \begin{bmatrix} u(0) \\ u(1) \\ u(2) \\ \vdots \\ u(N_u - 1) \end{bmatrix}.
\end{aligned} \tag{8.4}
$$

8.3.2 Linear parametric variation

The set of models $\mathcal{G}$ can be represented by the nominal model $G(q, \hat{\theta})$, with the first N terms of Markov parameters for $\hat{\theta}$. The variability on the N elements in θ constitutes the set of models for which the robust input signal must comply.

The variability on the Q parameters of the model are captured by a linear parametric variation defined by

$$\theta = \hat{\theta} + \hat{P}\delta, \tag{8.5}$$

where $\hat{\theta}$ denotes the $N \times 1$ *nominal Markov parameter vector*, and $\hat{P}$ is the $N \times Q$ *perturbation sensitivity matrix* to model the effect of Q bounded *scalar perturbations* $|\delta(m)| \leq 1$ for

$k = 1, 2, \ldots, Q$ organized in the $Q \times 1$ perturbation vector

$$\begin{bmatrix} \delta(1) \\ \delta(2) \\ \vdots \\ \delta(Q) \end{bmatrix}, \tag{8.6}$$

with the assumption of the DC gain remaining constant during the parameters variations, $\sum_{k=0}^{N-1} \theta(k) = c$ where c is a constant value.

The perturbation matrix $\hat{P}$ in the linear parametric variation (8.5) can be computed via a least squares (LS) solution, if case values θ_k are given for different parameters variations δ_k. Two elements are required: a combinations of variations of δ and the resulting parameters θ for each variation. Each θ is computed via the impulse response in (8.1), with combinations of n different values for each δ, $\Delta = [\delta_1, \delta_2, \ldots, \delta_n]$, where n is the defined number of variations on δ to evaluate.

Defining a new set of θ derived from $\hat{\theta}$ and the set of variations for the Q independent perturbations, the values of the $\hat{P}$ matrix can be computed as

$$\begin{aligned}
&[\theta_1 \quad \theta_2 \quad \ldots \quad \theta_n] \;= \hat{\theta} + P\Delta^T + \varepsilon(P), \\
&[\theta_1 - \hat{\theta} \quad \theta_2 - \hat{\theta} \quad \ldots \quad \theta_n - \hat{\theta}] \;= P\Delta^T + \varepsilon(P), \\
&\text{Defining } \; \Theta = \; [\theta_1 - \hat{\theta} \quad \theta_2 - \hat{\theta} \quad \ldots \quad \theta_n - \hat{\theta}] \;, \text{ we have} \\
&\Theta = P\Delta^T + \varepsilon(P) \;.
\end{aligned}$$

Minimizing the LS error $\mathrm{Tr}\,\{\varepsilon(P)^T \varepsilon(P)\}$ leads to the solution

$$\hat{P} = \Theta\Delta[\Delta^T\Delta]^{-1} \;\in \mathbb{R}_{N\times Q}, \tag{8.7}$$

where $\hat{P}$ is the P that minimizes the error. With the $\hat{P}$ matrix computed, the variation of the model can be estimated directly with the parameters of the impulse response.

8.3.3 Output with Markov parameter perturbations

The matrix $\Phi(\theta)$ is a (linear) matrix operation on the Markov parameters given by

$$\Phi(\theta) = \sum_{k=0}^{N-1} D_k \theta(k) \tag{8.8}$$

in which D_k denotes a $N \times N$ (diagonal) matrix with only 1 elements on the $-k$th diagonal of the matrix. For example,

$$D_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix},$$

$$\cdots \quad D_{N-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$$

The model perturbations modeled by $\theta = \hat{\theta} + \hat{P}\delta$ allow each perturbed Markov parameter $\theta(k),\ k = 1, 2, \ldots, N$ to be written as

$$\theta(k) = \hat{\theta}(k) + \sum_{m=1}^{Q} \hat{P}(k, m)\delta(m), \quad m = 1, 2, \ldots, Q$$

where $\hat{P}$ is the $N \times Q$ perturbation sensitivity matrix in (8.7).

Combining both results allows the output vector y in the presence of parameter

perturbations to be written as

$$
\begin{aligned}
y &= \Phi(\theta) \cdot u \\
&= \sum_{k=0}^{N-1} D_k \theta(k) \cdot u \\
&= \sum_{k=0}^{N-1} D_k \left(\hat{\theta}(k) + \sum_{m=1}^{Q} \hat{P}(k,m)\delta(m) \right) \cdot u \\
&= \left(\sum_{k=0}^{N-1} D_k \hat{\theta}(k) + \sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m)\delta(m) \right) \cdot u \ .
\end{aligned}
$$

With this equation (equality constraint) the robust open-loop input shaping problem can be formulated.

8.4 Robust open-loop input shaping problem

Designing an input $u(k)$ for a desired output $y(k)$ of model is the considered an open-loop input design problem and coined as an *input shaping problem*. In the input shaping process, a shaped input signal is applied to the model with the purpose to obtain the desired output of the system within constraints. The process for the robust input shaping design does not differ from from the input shaping process, but does have a strong addition. The robust input shaping design considers parametric variability for a set $\mathcal{G}$ of discrete-time models with parameter variability modeled by a finite number Q of independent perturbations. The variability of each of the parameters in the model needs to be addressed. Even one parameter varying within a known range represents a challenge to evaluate.

As an initial conjecture, it is considered that by evaluating the extreme models the whole range of variability for the set $\mathcal{G}$ will be included. Typically the extremes are not known, so part of this conjecture is defining these extreme models. The extreme models are

defined by evaluating the extreme values of the variable parameters following the design of the normalized and weighted variation represented by (8.5) and the extreme models in (8.14). The extreme models conjecture is based on the fact that the variation is captured as a linear parametric variation, therefore the extreme models represent the greater distance from the desired response.

A robust performance for a shaped input signal of an uncertain discrete-time model is achieved by designing a single shaped input signal $u(k)$ for the discrete-time models $\in \mathcal{G}$. Utilizing the same shaped input signal for the computation of the outputs $y(k)$ of each model generates outputs within the imposed constraints. The purpose of this process is to obtain one signal for all the different models $\in \mathcal{G}$ that is robust.

The detailed input and output constraints and the feasibility and optimal input design, follow the structure proposed in [82]. As an overview, the following restrictions will be imposed on the control (input) signal $u(k)$ and system outputs $y(k)$.

Consider

- the equality constraint previously defined

$$y = \left(\sum_{k=0}^{N-1} D_k \hat{\theta}(k) + \sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m)\delta(m) \right) \cdot u, \tag{8.9}$$

  with a given $N \times 1$ nominal Markov parameter vector $\hat{\theta}$ and $N \times Q$ perturbation sensitivity matrix $\hat{P}$.

- linear (time varying) inequality constraints on the output vector

$$\underline{y} \leq y \leq \bar{y} \tag{8.10}$$

- linear (time varying) inequality constraints on the input vector

$$\underline{u} \leq u \leq \bar{u} \tag{8.11}$$

- and possible linear (time varying) inequality constraints on the rate of change Δu on the input

$$\underline{u}_{\Delta} \leq \Delta u \leq \bar{u}_{\Delta}. \tag{8.12}$$

Given the constraints above, the robust open-loop input shaping problem can now be formulated as finding a feasible input u such that all equality and inequality constraints above are satisfied over all possible $|\delta(m)| \leq 1$ for $m = 1, 2 \ldots, Q$.

8.5 Robust open-loop input shaping solution

The product of u and $\delta(m)$ in the equality constraint

$$y = \left(\sum_{k=0}^{N-1} D_k \hat{\theta}(k) + \sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m)\delta(m) \right) \cdot u, \tag{8.13}$$

makes the robust open-loop input shaping problem bi-linear in the N variables of the input vector $u(k),\ k = 1, 2, \ldots, N$ and the Q variables $\delta(m),\ m = 1, 2, \ldots, Q$ of the allowable perturbation. To address this problem, the robust open-loop input shaping problem is solved by iterating between the following two steps:

1. Compute a feasible $N \times 1$ solution for the input vector u for a set of fixed extreme values of $\delta(m)$ with $|\delta(m)| \leq 1$ for $m = 1, 2 \ldots, Q$.

2. Compute extreme values of $\delta(m)$ with $|\delta(m)| \leq 1$ for $m = 1, 2 \ldots, Q$ for a fixed $N \times 1$

solution of the input vector **u**.

The main reason to alternate between both steps is that each step can be written as a linear program problem. In addition, a unique and smooth solution for the input vector **u** of Step 1. can be enforced by minimizing the finite-time variance of the rate-of-change of the input and solve via a QP problem. For initialization of the iteration between the two steps, the values of $\delta(m)$ are defined as $\delta(m) = \pm 1$, for $m = 1, \ldots, Q$.

8.5.1 Step 1: robust input shaping design for extreme perturbation values

An extreme model is defined as a model $G(\theta_x)$, for θ_x resulting of extreme values for δ. For Q variable parameters there are to 2^Q extreme models. For initialization of the robust input shaping problem, consider a set of 2^Q extreme Markov parameter vectors

$$\theta_x = \hat{\theta} + \hat{P}\delta_x, \quad x = 1, 2, \ldots, 2^Q \tag{8.14}$$

where the 2^Q uniquely different perturbation vectors δ_x for $x = 1, 2, \ldots, 2^Q$ are chosen via the extreme values $\delta_(m)_x = \pm 1$ of the Q bounded scalar perturbations $|\delta_(m)| \leq 1,\ m = 1, 2, \ldots, Q$.

The solution to the design of single *robust input vector*

$$u_r = \begin{bmatrix} u_r(1) \\ u_r(2) \\ \cdots \\ u_r(N) \end{bmatrix} \tag{8.15}$$

based on extreme perturbation values can be summarized as follows. With the 2^Q extreme

values of δ_x, consider the outputs y_x are given by the equality constraints from (8.10)–(8.12)

$$\begin{aligned} &y_x = \Phi(\theta_x) \cdot u_r, \quad \text{or written out as} \\ &y_x = \left(\sum_{k=1}^{N-1} D_k \hat{\theta}(k) + \sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m) \delta_x(m) \right) \cdot u_r, \\ &\text{for} \quad x = 1, 2, \ldots, 2^Q \end{aligned} \tag{8.16}$$

with a given $N \times 1$ nominal Markov parameter vector $\hat{\theta}$, a given $N \times Q$ perturbation sensitivity matrix $\hat{P}$ and the fixed extreme values for $\delta_x(m) = \pm 1$ for $m = 1, 2, \ldots, Q$.

The linear (time varying) inequality constraints on the output vectors remain

$$\underline{y} \leq y_x \leq \bar{y}, \quad x = 1, 2, \ldots, 2^Q \tag{8.17}$$

and linear (time varying) inequality constraints on the input vector remains

$$\underline{u} \leq u_r \leq \bar{u} \tag{8.18}$$

and possible linear (time varying) inequality constraints on the rate of change on the input

$$\underline{u}_\Delta \leq \Delta u_r \leq \bar{u}_\Delta. \tag{8.19}$$

The robust input shaping problem based on extreme perturbation values can now be formulated as finding a feasible input u_r such that all equality and inequality constraints above are satisfied for all dynamic models given in (8.16). This can be referred as a single-input, multi-output model formulation for which all outputs must satisfy (8.17) for a single robust input shape u_r constrained by (8.18) and (8.19). The linearity of the constraints shown above indicates that the feasibility and design of the robust input shape

u_r can be written as a convex optimization problem. A smooth solution for u_r can be enforced by including a quadratic objective on the rate of change on u_r and compute the solution via a QP problem.

8.5.2 Step 2: computation of extreme perturbations for a robust input design

The availability of a single robust input vector u_r, based on an input design using extreme perturbation values initialized as $\delta_x(m) = \pm 1$, can be used to verify for the actual scalar extreme values of $\delta(m)$ with $|\delta(m)| \leq 1$ for $m = 1, 2, \ldots, Q$. To address this problem, consider

- the single equality constraint

$$y = \sum_{k=1}^{N-1} D_k \hat{\theta}(k) \cdot u_r + \sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m) \cdot u_r \cdot \delta(m),$$

  with a given $N \times 1$ nominal Markov parameter vector $\hat{\theta}$, a given $N \times Q$ perturbation sensitivity matrix P and the given (robust) input vector u_r obtained from Step 1.

- inequality constraints on the *minimal* perturbation

$$\begin{aligned} \delta(m) &\leq z(m) \\ -\delta(m) &\leq z(m) \ , \quad m = 1, 2, \ldots, Q \\ z(m) &\geq 0 \end{aligned}$$

  bounded by the positive slack variables $z(m)$.

- Linear (time varying) inequality constraints

$$\bar{y} \leq y \leq \underline{y} \tag{8.20}$$

that now indicate a *violation* of the original constraints on the output for robust input shaping.

To find the extreme values $|\delta(m)| \leq 1$, an LP problem can be formulated in the variables $\delta(m)$ and $z(m)$, that minimizes the sum of the positive slack variables

$$\sum_{m=1}^{Q} z(m) \tag{8.21}$$

subjected to the linear constraints in (8.20).

Minimization of the linear objective function in (8.21) will make each $z(m)$ as small as possible (but bounded from below by 0) in order for the output to *violate* the original constraints for robust input shaping. If solutions for $z(m) \geq 1$ are found, then the input shape u_r from Step 1 is robust, as it was designed for a set of extreme values for which $|\delta(m)| \leq 1$. If, however, any solution for which $\delta(m) < 1$ is found, the corresponding value for $\delta(m)$ should be added to the set of extreme values and Step 1 has to be revisited for the next iteration in the computation of the robust input shape u_r.

The problem is rewritten with standard LMIs. The defined equality constraint bounded as $\bar{y} \leq y \leq \underline{y}$ can be rewritten as an LMI of the form $A_1\delta \leq b_1$, where

$$\begin{aligned} A_1 &= \begin{bmatrix} \sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m) \cdot u_r \\ -\sum_{k=0}^{N-1} D_k \sum_{m=1}^{Q} \hat{P}(k,m) \cdot u_r \end{bmatrix}, \in \mathbb{R}_{2N \times Q} \\ b_1 &= \begin{bmatrix} \underline{y} - \sum_{k=1}^{N-1} D_k \hat{\theta}(k) \cdot u_r \\ -\bar{y} + \sum_{k=1}^{N-1} D_k \hat{\theta}(k) \cdot u_r \end{bmatrix}, \in \mathbb{R}_{2N \times 1}. \end{aligned} \tag{8.22}$$

The search for δ shall match the range $[-1, 1]$ defined for the robust input shaping design computation. The second condition on the bounds rewritten as an LMI of the form $A_2\delta \leq b_2$, where

$$A_2 = \underset{\in \mathbb{R}_{2Q\times Q}}{\begin{bmatrix} I_Q \\ -I_Q \end{bmatrix}}, \quad b_2 = \underset{\in \mathbb{R}_{2Q\times 1}}{\begin{bmatrix} \mathbf{1} \\ -\mathbf{1} \end{bmatrix}}. \tag{8.23}$$

With the conditions covered as LMIs in the form $A\delta \leq b$, the feasibility for a value of δ can be formulated as the question of

$$\begin{gathered} \exists \delta \in \mathbb{R}_{Q\times 1}, \quad \text{subject to} \quad A\delta \leq b, \quad \text{with} \\ A = \underset{\in \mathbb{R}_{2N+2Q\times Q}}{\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}}, \quad b = \underset{\in \mathbb{R}_{2N+2Q\times 1}}{\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}}. \end{gathered} \tag{8.24}$$

The existence of δ is verified via the slack variable $\mathbf{z} \geq 0$ for all the linear constraints and solve the optimization

$$\min_{z,\delta} \sum \mathbf{z} \quad \text{subject to} \quad A\delta + \mathbf{z} \leq b \quad \text{and} \quad \mathbf{z} \geq 0. \tag{8.25}$$

With the definition of the full variable $\mathbf{x} = \begin{bmatrix} \delta^T & \mathbf{z}^T \end{bmatrix}^T$ the above optimization is written as an LP problem

$$\begin{gathered} \min_{\mathbf{x}} f^T\mathbf{x}, \quad \text{subject to} \\ \begin{bmatrix} A & -I_{2N+2Q} \\ 0_{2N+2Q\times Q} & -I_{2N+2Q} \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} b \\ 0_{2N+2Q\times 1} \end{bmatrix} \quad \text{with} \\ f^T = \begin{bmatrix} 0_{1\times Q} & 1_{1\times 2N+2Q} \end{bmatrix}. \end{gathered} \tag{8.26}$$

8.6 Simulation study

In the previous sections, the process to compute robust input shaping is defined. Those steps will be exemplified with an LTI CT second-order system. The algorithm can be summarized as follows.

Algorithm 1 Computation of robust input signal

```
Apply impulse signal to Ĝ to compute θ̂
Design input signal u_n for Ĝ                    ▷ For illustration purposes
Compute P̂
Fix δ at extreme values                          ▷ Conjecture: δ_x = ±1
while u_r ≠ robust do
    Compute u_r for the extreme models           ▷ via QP
    With u_r, compute the extreme δ values       ▷ via LP
    if |δ| < 1 then
        Add new δ values as extreme
    else if |δ| ≥ 1 then
        u_r ← robust
    end if
end while
```

8.6.1 Input Shaping for the nominal model

For illustration purposes, a second-order model is used for the simplified dynamic analysis and the capability of an oscillatory response to a step input while generating results that are easier to interpret. Consider the linear time-invariant continuous-time causal second-order system given by

$$G(s) = \frac{-K\omega_n^2(\frac{1}{10}s + s)}{s^2 + 2\zeta\omega_n s + \omega_n^2} \ , \tag{8.27}$$

modeling a mass-spring-damper mechanical system, with a force $u(k)$ as an input and a position $y(k)$ as an output, for which a 10% variability in the natural frequency ω_n and a 10% variability in damping ζ are present. The variability of the natural frequency ω_n and

damping ζ is an example of parametric variability.

For illustration purposes, consider the model $G(s)$ with $K = 1/100$, $\omega_n = 10$ and $\zeta = 0.2$ to find a nominal discrete-time model $G(q, \hat{\theta})$ with a sampling time $\Delta_T = 0.05$. A Zero-Order-Hold discrete-time equivalent $G(q, \hat{\theta})$ of the system in (8.27) with zero initial conditions is assumed. The system $G(q, \hat{\theta})$ is represented with the the impulse response system in (8.4).

$$y(k) = \hat{\theta}(k)u(k). \tag{8.28}$$

For the input shaping design is considered a normalized $y_{des}(k)$ of an even number of N samples. The $y_{des}(k)$ is created by the filtering of a constant and equal acceleration and deceleration over $N_s < N$ samples in the middle of the interval of the N samples, where N_s is the number of samples where the transitions of desired values are defined. The filtering to obtain the desired position $y_{des}(k)$ is a discrete-time ZOH equivalent of a double integrator. Output $y_{des}(k)$ can be characterized by

$$y_{des}(k) = \Delta_T^2 \tfrac{q+1}{(q-1)(q-1)} a_{des}(k),$$

$$a_{des}(k) = \begin{cases} 0, & k = 0, \ldots, N/2 - N_s \\ c, & k = N/2 - N_s + 1, \ldots, N/2 \\ -c, & k = N/2 + 1, \ldots, N/2 + N_s \\ 0, & k = N/2 + N_s + 1, \cdots, N - 1 \end{cases}$$

where c is chosen such that $y_{des}(k)$ is normalized to $y_{des}(0) = 0$ and $y_{des}(N-1) = 1$. In the numerical study included here, the values of $N = 100$ and $N_s = 4$ are chosen for a relatively short transition of $y_{des}(k)$ from 0 to 1 in just $2N_s = 10$ samples to illustrate the dynamic effects of $G(q)$ in the input planning.

The property of steady-state output tracking is used to create a quasi-static input

given by $u_q(k) = y_{des}(k)$ that would ignore the dynamics of $G(q, \hat{\theta})$ and serves as a baseline comparison for the input shaping design. For input shaping design, the time-invariant causality is enforced by imposing time-variant amplitude bounds $\bar{u}(k)$ and $\underline{u}(k)$. Results for the nominal input signal $u_n(k)$ are summarized in Figure 8.1 where it can be observed that $u_n(k)$ guarantees that the output remains within the narrow bounds after the transition period.

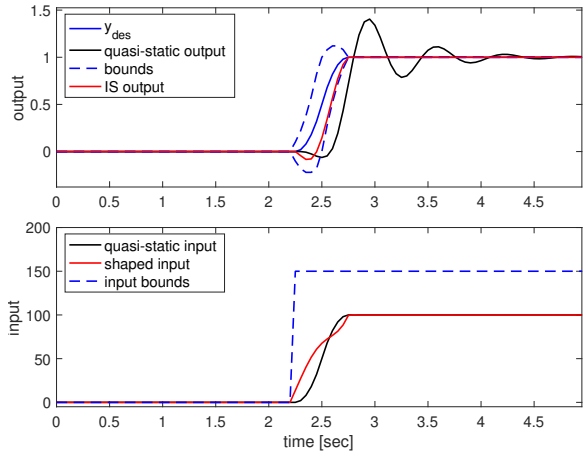


Figure 8.1: Comparison between the quasi-static input and input shaping for the nominal model G. Desired output y_{des} (solid blue line), bounds (dashed blue line), quasi-static signals (solid black line), shaped input signal (solid red line).

8.6.2 Computation of the perturbation matrix

Given the N points selected for this model the nominal parameter $\hat{\theta} \in \mathbb{R}_{N\times N}$. δ_1 represents the changes for natural frequency ω_n and δ_2 represents the changes for damping ratio ζ. Selecting 50 linearly spaced values for $\delta_1(k)$ and $\delta_2(k)$ in the range from -0.1 to

0.1 to perturb $\omega_n = 10 \cdot (1 + \delta_1(k))$ and $\zeta = 0.2 \cdot (1 + \delta_2(k))$, respectively, Δ is given by

$$\Delta = \begin{bmatrix} \delta_1(1) & \delta_2(1) \\ \vdots & \vdots \\ \delta_1(1) & \delta_2(50) \\ \delta_1(2) & \delta_2(1) \\ \vdots & \vdots \\ \delta_1(2) & \delta_2(50) \\ \vdots & \vdots \\ \vdots & \vdots \\ \delta_1(50) & \delta_1(1) \\ \vdots & \vdots \\ \delta_1(50) & \delta_2(50) \end{bmatrix}, \in \mathbb{R}_{2500\times 2}. \tag{8.29}$$

Computing the N impulse response parameters $\theta(k)$ organized in the matrix

$$\Theta = \begin{bmatrix} \theta_1 - \hat{\theta} & \theta_2 - \hat{\theta} & \cdots & \theta_{100} - \hat{\theta} \end{bmatrix} \in \mathbb{R}_{100\times 2500} \ , \tag{8.30}$$

can be illustrated that the parameter perturbations in θ can be computed with (8.5) with $|\delta_1| \leq 1$, $|\delta_2| \leq 1$ and a perturbation matrix $\hat{P}$. A least squares approximation is utilized to compute $\hat{P} \in \mathbb{R}_{100\times 2}$ as shown in (8.7).

$\hat{P}$ indicates all parameters in θ are perturbed with the proper scaling to capture the perturbation of the frequency ω_n and damping ζ. It is worthwhile to note that the values $\delta_1 = \pm 1$ and $\delta_2 = \pm 1$ in (8.5) will indicate the extreme values of the 10% frequency and 10% damping variations.

8.6.3 Robust input shaping for the extreme models

It is not defined what the extreme models are, so as an initial conjecture, the extreme models θ_1, θ_2, θ_3 and θ_4 are defined from the extreme values of the natural frequency ω_n and damping ζ. This is reflected as the extreme values of $\hat{P}$, computed with $\delta_1 = \pm 1$ and $\delta_2 = \pm 1$, in order to define the extreme models, as per (8.14)

$$\begin{aligned} \theta_1 &= \hat{\theta} + \hat{P}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \theta_2 = \hat{\theta} + \hat{P}\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \\ \theta_3 &= \hat{\theta} + \hat{P}\begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \theta_4 = \hat{\theta} + \hat{P}\begin{bmatrix} -1 \\ -1 \end{bmatrix}. \end{aligned} \tag{8.31}$$

Figure 8.2 illustrates the θ coefficients for each of the extreme values.

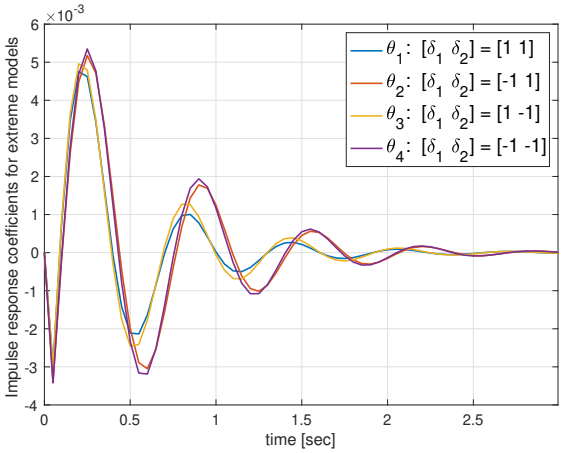


Figure 8.2: Comparison between the extreme values of θ.

Following with the model on (8.31), with $Q = 2$ variable parameters, the extreme models have different impulse response coefficients each represented by the values of $\hat{\theta}$, θ_x and the $\hat{P}$ matrix. Following Step 1 summarized in Section 8.5.1, a total of $2^Q = 4$ extreme

input/output maps are then defined by

$$\begin{aligned} y_1 &= \Phi(\theta_1)u \ , \quad y_2 = \Phi(\theta_2)u \ , \\ y_3 &= \Phi(\theta_3)u \ , \quad y_4 = \Phi(\theta_4)u. \end{aligned} \tag{8.32}$$

similar to (8.16). The variation of the output of the extreme models while applying the shaped input signal computed for the nominal model, u_n, is illustrated in Figure 8.3. The response of the extreme models to u_n, is not within the constraints $(\bar{y},\ \underline{y})$ defined for the output.

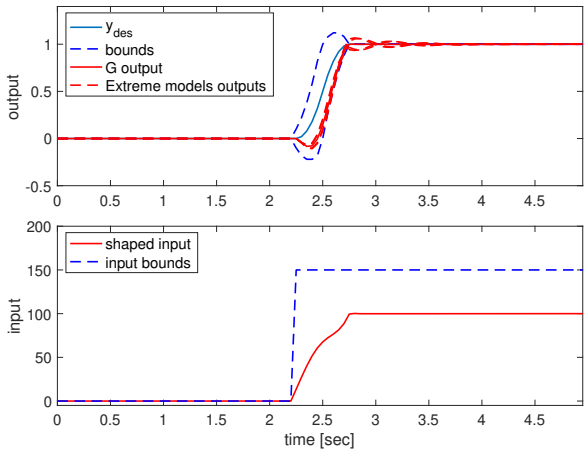


Figure 8.3: Comparison between input signal u_n for the nominal values of $\hat{\theta}$ applied to the extremes models y_1, y_2, y_3 and y_4. Desired output y_{des} (solid blue line), bounds (dashed blue line), shaped input/output signals (solid red line), extreme models (dashed red line).

The input shaping process is computed a second time considering the extreme models. The new input shaping design signal is computed for a new model $y_r = [y_1 \ \ y_2 \ \ y_3 \ \ y_4]$, this illustrates that the process for robust input shaping is the same as the regular process for input shaping, in this case for two or more (extreme) models. The linear input constraints and desired output remain the same, while having a different number of transfer functions. To estimate the shaped input signal for the extreme models the constraints on the output are wider in order to allow the output of the model to fit, but it is not as wide as their

initial oscillation on Figure 8.3. The shaped input signal computed for the extremes models and the response is illustrated in Figure 8.4, where it can be appreciated how the extreme models fit within the new constraints.

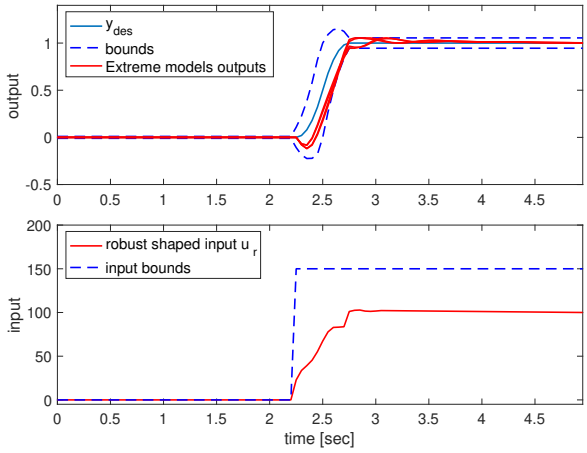


Figure 8.4: Shaped input signal computed for the extremes models y_1, y_2, y_3 and y_4. Desired output y_{des} (solid blue line), bounds (dashed blue line), shaped input/output signals (solid red lines).

As an example, to test the shaped robust input signal for a set of models within the range, an evaluation of 25 linearly spaced variations of δ_1 and δ_2, (with $|\delta| \leq 1$) reflected on the perturbation matrix $\hat{P}$ is performed and utilized to create the equivalent θ for 625 models within the range. In Figure 8.5, the output from the linearly spaced variations of δ and the respective perturbed models is illustrated.

8.6.4 Output verification

To verify the results beyond the plots, the method described in Section 8.5 is utilized to find if exists value for $|\delta_1| < 1$ and $|\delta_2| < 1$ where the constraints are violated.

Solving (8.26) via *linprog* in MATLAB®, the give solution is for (-1, -1). Which means all the outputs are within the constraints. These results prove the conjecture defined at the beginning as correct: defining the extreme models from θ_1, θ_2, θ_3 and θ_4 from the

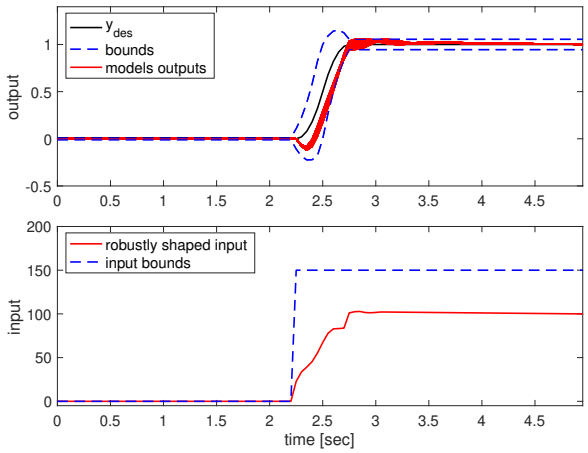


Figure 8.5: Robust input signal outputs for $25x25$ linearly spaced variations of $|\delta_1| \leq 1$ and $|\delta_2| \leq 1$ from (8.5). Desired output y_{des} (solid blue line), bounds (dashed blue line), shaped input/output signals (solid red line).

extreme values of the δ_1 and δ_2 for ω_n and ζ. That might not be the case for all models. If the results were different, e.g. a model θ^* has a value out of the constraint bounds, then the input shaping process is computed one more time with a new extreme model. The definition for the extreme models would be including the model out of the bounds as $y_r = [y_1 \;\; y_2 \;\; y_3 \;\; y_4 \;\; y^*]$. For the new computation, the linear input/output constraints and desired output remains the same, with a new number of transfer functions.

Figure 8.6 illustrates a graphical evaluation of the outputs. The distance from the output of each model to the upper and lower constraint bounds is computed, (8.33).

$$\begin{aligned} \bar{x}_d &= (y(k) - \bar{y}), \quad k = 1, 2, \ldots 625 \\ \underline{x}_d &= (\underline{y} - y(k)), \quad k = 1, 2, \ldots 625 \end{aligned} \tag{8.33}$$

where $\bar{x}$ and $\underline{x}$ represent the distance of each model to the top and bottom constraint bounds. Being $y(k)$ the response of a model, the distance to the (upper and lower) constraint bound shall be $\bar{x}_d \leq 0$ and $\underline{x}_d \leq 0$, where 0 indicates the response is at the boundaries (this is still valid), but a distance > 0, indicates the response $y(k)$ is out of the constraints bounds,

thus is not valid. All cases where $\bar{x}_d > 0$ and $\underline{x}_d > 0$ indicate the models are out of bounds constraints.

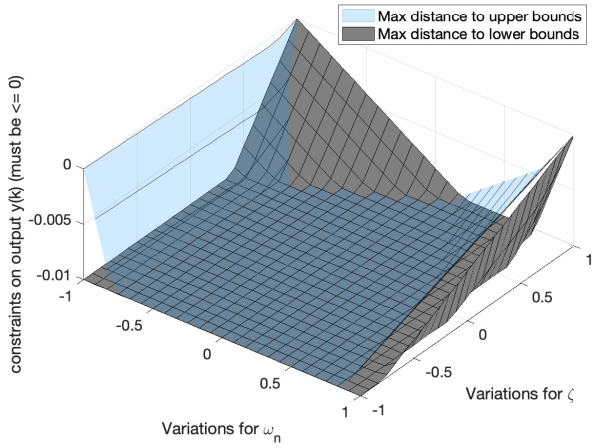


Figure 8.6: Distance to upper and lower bounds for each model as illustrated on (8.33).

8.7 2DOF Example with 3 variable parameters

Similar to the first exercise the with the ECP, two carts with mass $m1$, $m2$, stiffness $k1$, $k2$ and damping $d1$, $d2$. Considered parametric variability on m_1, m_2 and k_2, Figure 8.7 and Figure 8.8.

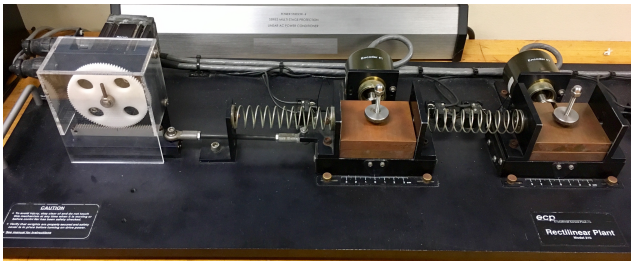


Figure 8.7: ECP Model 210 two-degree of freedom mechanical system.

The input shaping is computed for the nominal model of the 2dof system.

For this example there are $Q = 3$ uncertainties, thus $2^Q = 8$ extreme models.

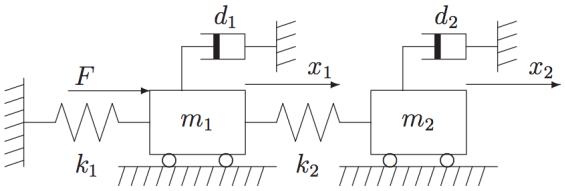


Figure 8.8: Two mass-spring-damper diagram for the ECP 210.

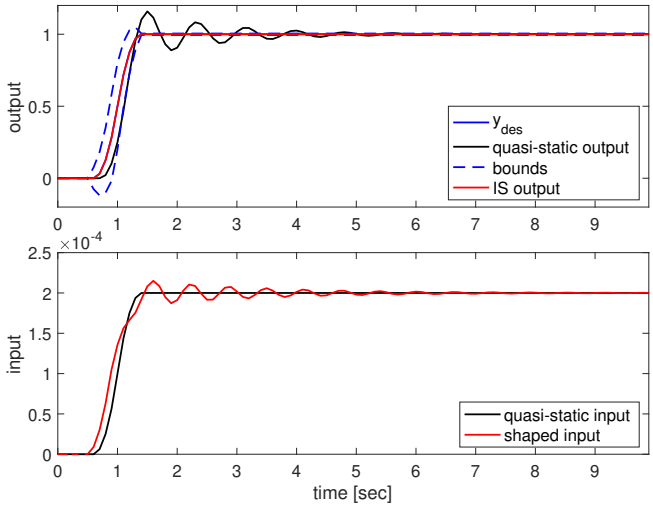


Figure 8.9: Comparison between the quasi-static input and input shaping for the nominal model G. Desired output y_des (solid blue line), bounds (dashed blue line), quasi-static signals (solid black line), shaped input signal (solid red line).

First, the perturbation matrix is computed as in (8.7). For illustration purposes, the impulse response, Figure 8.10, and the step response in Figure 8.11, are modeled via the perturbation matrix.

The input shaping for the nominal model is appliedo to the extreme models, for illustration purposes.

The input shaping process is computed a second time including the extreme models. This illustrates that the process for robust input shaping is the same as the regular process for input shaping, in this case for eight (extreme) models.

To illustrate the results with the shaped robust input signal, a set of models within

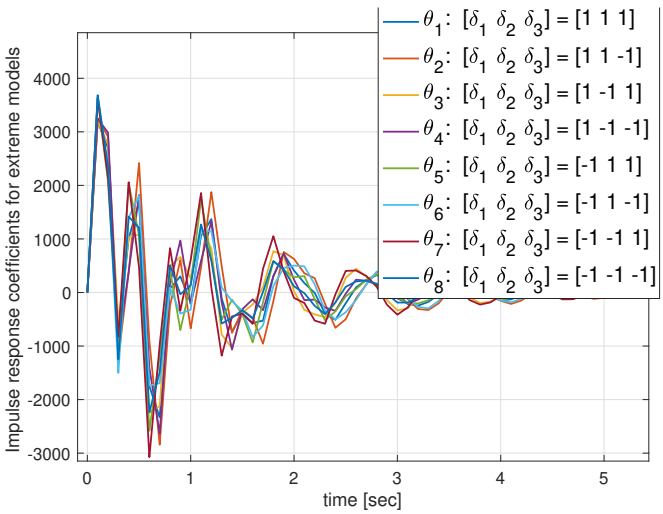


Figure 8.10: Impulse response of the extremes modeled via perturbation matrix.

the range of δ_1, δ_2 and $\delta_3$is evaluated with the reflection on the perturbation matrix $\hat{P}$. Figure 8.14 illustrates the output from the variations of δ and the respective perturbed models.

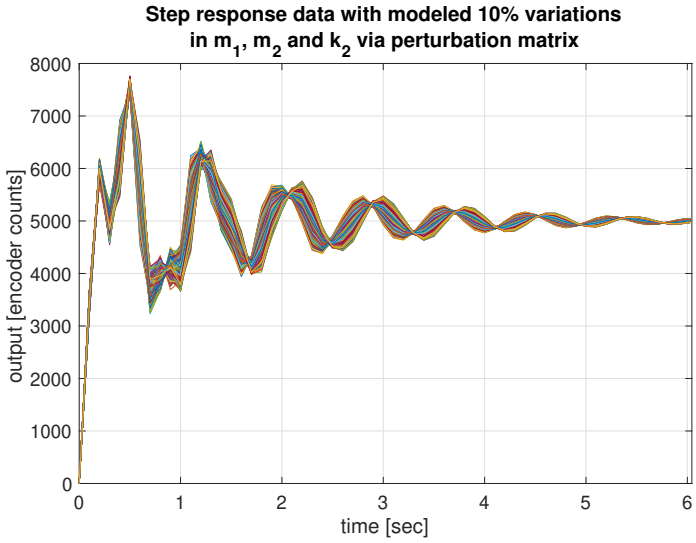


Figure 8.11: Step response modeled via perturbation matrix.

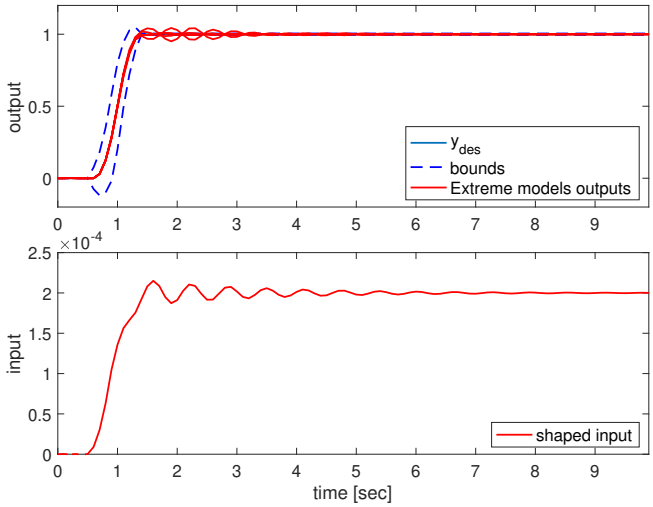


Figure 8.12: Comparison between input signal u_n for the nominal values of $\hat{\theta}$ applied to the extremes models $y_1, y_2, \ldots, y_8$. Desired output y_{des} (solid blue line), bounds (dashed blue line), shaped input/output signals (solid red line), extreme models (dashed red line).

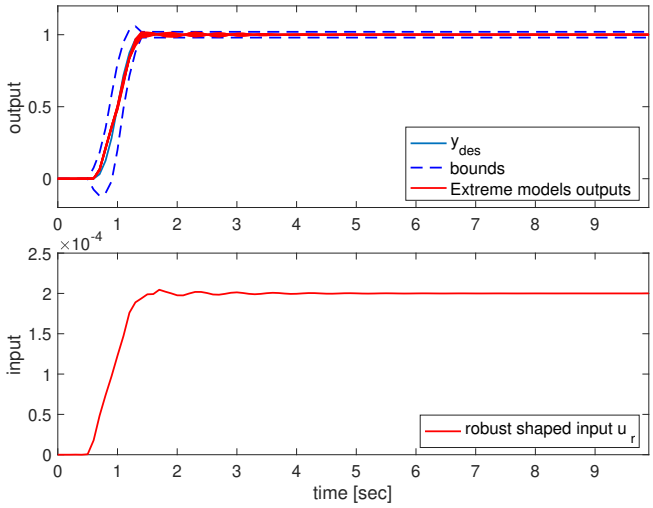


Figure 8.13: Shaped input signal u_r computed for the extremes models $y_1, \ldots, y_8$. Desired output y_{des} (solid blue line), bounds (dashed blue line), shaped input/output signals (solid red lines).

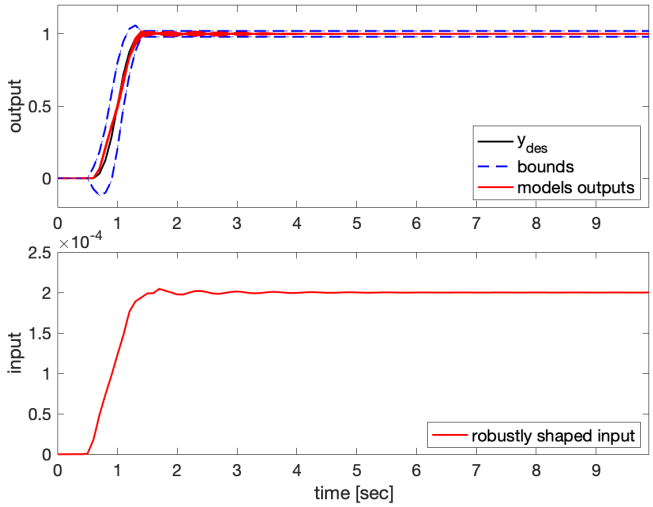


Figure 8.14: Robust input signal u_r applied to linearly spaced variations of $|\delta_1| \leq 1$ and $|\delta_2| \leq 1$ and $|\delta_3| \leq 1$. Desired output ydes (solid blue line), bounds (dashed blue line), shaped input/output signals (solid red line).

8.8 Conclusions

Input design is a powerful control technique for motion planning. However, the design of a shaped input signal in compliance with the input/output constraints may not be robust in the presence of parametric variability in the model. This paper proves how the tools of input shaping design can be extended to robust input shaping during the design process by adhering the extreme values of the parametric variability as extreme models. Model uncertainty is represented via the impulse response, which provides a direct approach to the parametric variability of the model. The uncertainty captured in the impulse response allow a linear parametric distribution model, where the parametric variability can be weighted within the full range. The robust input shaping design follows the conjecture that extreme models represent the worst output from a model, but this can be reevaluated. The defined constraints for a nominal model are relaxed and the extreme models are adhered to the single input shaping design. With the shaped input signal computed, the validation process verifies that the outputs are within the bounds. If this is not the case, the model out of the bounds is added as an extreme model, and the computation is performed again. Following these two steps, a robust input signal is derived. The two-steps process allows the tracking of the parametric variability and evaluation of the extreme in the model, however, this iterative process could be approached as a problem with bi-linear constraints or perhaps as a robust programming for future approaches. In the same way the notion of input shaping can be formulated as an input design that can be solved via a QP problem with linear constraints on the input and output signals, the robust input shaping design is able to find one input signal for multiple models, utilizing the same convex optimization tools for a multi-model input design.

8.9 Acknowledgement

This chapter, in part, has been submitted for publication, A. Dautt-Silva and R. A. de Callafon (2022) "Robust Input Shaping for Second-order Dynamic Systems with Parameters Variations." The dissertation author is the first author of this article.

Chapter 9

Conclusions

Input shaping is a powerful tool for solving open-loop control problems, however, it is an often overlooked alternative. This could be because IS traditional techniques solve the problem with complicated methods that produce an input signal with a significant margin for improvement. Attempting to improve the input signal with the traditional techniques results in more complicated problems.

This research initially aimed to solve the input shaping problem in an optimal way. Computing the input signal via convex optimization with linear constraints on the input and output signals, provides a better response compared to the responses obtained from most of the traditional techniques. However, with the input shaping schemes being inherently open-loop, the process in this research evolved to compensate for the uncertainties in the model, to avoid system performance degradation.

The aim of this research then focused on robustness in the open-loop input shaping problem, considering a set of discrete-time models derived from parametric uncertainty. A robust input shaping design provides an input signal that is guaranteed to hold the output signal for the nominal model as well as for all the uncertain models within the set $\mathcal{G}$; thus the input signal is said to be robust. The key for the robust input shaping design is a

conjecture for approaching all the models within the range by defining the extreme cases. The perturbations on the models are covered via the extreme models and the input signal computed for the extremes is verified for robustness. Robustness for input shaping results in the performance holding for all $\mathcal{G}$, while applying the computed input for the extreme models.

The first task on the uncertainty problem is defining the best way to approach it. Two approaches are provided here for the uncertainty analysis. One is directly working over the variation of the coefficients of the transfer function. The second one is analyzing the variation of the parameters reflected on the impulse response. Then in both cases the analysis is defined based on the range of the family of models $\mathcal{G}$.

The significance of this research is the solution to the input shaping problem for a model with parametric uncertainties with a robust response. Throughout each step of this research new challenges were found that defined the final results.

As a starting point, different input shaping techniques were analyzed for this research, going from inverse modeling and zero vibration to FIR filters. As a first result we found that the convex optimization approach, via LP and MILP methods, provided a superior input shaping signal with better settling time and less overshoot. Different simulation models were approached to evaluate the results with LP method: a laser system, a mass-spring-damper system, and a single-mirror system.

The results proved great response and reduction of the RMS of the tracking error. However, the input-shaping approach could not compensate for errors derived from model uncertainties. One exercise in particular, the dual-mirror system, showed how the plant uncertainties could be compensated via hardware. While the dual-mirror system is a common practice in optical systems, the hardware compensation is not viable most of the time.

A new approach to reduce the negative impact of uncertainties consisted of a closed-

loop system. The problem did create a more robust solution, however, the solution was not causal. It was found that when working the input shaping problem in closed-loop, causality has to be reinforced via a time-dependent input bound signal.

Moving forward, a new improvement was found for the input shaping process. While the LP solution was valid and within the constraints, the input signal was not smooth, and this is not the best approach for a real system. Considering the rate-of-change of the input signal and working the convex optimization problem via QP solved this drawback. The shaped input was still bounded and within the range, but the new input signal now with a smooth transition.

Although input design is powerful for motion planning, the lack of robustness of input shaping facing parametric uncertainties in the model was still present. A first way to directly address uncertainties was based on the coefficients of the transfer function of the model. Robust input shaping was approached as a multi-model input design. The multiple models represented the extreme cases of the variations in the dynamics due to the parametric uncertainties. This under the conjecture that extreme models represent the extreme values for the uncertainties. The improvement on the result against uncertainties was clear. One single shaped input signal holds valid for all the variations of the model due to the parametric uncertainty.

One more representation of the model uncertainty was analyzed via the impulse response. The uncertainties captured in the impulse response allow a linear parametric distribution model, where the uncertainties can be weighted within the full range. The robust input shaping design follows the conjecture that extreme models represent all the variations in $\mathcal{G}$. The extreme models adhere to the single input shaping design.

With the shaped input signal computed, a validation process was designed to verify the robustness (the output signal within the bounds) for a set of models within $\mathcal{G}$. If the verification fails, an extreme model is missing; thus the computation is performed again

considering this model. If the verification passes, then the shaped input signal is robust. The robust input shaping problem is solved by iterating between the definition of extreme models and the verification of outputs outside the bounds.

The input shaping design problem can be formulated and solved via a QP problem with linear constraints on the input and output signals. In the same way, the robust input shaping design is able to find one input signal for multiple models, utilizing the same convex optimization tools for a multi-model input design.

9.1 Future work

There are still plenty of possibilities for future research on the robust input shaping problem. The are more uncertainty factors to be considered, from the order of the model given for the system to the constraints being uncertain. The two-steps method used to find and verify the extreme models problem could be analyzed as a robust programming problem, given that if fits the criteria, e.g. the robust counterpart of the original LP problem is defined with a set of confidence $\mathbf{U}_i$, which should be *computationally tractable*. An option to consider for the same two-steps method, is defining the problem as bilinear matrix inequalities (BMI) and solve as a semidefinite program (SDP).

Appendix A

Optimal Input Shaping with Finite Resolution Computed from Step-Response Experimental Data

A.1 Appendix Abstract

The purpose of this work is the computation of optimal input signals for output tracking of a dynamic system, where knowledge of the dynamic system is only given in the form of step response-data. The starting point is a (non-optimal) input signal given in the form of a simple step input and the measurement of the resulting output. The computation of infinite precision optimal input signals are based on linear programming (LP) solutions that can be specialized to mixed integer linear programming (MILP) solutions with an integer parametrization to enforce possible finite resolution solutions for the input shaping signals. The approach of input shaping based on step-response data is illustrated on experimental data obtained from a class IV laser system with varying pulse length and a laboratory experiment of a mechanical vibration system. In the latter case, computed

optimal input signals were tested experimentally to verify the effectiveness of the input shaping to reduce residual vibrations.

A.2 Introduction

The evolving demands of modern technologies require the development and use of smaller, cheaper semiconductors which are both high speed and energy efficient. The modeling and analysis of novel techniques for high speed manufacturing of these new devices is motivated by an increase in yield and economical benefits. As many of the lithographic manufacturing processes for integrated circuits require repetitive signals for pulsed lasers and mechanical movement, it is worthwhile to compute optimal driving signals to improve accuracy and speed for these processes.

Obtaining accurate models that can be used for the corresponding analysis of accuracy and speed of the production process is often achieved by experimentation. Systematic dynamic analysis of input-output relationships can be analyzed via popular dynamic model estimation techniques, commonly known as system identification [84]. System identification techniques can produce dynamic models that capture the relevant dynamics of the system and experiments are usually designed that utilize the process input experimental variables.

A common input for the analysis of dynamics systems in lithographic manufacturing processes such as lasers and servo systems is a step-wise driving input. The resulting step-response provided valuable information on the system dynamics and can be used to obtain a dynamic model with a finite order using a realization algorithm [86, 38]. The knowledge obtained on the dynamics in the form of a finite order model can then be used to alter the step input in a favorable way as to minimize residual dynamic effects (e.g. vibrations or under/overshoot) during the manufacturing process [5].

To tune optimal input signals to track a desired output, approaches have been

developed for both "model free" tracking using a form of iterative learning control (ILC) [87] and "model-based" tracking based on various methods of input shaping and model predictive control [1, 88, 89]. Although these approaches provide powerful techniques for robustly learning or computing optimal input signals (iteratively) even for non-linear systems [90], this paper is exploring the possibility to develop input signals with limited resolution without the necessary iterations. In addition, the dynamic model is not used to simply compute an approximate inverse model to formulate the input shaping problem, as the input and its resulting output signal will be subjected to constraints during the input shaping process.

In this paper, the dynamic model obtained from step-response experiments will be combined with convex optimization techniques [20] to enforce both input/output constraints during the input shaping computations. Furthermore, mixed integer linear programming (MILP), in which at least one of the optimization variables is restricted to integers [21, 22], can be used to parametrize and enforce the input shaping signal to be represented in a finite resolution. Such finite resolution is important when input signals have to be implemented via a high speed digital-to-analog converter with a limited bit resolution.

The process and computations to go from step-response data to a dynamic model and its optimal input shaping signal subjected to constraints and/or finite resolution is summarized in this paper. Section A.3 provides motivation for this process based on experimental data from both a laser system with varying pulse length and a laboratory experiment of a mechanical vibration system. The realization algorithm to obtain a linear model is summarized in Section A.4, followed by the input shaping calculations in Section A.5 with (experimental) verification in Sections A.6 and A.7.

A.3 Experimental data

A.3.1 Two-mass system experiment

A two-mass system is used for verification and interpretation of the input shaping results presented in this paper. Since the dynamics of a two-degree of freedom mechanical system are well understood, the input shaping results can be interpreted and verified in light of the need to move a mechanical system with two inherent vibration modes as fast a possible from an initial to a final position.

The mechanical system in Figure A.1 can be moved and positioned by a time dependent control force, denoted by $u(k)$ where k refer to the discrete-time index $t_k = k\Delta_t$ in which the sampling time Δ_t is assumed to be constant. The control force $u(k)$ is applied to the first mass m_1 and the objective is to change the position $y_1(k)$ and $y_2(k)$ respectively of both masses m_1 and m_2 to a desired end position y_1 and y_2 as fast as possible via the computation of an optimal input force $u(k)$.

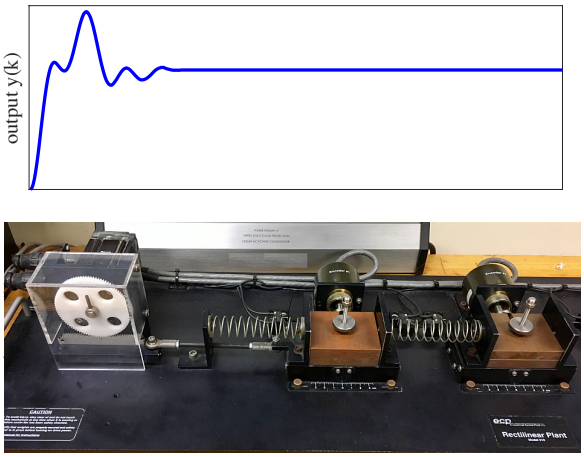


Figure A.1: Step-response position data $y(k) = y_1(k)$ of the mass m_1 in encoder units (top figure) for the ECP model 210 two-degree of freedom mechanical system with mass m_1 in center of bottom figure.

For the dynamic analysis, a step input force $u(k)$ is first applied to the mechanical system and measurements are taken of $y_1(k)$. The experimental data $y_1(k)$ is also shown in Figure A.1. It can be observed that the mechanical system indeed exhibits an undesirable

oscillation when a simple step-based input is applied. Without explicit knowledge on mass, spring or damping coefficients of the two-degree of freedom mechanical system, the experimental data in Figure A.1 will be used as a starting point for the computation of an optimal input force signal $u(k)$ with or without finite resolution.

A.3.2 Laser pulse length experiment

In the semiconductor industry, to expose good wafers it is necessary to control the amount of extreme ultraviolet (EUV) light $y(k)$ generated by a laser as a function of the time index k. The amount of EUV $y(k)$ generated is typically proportional to the energy $y(k)$ delivered by the laser, but may exhibit undesirable dynamic behavior. The experimental data provided here was aimed to identify a means to control the amount of EUV $y(k)$ using the laser excitation signal $u(k)$ dynamically.

Again, step-response experiments are used to characterize the dynamic response of the laser. Due to the noisy measurements of laser energy content, repeated step-response experiments are performed and the data is averaged to produce one set of step-response data. Furthermore, there may be a significant difference in laser energy dynamics when the energy is increased (step up) compared to decreased (step down) and a distinction is made between step up and step down operation. The final averaged step up $y_{up}(k)$ and step down $y_{down}(k)$ step-response data points are summarized in Figure A.2.

The overshoot and undershoot in the step-response data depicted in Figure A.2 are undesirable energy spiking/dipping during the operation of the laser. The goal is to characterize the dynamical properties of the laser pulse at the output of a laser and compute an optimal laser excitation $u(k)$ amplification chain as a function of the input pulse $u(k)$. Without explicit knowledge on laser energy dynamics, the experimental data in Figure A.2 is our starting point for the computation of an optimal and direction dependent input excitation signal $u(k)$ to eliminate energy spiking/dipping.

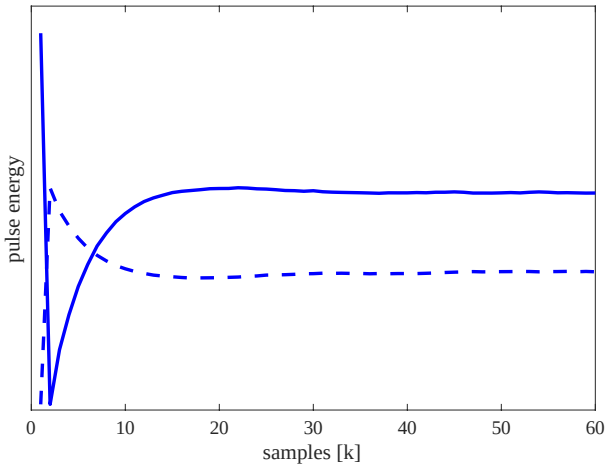


Figure A.2: Step up $y_{up}(k)$ (dashed line) and step down $y_{down}(k)$ (solid line) response data of the laser pulse energy.

A.4 Step-response model estimation

A.4.1 Realization algorithm

The process from step-response data to a linear dynamical model in the form of a state space model

$$\begin{aligned} \bar{x}(k+1) &= A\bar{x}(k) + Bu(k) \\ y(k) &= C\bar{x}(k) \end{aligned}, \quad \bar{x}(k) \in \mathbb{R}^{n\times 1} \tag{A.1}$$

is accomplished with a realization algorithm for step input experiments as presented in [38]. The same process is applied to obtain a dynamic model for both the mechanical and laser data, making a distinction between step up and step down step-response data. A quick summary of the realization algorithm follows to show the albraic results needed to compute a dynamic model (A.1), whereas details can be found in [38].

First, step-response measurements $y(k)$ are stored in a Hankel matrix R starting at $y(1)$. It can be shown [86, 38] that $R = H_Y - \underline{T}_G H_U$, where H_Y is a Hankel matrix with the Markov parameters, H_U is a full rank Hankel matrix with the (unit size) step-response

data and $\underline{T}_G$ is a lower Toeplitz matrix again with the Markov parameters of the dynamic system. The importance of creating the Hankel matrix R with step-response data $y(k)$ is as follows. In case of noise free step-response data, the rank of R is fully determined by the Hankel matrix H_Y of Markov parameters which equals the order n of the dynamic system. Therefore, a low rank n approximation and decomposition of R can be used to determine an nth order model that approximates the measured step-response data.

A low rank n approximation and decomposition of R can be computed using a singular value decomposition (SVD) to obtain

$$R = \begin{bmatrix} U_n & U_s \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & \Sigma_s \end{bmatrix} \begin{bmatrix} V_n^T & V_s^T \end{bmatrix} \tag{A.2}$$

The SVD can be used to find a low rank approximation $R_n = U_n \Sigma_n V_n^T$ of the matrix R and a decomposition $R_n = R_1 R_2$, where $R_1 = U_n \Sigma_n^{1/2}$ and $R_2 = \Sigma_n^{1/2} V_n^T$. The diagonal values of Σ can be used to determine the low rank n approximation and the resulting order n of the model.

A so-called shifted Hankel matrix $\overrightarrow{R}$ again with (noise free) step-response data, but now starting at $y(2)$ [38] allows us to write $\overrightarrow{R} = R_1 A R_2$ where the left inverse of R_1 and the right inverse of R_2 can be computed respectively due to the full column and row rank of the matrices R_1 and R_2. The low rank decomposition $R_n = R_1 R_2$, where $R_1 = U_n \Sigma_n^{1/2}$ and $R_2 = \Sigma_n^{1/2} V_n^T$ in (A.2) can be used to find the state matrix A, but also find the input matrix $B = R_2(:, 1)$ and $C = R_1(1, :)$ in (A.1). From the state-space representations, discrete-time transfer functions can also be calculated.

A.4.2 Realization algorithm for two-mass system experiment

The two-mass model is defined as 4^{th} order model. The transfer function for the model is given by

$$G_{m1}(q) = \frac{13.44q^3 - 25.86q^2 + 12.45q + 0.20}{q^4 - 3.87q^3 + 5.68q^2 - 3.73q + 0.93}. \tag{A.3}$$

The general realization algorithm (GRA) model response is simulated with a step input to compare its response with the measured step-response data from the experiment in Figure A.3.

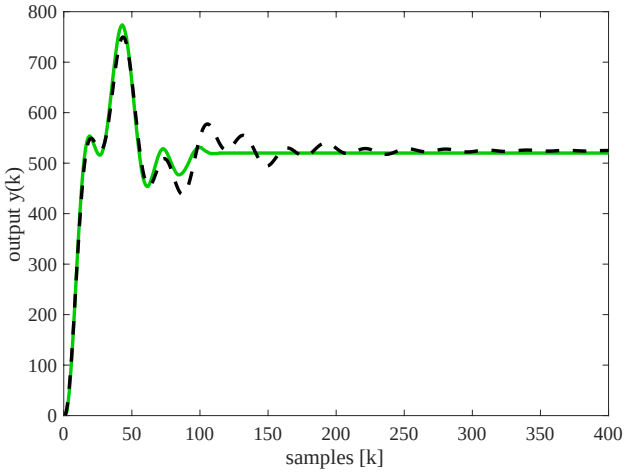


Figure A.3: Measured response vs simulated response. Data from the two-mass system experiment (green solid line), and simulated response from the GRA model obtained (black dashed line).

The comparison in Figure A.3 verifies that the identified model is a reasonably good estimate.

A.4.3 Realization algorithm for laser experiment

Both the step up and step down models are defined as 3$^{\text{rd}}$ order models. The transfer function for the two models are:

$$G_{up}(q) = \frac{0.31q^2 - 0.58q + 0.26}{q^3 - 1.73q^2 + 0.72q + 0.01}, \tag{A.4}$$

$$G_{down}(q) = \frac{-6.51q^2 + 12.25q - 5.75}{q^3 - 1.73q^2 + 0.72q + 0.01}. \tag{A.5}$$

The obtained models of the systems are simulated with a step input to compare their response with the measured step-response data from the experiment. The comparison in Figure A.4 verifies that the identified models are a good estimate.

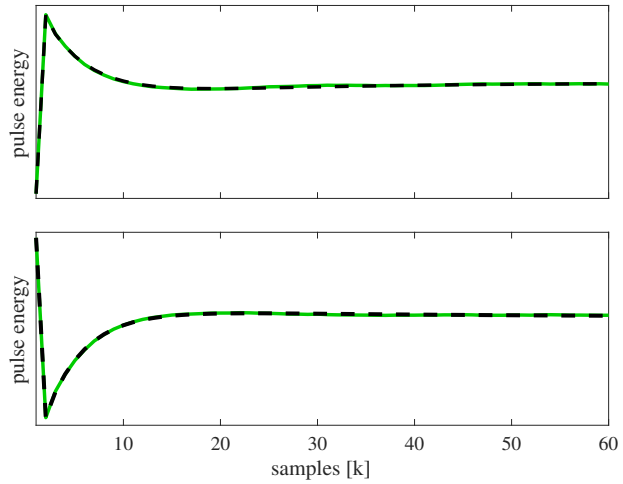


Figure A.4: Measured response vs simulated response. Top: Step up model. Bottom: Step down model. In each plot: data from the laser experiment (green solid line), and simulated response from the GRA models obtained (black dashed line).

It is worth noticing that the models in (A.4) and (A.5) have at least one step time delay, but are stably invertible since the numerators' roots are within the unit circle. Such a property may not hold in general for a model obtained by step response realization. If the model is stably invertible, input shaping can be done by computing input signals $u(k)$ via direct model inversion, e.g. $u_{up}(k) = G_{up}^{-1}(q)y_{up}(k+1)$ and these input shaping signals

will be used as a comparison to the optimal input signals found by linear programming solutions provided in this paper.

A.5 Input shaping techniques

A.5.1 Inverse model approach

Since the models in (A.4) and (A.5) are stably invertible, a straightforward approach for input shaping is to compute a solution via model inversion. Assuming the desired output $y(k)$ is an ideal step signal, we may apply a step input signal with the desired size to the inverse of the model to compute an input $u(k)$ to obtain the fastest possible desired output. This approach can be applied to the dynamic models of the laser data and is provided here as a reference.

A simulation with the models $G_{up}(q)$ and $G_{down}(q)$ confirms the computed input and we are able to get the "ideal" response as shown in Figure A.5.

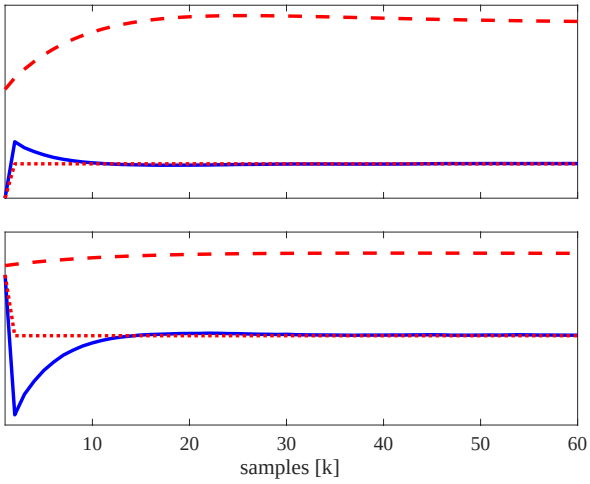


Figure A.5: Inverse model approach. Top: Step up model. Bottom: Step down model. In each plot: data from the laser experiment (blue solid line), ideal response (red dotted line), and computed input with G^{-1} (red dashed line).

Although we can perform a simple inversion on the obtained model to compute an

"ideal" input $u(k)$, the computed input $u(k)$ may not be feasible due to input constraints in size. Modifications of the computed input $u(k) = G(q)^{-1}y(k)$ via model inversion would require iterations on the desired output $y(k)$ to ensure $u(k)$ is feasible.

A.5.2 Convex optimization

Instead of assuming model invertibility or perform iterations to find a desired input $u(k)$, a different approach is to formulate the input shaping as an optimization problem. We look to minimize the maximum error $E = max|y(k) - y_{des}(k)|$ over an optimization horizon of length M, based on a desired trajectory y_{des}. The constraint used is a range for both the input $U_{low} \leq u(t) \leq U_{high}$ and the output $Y_{low} \leq y(t) \leq Y_{high}$. The optimization problem is now expressed as:

$$\begin{aligned} \text{minimize} \quad & E = \max_{k=1,\ldots,M} |y(k) - y_{des}(k)| \\ \text{subject to} \quad & U_{low} \leq u(k) \leq U_{high}, \quad k = 1, \ldots, N \\ & Y_{low} \leq y(k) \leq Y_{high}, \quad k = 1, \ldots, N \end{aligned} \tag{A.6}$$

where M is a defined optimization horizon and $u(k),\ k = 1, \ldots, N$ is the optimization variable with $M \geq N$ to ensure end conditions will be met.

With the optimization variable $y(k)$ linear in E and the linear constraints in (A.6), the optimization can be formulated as a linear programming (LP) problem. The general formulation for this is [20]:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & a_i^T x \leq b_i, \quad i = 1, ..., m, \end{aligned} \tag{A.7}$$

where x is the vector of optimization variables and a_i, b_i and c are parameters and constraint functions.

With knowledge of the numerator coefficients b_j and denominator coefficients a_j of the transfer functions in (A.4), (A.5) for the laser system and (A.3) for the mechanical system, the output *y(k)* is written in a linear regression form

$$\begin{aligned} y(k) &= b_0 u(k-1) + b_1 u(k-2) + ... + b_n u(k-n-1) \\ &- a_1 y(k-1) - ... - a_n y(k-n). \end{aligned} \tag{A.8}$$

Similarly as in the realization algorithm, we can define $y = \Phi u$, where Φ is a Toeplitz matrix with impulse response coefficients

$$A_{ij} = \begin{cases} h_{i-j} & \text{if } \ 0 \leq i-j \leq k, \\ 0 & \text{otherwise} \end{cases}$$

With the linear relationship we can now solve the optimal control problem using standard LP software, such as the Matlab implementation in the form of the function `linprog` [30].

The use of a time varying output constraints $Y_{low}(k)$, $Y_{high}(k)$ has also been illustrated in [26] to enforce a particular settling time $k^\star$ for the input shaping. Although finding a minimal value k^*_{min} is not an LP problem, an LP problem can be used to check the existence and computation of a solution for the input shape $u(k)$ for a given k^*. Subsequently, finding a minimal value k^*_{min} to find a minimum-time input shape $u(k)$ that satisfies all constraints can be found via a bisection method [20] on k^*.

A.5.3 Mixed integer linear programming

The LP problem will provide a solution for the input shape $u(k)$ under linear constraints where each value of $u(k) \in \mathbb{R}$ for $k = 1, \ldots, N$. In many practical applications the input $u(k)$ may not be specified with infinite resolution and a common example is the case where $u(k)$ is generated via an Digital-to-Analog (DA) converter with a limited number

n of bits to allow 2^n values (including 0) in $u(k)$. Simply rounding (a scaled version of) $u(k)$ to the nearest integer value may negatively affect the optimal input shaping solution and re-introduce undesirable dynamic effects on the desired output $y(k)$.

Following the example of a n-bit DA converter, it is beneficial to parametrize $u(k)$ via

$$u(k) = U_{max}/2^n x(k)$$

using integer values $x(k) = 0, 1, \ldots, 2^n - 1$, e.g. $0 \leq x(k) \leq 2^n - 1$. Optimal input shaping would then have to be formulated as a Mixed Integer Linear Programming (MILP) problem in which at least one of the variables is restricted to integers [21, 22]. The input shaping formulated in this paper, where the input $u(k)$ is re-parametrized in integer values, is now solved by formulating a solution to [27].

$$\begin{aligned} &\text{minimize} \quad c^T x \\ &\text{subject to} \quad a_i^T x \leq b_i, \quad i = 1, ..., m, \\ &\text{where} \quad x \in \mathbb{Z}. \end{aligned} \tag{A.9}$$

The same constraints on input $u(k)$ and output $y(k)$ can be used, but optimization is done over the integer values x. Solutions to the MILP problem is again computed with Matlab using the Gurobi Optimizer [91].

A.6 Input shaping for laser experiment

To provide a comparison with existing input shaping methods, first input shaping based on zero vibration (ZV) [1, 18] and finite impulse response filtering (FIR) [19, 4] are applied to the data and models of the laser experiment. Figure A.6 shows the comparison of the results for each method. We can still see room for improvement and in FIR applications

there may be a large delay due to the filtering. The ZV and FIR methods are specifically designed for systems with (multiple) vibrations, but the laser experiment as indicated in Figure A.4 only exhibits fast dynamics with an overshoot with a exponential decay.

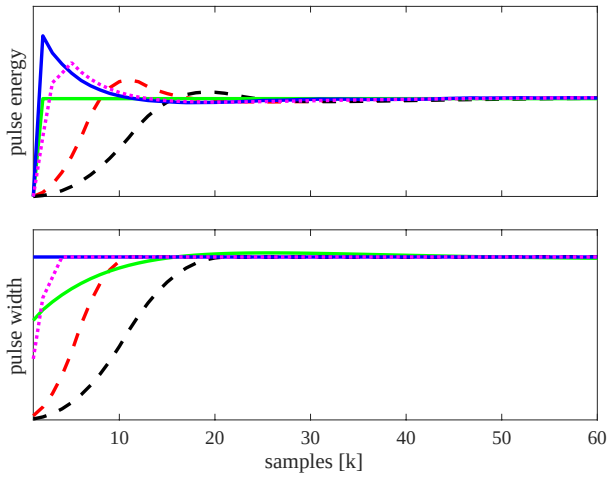


Figure A.6: Step up model. Top: System responses. Bottom: System inputs. In each plot: data from laser experiment (solid blue line), data from G^{-1} (solid green line), data from ZV (magenta dotted line), 10^{th} order FIR filter (red dashed line), and 20^{th} order FIR filter (black dashed line).

Following an approach where we do not have a desired output trajectory vector (as in model inversion), we exercise the LP problem approach with a set of constraints on the output and the input. There are two important parameters to consider when defining output constraints: the output constraint $\overline{y_1}$ and the time k^* when switching from "large" to "small" constraints on the ouput $y(t)$ as a way to enforce a (minimum) settling time. This is clearly reflected in the results depicted in Figure A.7, exemplifying the step up model with an minimal k^* found by bisection and a k^* longer than required for a minimum settling time.

Following the same approach as in convex optimization, we exercise a MILP problem approach to find an input shape $u(k)$ with a finite resolution. A comparison results is shown in Figure A.8 and it can be observed that the results are very similar to those found from the inverse model solution, but now with limited resolution on the input $u(k)$.

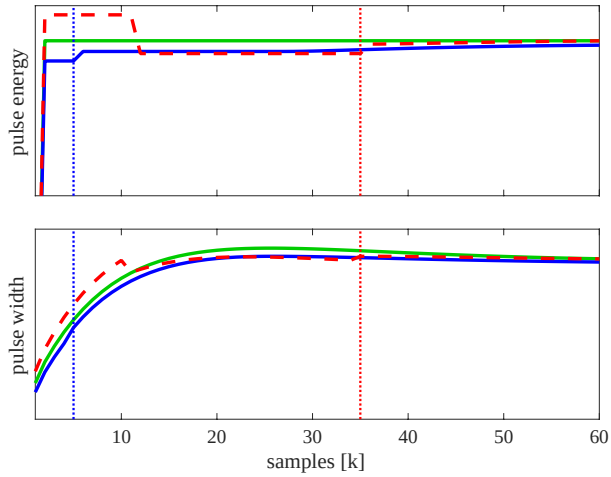


Figure A.7: Optimization for step up model varying k^*. Top: System responses. Bottom: System inputs. In each plot: desired data from inverse model (green solid line), optimization with optimal k^* (blue solid line), and optimization with long k^* (red dashed line).

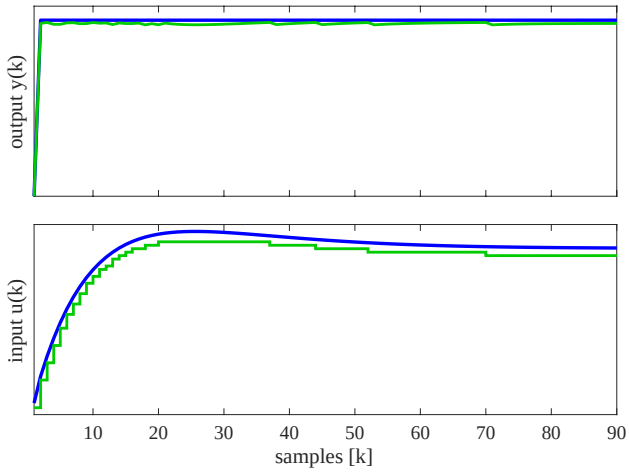


Figure A.8: Computed MILP input and simulated response for step up model. Top: System responses. Bottom: System inputs. In each plot: desired data from inverse model (blue solid line), and MILP (green solid line)

A.7 Input shaping for mechanical system

To show the power of both the LP and MILP approach for input shaping for the mechanical system example, no desired output trajectory vector (i.e. y_{des}) is used. Instead, only constraint on the output and input are used to compute a input shape $u(k)$ by varying

the value of $k^\star$ to influence the settling time. In the LP approach the input $u(k) \in \mathbb{R}$, whereas in the MILP the input $u(k)$ is restricted to 256 levels to resemble an $n = 8$ bit DA converter.

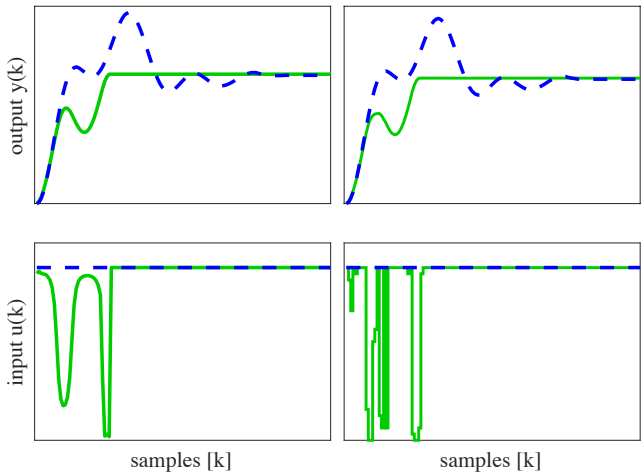


Figure A.9: Optimization for Two-Mass system. Top left: output response for LP optimized input, bottom left: LP optimized input. Top right: output response for MILP optimized input, bottom right: MILP optimized input where original step response is shown in dashed lines in the figures.

Figure A.9 shows the LP and MILP optimization results for the Two-mass system, compared with the original step response data. The overshoot also observed earlier in the step-response of Figure A.1 is completely removed and the response settles at (less than) half of the original number of output samples. Furthermore, in order to verify the validity of the input shaping signals experimentally, the computed input signal is re-applied to the ECP mechanical system from which the step-response data was obtained. The resulting experimental results are shown in Figure A.10 and it can be verified that the output response accurately tracks the predicted optimized output.

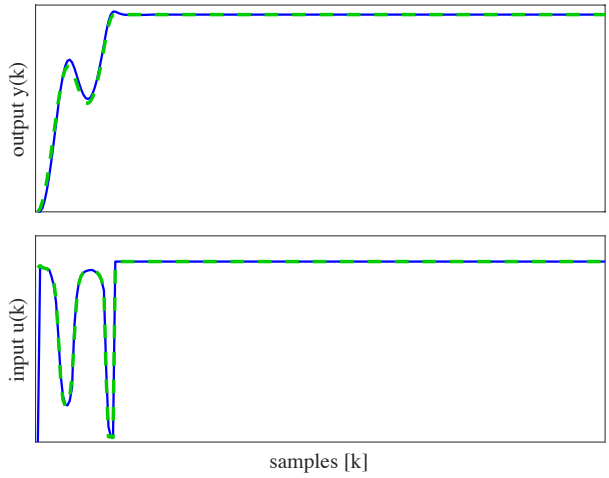


Figure A.10: Comparison of ECP experimental results and computed optimized input shaping results (outputs in top figure, inputs in bottom figure). In each plot: calculated optimization data (green solid line), and ECP experimental data (blue dashed line).

A.8 Conclusions

Starting from a measurement of a step-response data, the input shaping methodology presented in this paper formulates an optimal input to minimize settling times in a dynamic system via a two-step procedure. The first step is the formulation of a linear dynamic model on the basis of the step-response data via a realization algorithm. The second step computes an infinite precision optimal input signal on the basis of a linear programming (LP) problem that can be specialized to mixed integer linear programming (MILP) solutions to enforce a finite resolution on the input shaping signal. The approach is demonstrated on both laser pulse data and the motion of a mechanical system to illustrate the improvements in settling time for typical lithographic manufacturing processes for integrated circuits that require repetitive signals.

A.9 Acknowledgment

The authors would like to acknowledge and thank CYMER, an ASML company, for the use of the laser Pulse Data from lasers Technology Development Testing.

This appendix, in full, is a reprint of A. Dautt-Silva and R. A. de Callafon (2018) "Optimal Input Shaping with Finite Resolution Computed from Step-Response Experimental Data." *2018 Annual American Control Conference (ACC)*, pp. 6703-6708. The dissertation author is the first author of this article.

Appendix B

Application of Dynamic Input Shaping for a Dual-Mirror System

B.1 Appendix Abstract

The task of trajectory planning for a dual-mirror optical pointing system greatly benefits from carefully designed dynamic input signals. This paper summarizes the application of multivariable input shaping (IS) for a dual-mirror system, starting from initial open-loop step-response data. The optical pointing system presented consists of two Fast Steering Mirrors (FSM) for which dynamically coupled input signals are designed, while adhering to mechanical and input signal constraints. For the solution, the planned trajectories for the dual-mirrors are determined via (inverse) kinematic analysis. A linear program (LP) problem is used to compute the dynamic input signal for each of the FSMs, with one of the mirrors acting as an image motion compensation device that guarantees tracking of a planned trajectory within a specified accuracy and the operating constraints of the FSMs.

B.2 Introduction

A fast steering mirror (FSM) is especially useful in image motion stabilization as it can be used to alter the optical pathway for beam controls [47, 48]. The resulting challenge of dynamic pathway planning becomes compounded when two FSMs are used in sequence to increase motion range in a so-called dual-mirror configuration [49, 50]. The motivation of the study in this paper is based upon a two voice coil motor (VCM) actuated FSM with a flexure suspension, organized in the dual-mirror configuration of Figure B.1. It is worthwhile to mention the mechanical vibrations and (dynamic/static) cross coupling between the rotational axes of the FSM. The application of a step-wise voltage input signal on one of the VCMs in the FSM results not only in mechanical vibrations of the angular rotation of the FSM around one of its main orthogonal axis; it also creates a dynamic coupling in the form of vibrations around the other orthogonal axis.

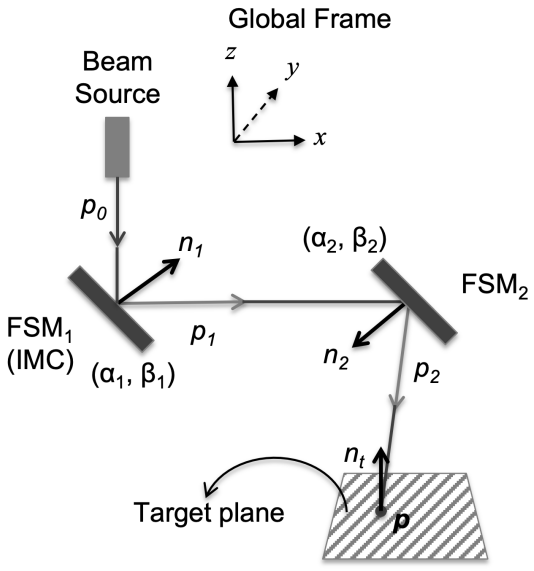


Figure B.1: Mechanical layout of a dual fast steering mirror system with FSM_1, FSM_2 and a beam source.

The purpose of this technical brief is to illustrate the application of a method

for dynamic trajectory planning of a dual-mirror optical pointing system (as illustrated in Figure B.1) that guarantees that optical signals are projected within a pre-specified accuracy on a target plane. The proposed solution to address the task of dual-mirror trajectory planning is a combination of dynamic modeling and multivariable optimal input shaping (IS) that takes into account both the dynamics and the kinematic coupling between the FSMs, as well as constraints on the input signal that can be applied to the FSMs. The IS is applied to both multivariable mirrors in the system, where the second mirror compensates for the error in the target. The definition of constraints of the input signals and bounds on the desired response are written as linear matrix inequalities (LMI), which can be solved as a linear program (LP) problem.

For this application, input shaping is formulated as an open-loop problem, with linear constraints on input and output signals imposed to formulate a LP problem to find optimal input profiles. Although a precise decoupled mechanical design of the FSMs provides the opportunity of performing two independent IS problems, realistically a multi-input, multi-output approach is required [47]. The proposed procedure covers from multivariable step-response to feasible multivariable input shape, while working with convex optimization to compute the dynamic input signal that guarantees tracking of a planned trajectory within a specified accuracy and the operating constraints.

The method of data-based IS for a dual-mirror system proposed in this paper consists of three steps. First, step-response data is used to estimate a dynamic model of each FSM via a realization algorithm [38, 54] that captures the coupled dynamics of the two-axis actuation based on the realization algorithm described in [38]. Second, via (inverse) kinematic analysis the planned trajectories for the dual-mirrors are determined. Third, the problem of dynamic IS based on the planned trajectories for the dual-mirrors is solved as a LP problem similar to [42], but extended to the multivariable dynamic model of the FSMs, under constraints on size and rate-of-change on the input signal to find a

feasible solution. The LP result for one FSM [55] shows room for improvement if a second mirror is to be used as an image motion compensation (IMC) device.

B.3 Mirror rotation dynamics

B.3.1 Step-response realization

The process of estimating a multivariable discrete-time model from step-response data is accomplished with a step-response realization algorithm [38, 54]. In particular for a FSM that exhibits (dynamic) coupling between the mirror axes rotation, a multivariable model can be used to model this coupling. For each FSM, the realization algorithm produces a fourth order multivariable discrete-time state-space model

$$x(k+1) = Ax(k) + B \begin{bmatrix} V_\alpha(k) \\ V_\beta(k) \end{bmatrix}, \quad \begin{bmatrix} \alpha(k) \\ \beta(k) \end{bmatrix} = Cx(k). \tag{B.1}$$

Figure B.2 and Figure B.3 are the bode plots produced by the fourth order state space model computed for each FSM. The model has 2 inputs and 2 outputs with the following modes in Table B.1 for each mirror (B.1).

Table B.1: Natural frequency f_n [Hz] and damping ratio ζ.

	f_n	ζ
FSM_1	19.64	6.97e^{-2}, 6.45e^{-2}
FSM_2	13.21	1.50e^{-2}, 1.21e^{-2}

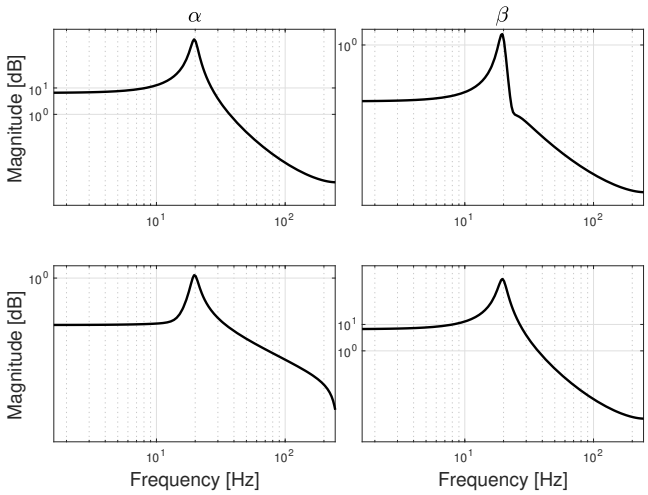


Figure B.2: Bode plot of the fourth order state space model in (B.1) for FSM_1.

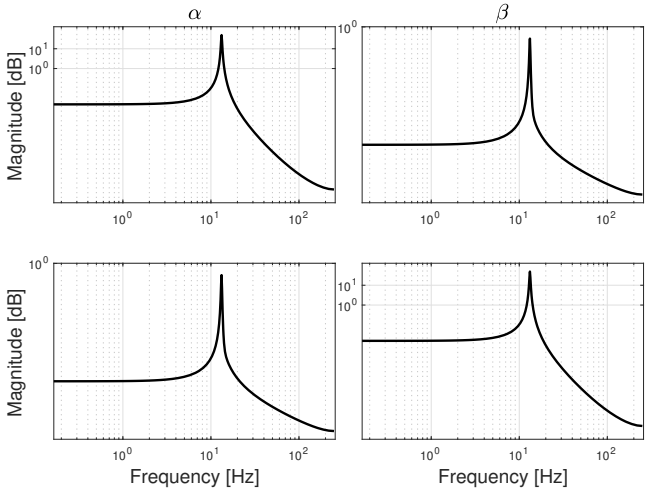


Figure B.3: Bode plot of the fourth order state space model in (B.1) for FSM_2.

B.4 Mirror input shaping

B.4.1 Target parametrization

To illustrate the effectiveness of dynamic input shaping for FSMs, a target trajectory with fast dynamic transitions in the motion is chosen, in form of a square trajectory. The trajectory is parametrized by ϕ following the square shape according to

$$\begin{aligned}(x_p(\phi), y_p(\phi))_1 &= (\underline{x}_p, \overline{y}_p - \phi) \\ (x_p(\phi), y_p(\phi))_2 &= (\underline{x}_p + \phi, \underline{y}_p) \\ (x_p(\phi), y_p(\phi))_3 &= (\overline{x}_p, \underline{y}_p + \phi) \\ (x_p(\phi), y_p(\phi))_4 &= (\overline{x}_p - \phi, \overline{y}_p)\end{aligned} \tag{B.2}$$

for $\phi \leq L$, where $L = 4$ inches.

B.4.2 Inverse kinematics

Given the target trajectory $(x_p(\phi), y_p(\phi))$ parametrized by (B.2), the values for the desired mirror rotation pairs (α_1, β_1) and (α_2, β_2) can be computed as a function of ϕ by inverse kinematic analysis of the dual-mirror system. The inverse kinematics will lead to trapezoidal input signals for both α_2 and β_2, [55]. In this analysis, a number N of points along the parametrized trajectory is chosen to be $N = 200$.

B.4.3 Application of quasi-static input

The angle pairs $(\alpha_2(\phi), \beta_2(\phi))$ are computed as discrete-time dependent angles $(\alpha_2(k), \beta_2(k))$ via $\phi = \phi(k)$ over a time interval $t_k = k\Delta_t, k = 1, 2, \ldots, N$. Tracking the time-dependent angles presents a challenge for the FSM when Δ_t is small and the time interval $N\Delta_t$ is short. This challenge is apparent in Figure B.4, where $\Delta_t = 2$ msec, $N\Delta_t = 400$ msec

and voltage signals are applied to FSM_2 that are simply scaled proportionally to the quasi-static values of $(\alpha_2(k), \beta_2(k))$. It is clear that the vibration dynamics of FSM_2 requires carefully designed voltage input signals.

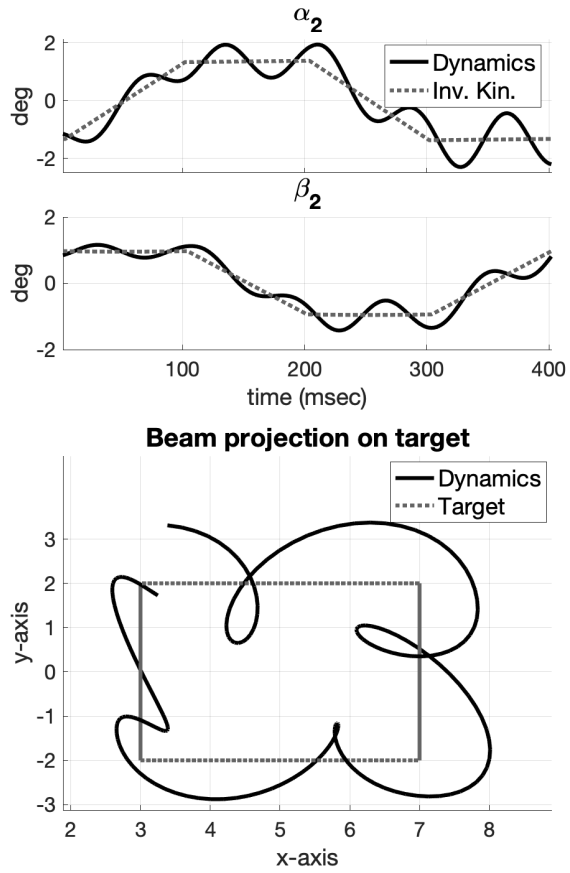


Figure B.4: Dynamic response of FSM_2 with quasi-static inputs. Top figure: comparison of discrete-time dynamic angle pair $(\alpha_2(k), \beta_2(k))$ (solid lines) and desired inverse kinematic angle pair $(\alpha_2(\phi), \beta_2(\phi))$ (dashed lines). Bottom figure: comparison of beam projection on the target plane for $(\alpha_2(k), \beta_2(k))$ (solid lines) and kinematic angles $(\alpha_2(\phi), \beta_2(\phi))$ (dashed line).

B.4.4 Dynamic input shaping

For dynamic input shaping, knowledge of the state space model in (B.1) can be used to constrain the dynamic motion of FSM_2. An approach as the one described in [55]

is followed to define the (linear) motion constraints and formulate the input design as a LP problem [20, 29]. A brief overview is described in this section.

The first constraint is to enforce the resulting output to be close to the desired output via the requirement $|y(k) - y(\phi)| \leq \varepsilon(k)$. The first constraint is formulated as,

$$\begin{array}{rcl} \Phi\mathbf{u} & \leq & \mathbf{y} + \varepsilon \\ -\Phi\mathbf{u} & \leq & -\mathbf{y} + \varepsilon \end{array}, \quad \varepsilon = \begin{bmatrix} \varepsilon(1) \\ \varepsilon(2) \\ \vdots \\ \varepsilon(N) \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix}. \tag{B.3}$$

The second constraint is a limit on the rate-of-change $|u(k) - u(k-1)| \leq d(k)\Delta_t$, to reduce volatility in the voltage input signals $u(k)$. Such a rate-of-change constraint is implemented via $E\mathbf{u} \leq \mathbf{d}$ where E and $\mathbf{d}$ are given by

$$E = \begin{bmatrix} 1 & 0 & -1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & -1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & 1 & 0 & -1 \end{bmatrix}, \ \mathbf{d} = \Delta_t \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(N) \end{bmatrix}. \tag{B.4}$$

The last constraint limits the size of the input signal via

$$\begin{array}{rcl} \mathbf{u} & \leq & \bar{\mathbf{u}} \\ \mathbf{u} & \geq & \underline{\mathbf{u}} \end{array}, \quad \bar{\mathbf{u}} = \begin{bmatrix} \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(N) \end{bmatrix}, \ \underline{\mathbf{u}} = \begin{bmatrix} \underline{u}(1) \\ \underline{u}(2) \\ \vdots \\ \underline{u}(N) \end{bmatrix}. \tag{B.5}$$

With all constraints defined as LMIs, the feasibility of a multivariable input signal

$\mathbf{u}$ that satisfies the constraints (B.3)–(B.5) can be written as a LP problem

$$\begin{aligned} &\min\ \boldsymbol{f}^T\mathbf{x},\ \text{subject to} \\ &\qquad \mathbf{z} \geq 0 \\ &\begin{bmatrix} L & -\mathbf{1} \end{bmatrix} \mathbf{x} \leq \mathbf{b}\ , \quad L = \begin{bmatrix} \Phi \\ -\Phi \\ E \\ -E \end{bmatrix},\ \mathbf{b} = \begin{bmatrix} \mathbf{y}+\varepsilon \\ -\mathbf{y}+\varepsilon \\ \mathbf{d} \\ \mathbf{d} \end{bmatrix} \\ &\qquad \mathbf{u} \leq \bar{\mathbf{u}} \\ &\qquad \mathbf{u} \geq \underline{\mathbf{u}} \end{aligned} \tag{B.6}$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{u}^T & \mathbf{z}^T \end{bmatrix}^T$. The variable ε is generally related to the obtainable angle tracking error due to possible limited resolution of the FSM or tolerances on the angles. Finding a feasible solution to the LP problem (B.6) using the desired angle pair $(\alpha_2(k), \beta_2(k))$ now leads to the dynamic response of FSM_2 in Figure B.5, top figure.

B.5 Image motion compensation

A basic IMC concept illustrated in [56] consists of a *fine* device working with a *coarse* device. FSM_2 is considered the coarse device while the fine device is considered the IMC device, represented by FSM_1 in this exercise. The residual FSM_2 error signal is used to define the motion of the IMC device. The definition of a new trajectory for FSM_1, $(x_1(\phi), y_1(\phi))$, is the difference of the desired target and the obtained result. These computations lead to the dynamic response of the combination of FSM_1 and FSM_2 summarized in Figure B.5.

The IMC implemented by FSM_1, following the limitations of the position accuracy of FSM_2, is capable of a fast change in the direction of the beam due to its high angular resolution and bandwidth. Keeping FSM_1 initially centered, e.g. $(\alpha_1, \beta_1) = (0, 0)$ and computing the position angles of FSM_2 first, provides FSM_1 with the capability to rotate to its full capacity on its two axis for error correcting purposes.

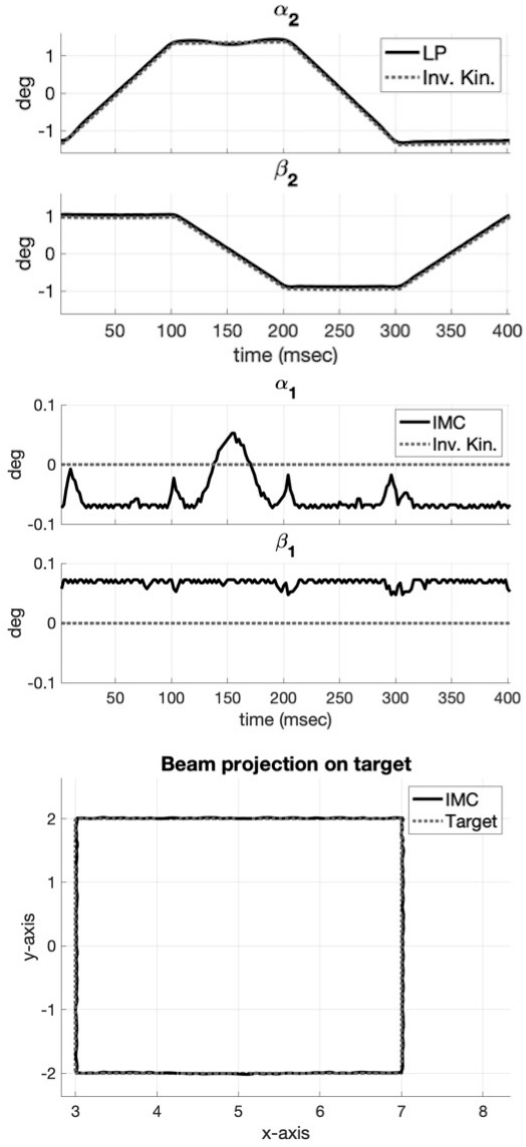


Figure B.5: Combined response of FSM_2 with dynamic IS and FSM_1 with dynamic IS for target error correction. Top figure: comparison of discrete-time dynamic angle pair $(\alpha_2(k), \beta_2(k))$ (solid lines) and desired inverse kinematic angle pair $(\alpha_2(\phi), \beta_2(\phi))$ (dashed lines). Middle figure: comparison of discrete-time dynamic angle pair $(\alpha_1(k), \beta_1(k))$ (solid lines) and initial zero angle pair $(0, 0)$ (dashed lines). Bottom figure: comparison of beam projection on the target plane with FSM_1 $(\alpha_1(k), \beta(k))$ and FSM_2 $(\alpha_2(k), \beta_2(k))$ (solid line) and kinematic angles $(\alpha_2(\phi), \beta_2(\phi))$ (dashed line).

B.6 CONCLUSIONS

Tracking performance of a target in a dual-mirror system is significantly improved by an open-loop approach in which kinematic analysis and modeling of dynamics is combined with dynamic input shaping. Using (inverse) kinematic analysis, the desired rotation angles of the main FSM can be computed while the state-space model can be used to formulate the dynamic input signal design via a LP problem. It is worthwhile to notice the LP problem offers the advantage to compute IS while satisfying several constraints. The constraints are given by the mechanical specifications of the FSM and the desired result for a specific application; the rate of change $\mathbf{d}$ and the size of the input signal $\bar{\mathbf{u}}$ and $\underline{\mathbf{u}}$ are attached to the mechanical specs of the mirror, while the margin of error ε is defined by the desired response of the application. Any remaining target tracking errors can be picked up by the second FSM acting as an IMC device. The end result is a reduction of the tracking error, while maintaining constraints on rate-of-change and size of the voltage signals applied to the FSMs in the dual-mirror system.

B.7 Acknowledgment

This appendix, in full, is a reprint of A. Dautt-Silva and R. A. de Callafon (2022) "Application of Dynamic Input Shaping for a Dual-Mirror System." *ASME. J. Dyn. Sys., Meas., Control*, 144(4). The dissertation author is the first author of this article.

Appendix C

Steering Mirror Kinematics

C.1 Steering Mirror Beam-Pointing Kinematics

Considering the general reflection problem: a beam in an arbitrary plane being reflected by a mirror and the mirror normal, this problem can be expressed as a matrix multiplication.

$$p_1 = Tp_0, \tag{C.1}$$

where p_0 and p_1 represent unit vectors in the directions of the incident beam and reflected beam, respectively; and T is the reflection matrix in terms of the mirror normal components, n_x, n_y and n_z. We talk in the further sections about the full development of the reflection matrix T, [41].

C.1.1 Normal of the mirror

Consider the steering mirror in Figure C.1, whose rotation axes are located at the center of the mirror. The x and y axes are in the plane of the mirror and z axis is the normal n_0 to the surface and considered the local mirror frame. The mirror can be rotated

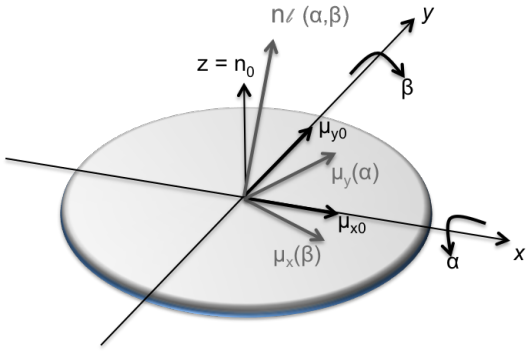


Figure C.1: Steering mirror rotation axes and normal.

about both its x and y axes independently by the angles α and β. The process to obtain the new normal vector n of the mirror in the local mirror frame is through the following steps.

We first define the unit vectors μ_{x0} and μ_{y0}, for axes x and y respectively,

$$\mu_{x0} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mu_{y0} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{C.2}$$

Knowing the rotation angles α and β we utilize the Euler transform matrices (also known as rotation matrices),

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \qquad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}. \tag{C.3}$$

Rotating the mirror on both axes modifies the unit vectors. The new unit vectors are derived by multiplying the Euler transform matrix by the original unit vectors,

$$\mu_x(\beta) = R_y(\beta)\mu_{x0} = \begin{bmatrix} \cos\beta \\ 0 \\ -\sin\beta \end{bmatrix}, \qquad \mu_y(\alpha) = R_x(\alpha)\mu_{y0} = \begin{bmatrix} 0 \\ \cos\alpha \\ \sin\alpha \end{bmatrix}. \tag{C.4}$$

The normal of the mirror in local mirror frame, n_ℓ, is obtained with the cross product of both new unit vectors,

$$n_\ell = n_\ell(\alpha, \beta) = \mu_x(\beta) \times \mu_y(\alpha) = \begin{bmatrix} \cos\alpha\sin\beta \\ -\cos\beta\sin\alpha \\ \cos\beta\cos\alpha \end{bmatrix}, \tag{C.5}$$

.

Working on a global frame coordinate system, we need to reference the normal $n_\ell(\alpha, \beta)$ of the mirror in the local mirror frame, relative to the global frame (e.g. the mounting angle of the mirror). The angle of rotation of the mirror relative to the global frame can be described by a rotation around the x axis $R_x(\phi)$, the y axis $R_y(\theta)$ and the z axis $R_z(\psi)$,

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix},$$

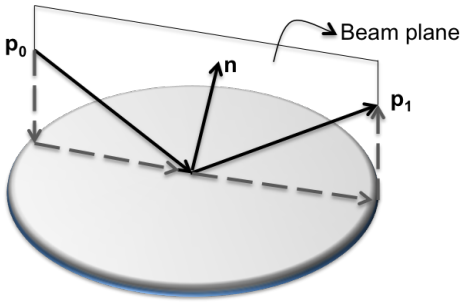


Figure C.2: Incident and reflected beam decomposition.

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.6}$$

The normal n in global frame can be described as

$$n(\phi, \theta, \psi, n_\ell(\alpha, \beta)) = R_x(\phi)R_y(\theta)R_z(\psi)n_\ell(\alpha, \beta), \tag{C.7}$$

where the normal n$(\cdot)$ in Eq.(C.7) is now a function of the rotation angles (ϕ, θ, ψ) and the local normal n_l in Eq.(C.5).

Keep in mind the mounting angles (ϕ, θ, ψ) are fixed, so $R_x(\phi)$, $R_y(\theta)$ and $R_z(\psi)$ can be referred as a constant $R = R_x(\phi)R_y(\theta)R_z(\psi)$, so that

$$n(\alpha, \beta) = Rn_\ell(\alpha, \beta). \tag{C.8}$$

C.1.2 Beam reflection

With the rotated normal $n(\alpha, \beta)$ given in Eq.(C.8) the reflection of a beam can be computed as follows.

Figure C.2 shows an incident beam p_0 and the reflected beam p_1; p_0 and $n(\alpha, \beta)$ span the

beam plane and the reflected beam lies in the same beam plane, due to the assumption of a flat mirror. Figure C.2 shows the beams' vectors decomposed into perpendicular and parallel components to the normal vector [50]. With the definition of $p_{1\perp}$ and $p_{1\parallel}$ it is clear that

$$\begin{aligned} p_1 &= p_{1\perp} + p_{1\parallel} = p_{0\perp} - p_{0\parallel} \\ p_1 &= p_0 - 2p_0 n(\alpha,\beta) n^T(\alpha,\beta), \end{aligned} \tag{C.9}$$

rewritten in matrix form, the expression for the vector p_1 is given by

$$p_1 = \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} - 2n(\alpha,\beta) n^T(\alpha,\beta) \right) p_0, \tag{C.10}$$

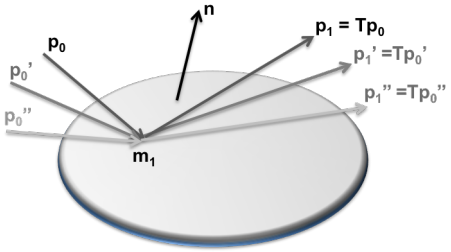
where $T(n(\alpha,\beta))$ is the reflection matrix in terms of the normal vector $n(\alpha,\beta)$ of the mirror,

$$T(n(\alpha,\beta)) = I_3 - 2n(\alpha,\beta) n^T(\alpha,\beta), \tag{C.11}$$

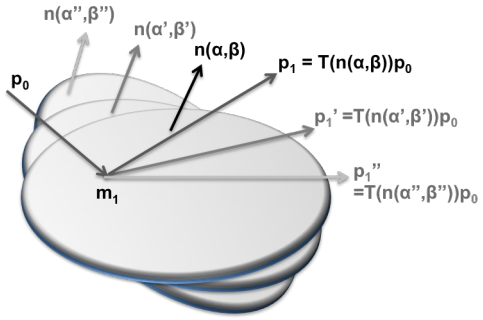
abbreviated to $p_1 = Tp_0$.

If we keep a mirror in a fixed position, the normal $n(\alpha,\beta)$ is constant, thus we can see the reflection matrix $T(n(\alpha,\beta))$ is constant as well; and the reflected beam p_1 is only a function of the incident beam p_0. Given a point m_1, of intersection of the beam with the mirror, all the reflected beams will be in function of the incident beams as shown on Figure C.3a.

A similar approach is keeping the incident beam constant, and varying the (α,β) angles of the mirror. In this case, the reflection matrix $T(n(\alpha,\beta))$ varies in function of the mirror position. Since p_0 is constant, the reflected beam is in function of the reflection matrix $T(n(\alpha,\beta))$, Figure C.3b.



(a) Reflected beams in function of incident beams



(b) Reflected beams in function of mirror rotation

Figure C.3: Reflected beams.

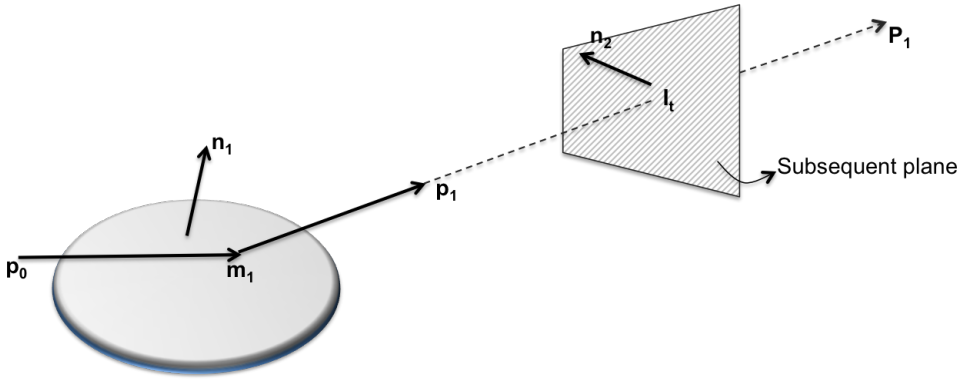


Figure C.4: Reflected beam intersection.

C.1.3 Beam intersection point

As the reflected beam p_1 leaves the mirror it is required to compute the location of the beam on a subsequent plane. This plane could be an imaging surface or another secondary mirror, Figure C.4.

Consider the reflected beam with a direction vector p_1, start point m_1, and a final point P_1 which is derived from the previous values. We have the segment $\overline{m_1 P_1}$ [43].

$$P_1(m_1, p_1) = m_1 + kdp_1^T, \tag{C.12}$$

where k is a scale factor to assure the segment will be longer than the distance d to the plane. The distance d is a constant value known based on the opto-mechanical layout.

The intersection with the plane I_t is derived with the parametric equation representation for lines,

$$I_t(m_1, P_1(m_1, p_1), n_2) = m_1 + s \times (P_1 - m_1), \tag{C.13}$$

where n_2 is the normal of the subsequent plane and s is the intersection parameter scalar obtained as

$$s = \frac{n_2 \cdot (m_1 - p)}{n_2 \cdot (P_1 - m_1)}, \tag{C.14}$$

the plane is given by any point p on the plane and its normal vector n_2. The opto-mechanical layout gives us this information.

C.2 One mirror system

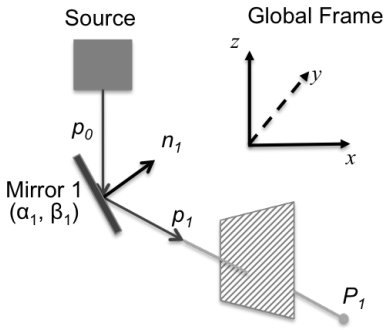


Figure C.5: Beam and steering mirror.

Consider a beam source and a mirror as in Figure C.5. Mirror 1 is rotated 45° with respect to y in the global frame and the incident beam intersects mirror 1 in point m_1. Following the process from Section C.1, the derivation of the target based on the angles is as follows.

Obtain the x and y components for the normal n_1 of the mirror based on the rotation angles and the the original vectors,

$$\mu_{x1}(\beta_1) = R_{y1}(\beta_1) \cdot \mu_{x0}, \qquad \mu_{y1}(\alpha_1) = R_{x1}(\alpha_1) \cdot \mu_{y0}, \tag{C.15}$$

with these components calculate the normal of the mirror in the local mirror frame and then the normal of the mirror in the global frame,

$$n_{\ell 1}(\alpha_1, \beta_1) = \mu_{x1} \times \mu_{y1}, \qquad n_1(\alpha_1, \beta_1) = R_y(45°) n_{\ell 1}(\alpha_1, \beta_1). \tag{C.16}$$

With the normal n_1, the reflection matrix and the outgoing beam direction can be calculated,

$$T_1 = T_1(n_1(\alpha_1, \beta_1)) = I_3 - 2n_1(\alpha_1, \beta_1)n_1^T(\alpha_1, \beta_1), \qquad p_1 = T_1 p_0. \tag{C.17}$$

The last step is finding the intersection of the reflected beam with the target.

$$P_1(m_1, p_1) = m_1 + kdp_1^T \tag{C.18}$$

The intersection with the plane I_t is derived with the parametric equation representation for lines,

$$I_t(m_1, P_1, n_t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = m_1 + \frac{n_t \cdot (m_1 - p)}{n_t \cdot (P_1 - m_1)} \times (P_1 - m_1), \tag{C.19}$$

where p is a point in the subsequent plane and n_t is its normal.

Figure C.6 shows a simulation of the full mirror rotation in both axes and the intersections in a plane at a distance $d = 25$.

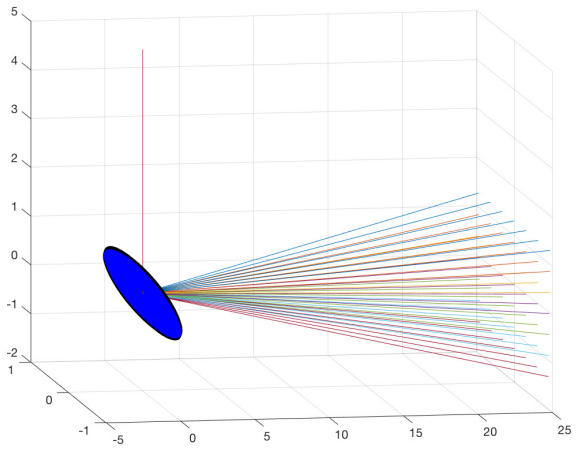


Figure C.6: Full mirror 1 rotation on both axes.

C.3 Forward kinematics of the two mirrors system

Consider a system of two mirrors with the opto-mechanical layout shown in Figure C.7. With the forward kinematics definitions on Section C.1 and the exercise on Section C.2, we can derive $[x,\ y,\ z]$ starting from p_0. The process for both optical devices is as the one followed in Section C.2, though some changes are considered based on the mounting of the devices. Mirror 1 is rotated 45° with respect to y axis in the global frame; mirror 2 is rotated -135° also with respect to y axis in the global frame.

The process to find the target is as follows:

1. Derive the reflected beam coming from mirror 1 based on the angles and normal of the mirror

$$p_1 = T_1(n_1(\alpha_1, \beta_1))p_0 \tag{C.20}$$

2. Derive the reflected beam coming from mirror 2 based on the angles and normal of the mirror

$$p_2 = T_2(n_2(\alpha_2, \beta_2))p_1 \tag{C.21}$$

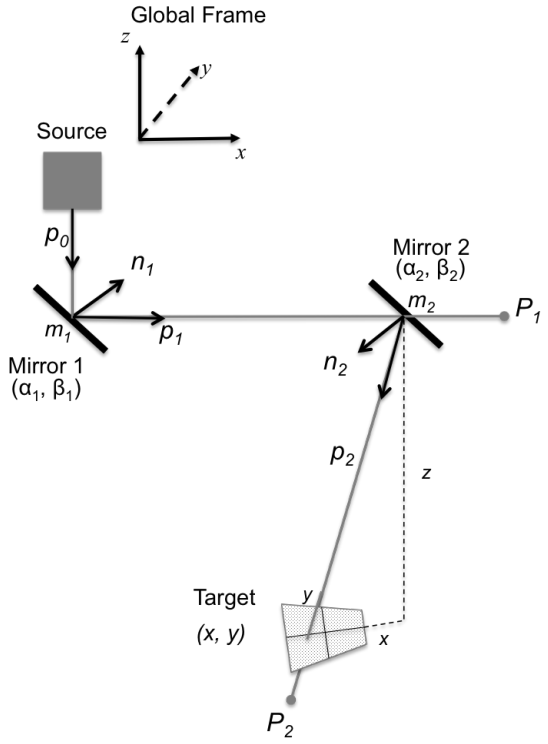


Figure C.7: Steering mirror rotation axes and normal.

3. Derive the final point of a segment from the origin of the beam in mirror 1 passing through the plane in mirror 2

$$P_1 = P(m_1, p_1) \tag{C.22}$$

4. Derive the point of intersection of the beam with mirror 2

$$m_2 = I(m_1, P_1, n_2) \tag{C.23}$$

5. Derive the final point of a segment from the origin of the beam in mirror 2 passing through the plane of the target

$$P_2 = P(m_2, p_2) \tag{C.24}$$

6. Derive the point of intersection of the beam with the target

$$t = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = I(m_2, P_2, n_t) \tag{C.25}$$

We follow these steps in detail in the next sections.

C.3.1 Mirror 1 to mirror 2

Following the process explained in the previous sections we show the relation from (α_1, β_1) of the reflected beam p_1 to the point of intersection m_2 with mirror 2.
We have Euler rotation matrices $\mathrm{R}_{x1}(\alpha_1)$ and $\mathrm{R}_{y1}(\beta_1)$. We obtain the normal vector of the mirror $n_{\ell 1}$ and the normal in global frame n_1.

$$n_{\ell 1}(\alpha_1, \beta_1) = \mu_{x1}(\beta_1) \times \mu_{y1}(\alpha_1), \qquad n_1(\alpha_1, \beta_1) = R_y(45°) n_\ell(\alpha_1, \beta_1). \tag{C.26}$$

The reflection vector p_1 is obtained from the reflection matrix T_1 based on the normal of the mirror n_1 and the incident beam p_0,

$$T_1 = T_1(n_1(\alpha_1, \beta_1)) = I_3 - 2n_1(\alpha_1, \beta_1) n_1^T(\alpha_1, \beta_1), \tag{C.27}$$

$$p_1 = T_1 p_0.$$

The point of intersection m_2 of the beam p_1 with mirror 2, is derived by creating a segment starting in m_1 and ending in point P_1 in the direction of the beam.

$$P_1 = m_1 + k d p_1^T. \tag{C.28}$$

Before finding m_2, we need to derive the normal of mirror 2.

C.3.2 Mirror 2 to target

Following the same process, we show the relation from (α_2, β_2) of the reflected beam p_2 to the point of intersection t = *[x,y,z]'* with the target.
We have Euler rotation matrices $\mathrm{R}_{x2}(\alpha_2)$ and $\mathrm{R}_{y2}(\beta_2)$. We obtain the normal vector of the mirror $n_{\ell 2}$ and the normal in global frame n_2.

$$n_{\ell 2}(\alpha_2, \beta_2) = \mu_{x2}(\beta_2) \times \mu_{y2}(\alpha_2), \qquad n_2(\alpha_2, \beta_2) = R_y(-135°) n_{\ell 2}(\alpha_2, \beta_2). \tag{C.29}$$

The reflection vector p_2 is obtained from the reflection matrix T_2 based on the normal of the mirror n_2 and the incident beam p_1,

$$T_2 = T_2(n_2(\alpha_2, \beta_2)) = I_3 - 2n_2(\alpha_2, \beta_2) n_2^T(\alpha_2, \beta_2), \tag{C.30}$$

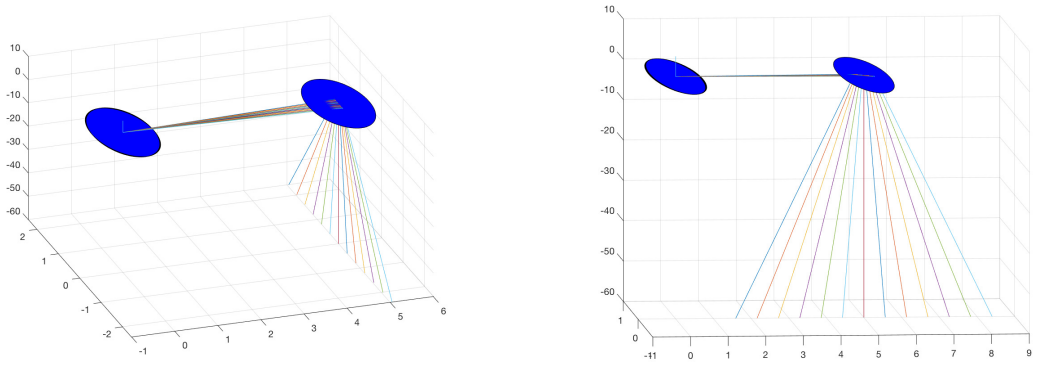
$$p_2 = T_2 p_1.$$

Following with the point of intersection of p_1 with mirror 2,

$$m_2 = m_1 - \frac{n_2 \cdot (p_2 - m_1)}{n_2 \cdot (P_1 - m_1)} \times (P_1 - m_1). \tag{C.31}$$

$$P_2 = m_2 + k d_t p_2^T,$$

Now we have the segment starting in m_2 and ending in point P_2. In the same way, we derive the intersection with the target t,



(a) Full rotation in y axis of mirror 1

(b) Full rotation in y axis of mirror 1

Figure C.8: Mirror 1 rotation.

$$t = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = m_2 - \frac{n_t \cdot (m_2 - p_t)}{n_t \cdot (P_2 - m_2)} \times (P_2 - m_2), \tag{C.32}$$

where p_t is a point in the target plane and n_t is the normal of the plane. In this case we assume the plane is flat, n_t= [0,0,1]'.

C.3.3 Mirrors system simulation

Figure C.8a and Figure C.8b show the beam rays that result of rotating mirror 1 in the x and y axis, respectively, through the full α range.

Figure C.9 shows the result of both mirrors, moved in their full range in both angles. This illustrates the target area that could be reached with this opto-mechanical layout; the area resembles a trapezoidal shape.

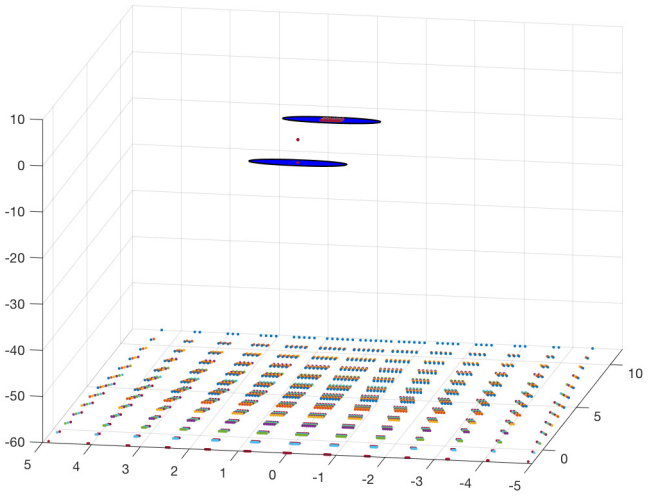


Figure C.9: Full range of both mirrors on both axes.

C.4 Mathematical relation between angles and target

C.4.1 Angles to intersection point

Consider a beam source and a mirror as in Figure C.5. First we get a target m_2 as a function of (α_1, β_1).

$$n_1(\alpha_1, \beta_1) = Rn_{\ell 1}(\alpha_1, \beta_1) \tag{C.33}$$

With the normal n_1, the reflection matrix and the outgoing beam direction can be calculated,

$$T_1 = T_1(n_1(\alpha_1, \beta_1)) = I_3 - 2n_1(\alpha_1, \beta_1)n_1^T(\alpha_1, \beta_1) \tag{C.34}$$

$$p_1 = T_1 p_0. \tag{C.35}$$

The next step is finding the intersection of the reflected beam with the target,

$$P_1(m_1, p_1) = m_1 + kdp_1^T \tag{C.36}$$

$$= m_1 + kd(T_1 p_0)^T .. \tag{C.37}$$

The intersection with the plane m_2 is derived with the parametric equation representation for lines, and substituting P_1 from Eq. (C.37),

$$m_2(m_1, P_1, n_t) = m_1 + s \times (P_1 - m_1), \tag{C.38}$$

$$m_2(m_1, T_1, n_t) = m_1 + s \times (kd(T_1 p_0)^T). \tag{C.39}$$

Assuming our only variables are (α_1, β_1), let's use m_2 as a function of $T_1(n_1(\alpha_1, \beta_1))$.

A function $f : A \to B$ is said to be injective (or one-to-one) if, for all x_1, $\mathrm{x}_2 \in A$, $f(\mathrm{x}_1) = f(\mathrm{x}_2)$ implies $\mathrm{x}_1 = \mathrm{x}_2$.

Let α_1, α_2, β_1 and $\beta_2 \in \mathbb{R}$ and suppose that $f(\alpha_1, \beta_1) = f(\alpha_2, \beta_2)$, then

$$m_1 + s \times kd[T_1(n_1(\alpha_1, \beta_1))p_0]^T = m_1 + s \times kd[T_2(n_2(\alpha_2, \beta_2))p_0]^T, \tag{C.40}$$

Simplify Eq.(C.40),

$$m_1 + s \times kd[T_1(n_1(\alpha_1, \beta_1))p_0]^T = m_1 + s \times kd[T_2(n_2(\alpha_2, \beta_2))p_0]^T, \tag{C.41}$$

$$s \times kd[T_1(n_1(\alpha_1, \beta_1))p_0]^T = s \times kd[T_2(n_2(\alpha_2, \beta_2))p_0]^T, \tag{C.42}$$

s, k and d are constant scalar values, and p_0 is a constant vector, so we can simplify further

$$[T_1(n_1(\alpha_1, \beta_1))p_0]^T = [T_2(n_2(\alpha_2, \beta_2))p_0]^T, \tag{C.43}$$

$$T_1(n_1(\alpha_1, \beta_1)) = T_2(n_2(\alpha_2, \beta_2)). \tag{C.44}$$

Going further on the derivation, use the definition for T,

$$T_1(n_1(\alpha_1, \beta_1)) = T_2(n_2(\alpha_2, \beta_2)) \tag{C.45}$$

$$I - 2n_1(\alpha_1, \beta_1)n_1^T(\alpha_1, \beta_1) = I - 2n_2(\alpha_2, \beta_2)n_2^T(\alpha_2, \beta_2) \tag{C.46}$$

$$n_1(\alpha_1, \beta_1)n_1^T(\alpha_1, \beta_1) = n_2(\alpha_2, \beta_2)n_2^T(\alpha_2, \beta_2) \tag{C.47}$$

$$n_1(\alpha_1, \beta_1) = n_2(\alpha_2, \beta_2). \tag{C.48}$$

Using the derivation for n,

$$n_1(\alpha_1, \beta_1) = n_2(\alpha_2, \beta_2) \tag{C.49}$$

$$Rn_{\ell 1}(\alpha_1, \beta_1) = Rn_{\ell 2}(\alpha_2, \beta_2) \tag{C.50}$$

$$n_{\ell 1}(\alpha_1, \beta_1) = n_{\ell 2}(\alpha_2, \beta_2). \tag{C.51}$$

The normal of the mirror in the local mirror frame, n_ℓ, is obtained with the cross product of the unit vectors, Eq. (C.5). If we linearize it we obtain

$$n_{\ell 1}(\alpha_1, \beta_1) = n_{\ell 2}(\alpha_2, \beta_2) \tag{C.52}$$

$$\begin{bmatrix} \cos\alpha_1 \sin\beta_1 \\ -\cos\beta_1 \sin\alpha_1 \\ \cos\alpha_1 \cos\beta_1 \end{bmatrix} = \begin{bmatrix} \cos\alpha_2 \sin\beta_2 \\ -\cos\beta_2 \sin\alpha_2 \\ \cos\alpha_2 \cos\beta_2 \end{bmatrix} \tag{C.53}$$

$$\begin{bmatrix} \beta_1 \\ \alpha_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \beta_2 \\ \alpha_2 \\ 1 \end{bmatrix} \tag{C.54}$$

Since *f(x,y)* is an injective functions, it is possible to solve the task inversely: given a point of intersection m_2, we can compute (α_1, β_1).

C.5 Steering Mirror Inverse Kinematics

In the first approach we give angles (α_1, β_1) and (α_2, β_2) for each mirror and compute the target $[x,\ y,\ z]$.

A more realistic approach is the opposite. You have a defined target $[x,\ y,\ z]$ for which we find a set of ¡$\alpha_1, \beta_1,\ \alpha_2, \beta_2$¿ that matches the target.

C.5.1 Inverse Kinematics with mirror 2 in a fixed position

In this first approach let's define $(\alpha_2, \beta_2) = (0,0)$ as the restriction. We can consider any value, but the main point is to keep mirror 2 fixed.

To start the inverse kinematics process, we normalize and invert the target I_t to obtain the inverse direction vector p_2 towards mirror 2. Having (α_2, β_2) defined as (0,0) we can obtain p_1, the incoming beam direction from mirror 1. Having the beam direction p_1 and p_0 we can obtain n_1, the normal of mirror 1, and the values of angles (α_1, β_1) with arctangent function.

The new equations we bring over for this process are the following.

$$p_2 = \frac{I_t}{\|I_t\|}, \tag{C.55}$$

$$p_1 = T_2 p_2, \tag{C.56}$$

$$n_1 = \frac{p_1 - p_0}{\|p_1 - p_0\|}, \tag{C.57}$$

$$n_{\ell 1} = R_y^{-1}(45°) n_1 \tag{C.58}$$

With the $[x,\ y\ ,\ z]$ coordinates from the normal $n_{\ell 1}$ we can then obtain the angles in

(α_1, β_1).

$$\alpha_1 = atan(\frac{-n_{\ell 1}(y)}{n_{\ell 1}(z)}), \tag{C.59}$$

$$\beta_1 = atan(\frac{n_{\ell 1}(x)}{n_{\ell 1}(z)}). \tag{C.60}$$

C.5.2 Inverse Kinematics with mirror 1 in a fixed position

In this approach let's define $(\alpha_1, \beta_1) = (0,0)$ as the restriction. Again, We cam consider any value, but the main point is to keep mirror 1 fixed.

To start the inverse kinematics process, we normalize and invert the target I_t to obtain the inverse direction vector p_2 towards mirror 2. Having (α_1, β_1) defined as (0,0) we can obtain p_1, the incoming beam direction from mirror 1. Having the beam direction p_1 and p_2 we can obtain n_2, the normal of mirror 2, and the values of angles (α_2, β_2) with arctangent function.

The new equations we bring over for this process are the following.

$$p_2 = \frac{I_t}{\|I_t\|}, \tag{C.61}$$

$$p_1 = T_1 p_0, \tag{C.62}$$

$$n_2 = \frac{p_2 - p_1}{\|p_2 - p_1\|}, \tag{C.63}$$

$$n_{\ell 2} = R_y^{-1}(-135^\circ) n_2 \tag{C.64}$$

With the $[x,\ y\ ,\ z]$ coordinates from the normal $n_{\ell 2}$ we can then obtain the angles in

(α_2, β_2).

$$\alpha_2 = atan(\frac{-n_{\ell 2}(y)}{n_{\ell 2}(z)}), \tag{C.65}$$

$$\beta_2 = atan(\frac{n_{\ell 2}(x)}{n_{\ell 2}(z)}). \tag{C.66}$$

Bibliography

[1] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 76–82, 03 1990.

[2] L. Y. Pao, "Multi-input shaping design for vibration reduction," *Automatica*, vol. 35, pp. 81–89, 1999.

[3] M. D. Baumgart and L. Y. Pao, "Discrete time-optimal command shaping," *Automatica*, vol. 43, no. 8, pp. 1403–1409, 2007.

[4] M. O. T. Cole and T. Wongratanaphisan, "Optimal FIR Input Shaper Designs for Motion Control With Zero Residual Vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, 03 2011.

[5] N. Singer, W. Singhose, and E. Kriikku, "An input shaping controller enabling cranes to move without sway," in *Conference: 7. American Nuclear Society topical meeting on robotics and remote systems.*

[6] M. A. Lau and L. Y. Pao, "Input shaping and time-optimal control of flexible structures," *Automatica*, vol. 39, no. 5, pp. 893–900, 2003.

[7] D. Adair and M. Jaeger, "Aspects of input shaping control of flexible mechanical systems," *The Mathematica Journal*, vol. 19, 06 2017.

[8] Y. Al Hamidi and M. Rakotondrabe, "Comparison study of input shaping techniques to control an underactuated flexible link system," in *Sensors for Next-Generation Robotics III* (D. Popa and M. B. J. Wijesundara, eds.), vol. 9859, pp. 74 – 89, International Society for Optics and Photonics, SPIE, 2016.

[9] W. Singhose, "Command shaping for flexible systems: A review of the first 50 years," *International Journal of Precision Engineering and Manufacturing*, vol. 10, no. 4, pp. 153–168, 2009.

[10] K. Zhou and J. Doyle, *Essentials of Robust Control*. Prentice Hall, 1998.

[11] J. Doyle, B. Francis, and A. Tannenbaum, *Feedback Control Theory*. Macmillan Publishing Co., 1990.

[12] S. K. Shome, A. Mukherjee, and U. Datta, "Input shapers against system parametric uncertainty," in *2013 International Conference on Machine Intelligence and Research Advancement*, pp. 299–304, 2013.

[13] L. Y. Pao and M. A. Lau, "Robust Input Shaper Control Design for Parameter Variations in Flexible Structures ," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 63–70, 10 1999.

[14] L. Pao, T. Chang, and E. Hou, "Input shaper designs for minimizing the expected level of residual vibration in flexible structures," in *Proceedings of the 1997 American Control Conference*, vol. 6, pp. 3542–3546 vol.6, 1997.

[15] J. Vaughan, A. Yano, and W. Singhose, "Robust Negative Input Shapers for Vibration Suppression," *Journal of Dynamic Systems, Measurement, and Control*, vol. 131, 03 2009.

[16] P. Singla, T. Singh, and U. Konda, *A Probabilistic Approach for Robust Input Shapers Design for Precise Point-to-Point Control*.

[17] U. Staehlin and T. Singh, "Design of closed-loop input shaping controllers," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 6, pp. 5167–5172 vol.6, 2003.

[18] T. Singh and W. Singhose, "Tutorial on input shaping/time delay control of maneuvering flexible structures," in *In American Control Conference*, 2002.

[19] M. K. Jouaneh and E. Anderson, "Input shaping using finite impulse response filters," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 6525–6530, 2006.

[20] S. Boyd and L. Vandenbergue, *Convex Optimization*. Cambridge University Press, 2004.

[21] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[22] P. V. Ubale, "An approach for the optimal solution of milp problems," *Bulletin of Pure & Applied Sciences- Mathematics and Statistics*, vol. 31e, no. 2, pp. 169–173, 2012.

[23] W. Singhose, R. Eloundou, and J. Lawrence, "Command generation for flexible systems by input shaping and command smoothing," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 6, pp. 1697–1707, 2010.

[24] M. Gniadek and S. Brock, "Basic algoruthms of input shaping autotuning," in *MM Science Journal*, 2015.

[25] W. E. Singhose, "Command generation for flexible systems," in *Ph.D. thesis, MIT*, 1997.

[26] U. Boettcher, R. A. de Callafon, and F. E. Talke, "Closed-loop input shaping in discrete-time lti systems," in *ASME Information Storage and Processing Systems Confeference*, 2010.

[27] G. Cornuéjols, "Valid inequalities for mixed integer linear programs," *Mathematical Programming*, vol. 112, pp. 3–44, 2008.

[28] U. Boettcher, R. A. de Callafon, and F. E. Talke, "Multiobjective time domain input shaping for closed-loop discrete-time systems," in *Prepr. 5th IFAC Symposium on Mechatronic Systems*, (Cambridge, MA), pp. 200–205, 2010.

[29] P. E. Gill, W. Murray, and M. H. Wright, *Numerical linear algebra and optimization*. Addison-Wesley Publishing Company, 1991.

[30] W. Durham, K. A. Bordignon, and R. Beck, *Aircraft Control Allocation*. Wiley, 2016.

[31] M. Vossler and T. Singh, "Minimax State/Residual-Energy Constrained Shapers for Flexible Structures: Linear Programming Approach," *Journal of Dynamic Systems, Measurement, and Control*, vol. 132, 04 2010.

[32] N. Chen, B. Potsaid, J. T. Wen, S. Barry, and A. Cable, "Modeling and control of a fast steering mirror in imaging applications," in *IEEE International Conference on Automation Science and Engineering*, pp. 27–32, 2010.

[33] P. Merritt and M. Spencer, *Beam Control for Laser Systems*. Directed Energy Professional Society, 2018.

[34] W. Zhu, L. Bian, Y. An, G. Chen, and X. Rui, "Modeling and control of a two-axis fast steering mirror with piezoelectric stack actuators for laser beam tracking," *Smart Materials and Structures*, vol. 24, no. 7, 2015.

[35] M. Roggemann and B. Welsh, *Imaging Through Turbulence*. CRC Press, 1996.

[36] B.-S. Kim, S. Gibson, and T.-C. Tsao, "Adaptive control of a tilt mirror for laser beam steering," in *Proc. American Control Conference*, (Boston, Massachusetts), pp. 3417–3421, 2004.

[37] G. Schitter, P. Thurner, and P. Hansma, "Design and input-shaping control of a novel scanner for high-speed atomic force microscopy," *Mechatronics*, 2008.

[38] R. A. de Callafon and D. J. Miller, "Identification of linear time-invariant systems via constrained step-based realization," in *Prepr. 16th IFAC Symposium on System Identification*, vol. 45, pp. 1155–1160, 2012.

[39] K. Kim, S. Moon, J. Kim, Y. Parkand, and J.-H. Lee, "Input shaping based on an experimental transfer function for an electrostatic microscanner in a quasistatic mode," *Micromachines*, vol. 10, pp. 217–220, 2019.

[40] J. C. DeBruin and D. B. Johnson, "Line-of-sight reference frames: a unified approach to plane-mirror optical kinematics," in *Acquisition, Tracking, and Pointing VI*, vol. 1697, pp. 111–129, SPIE, 1992.

[41] J. M. Hilkert, G. Kanga, and K. Kinnear, "Line-of-sight kinematics and corrections for fast-steering mirrors used in precision pointing and tracking systems," in *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XI*, vol. 9076, pp. 115–129, SPIE, 2014.

[42] A. Dautt-Silva and R. A. de Callafon, "Optimal input shaping with finite resolution computed from step-response experimental data," in *2018 Annual American Control Conference*, pp. 6703–6708, 2018.

[43] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, 1998.

[44] Q. Zhou, P. Ben-Tzvi, D. Fan, and A. A. Goldenberg, "Design of fast steering mirror systems for precision laser beams steering," in *ROSE 2008 – IEEE International Workshop on Robotic and Sensors Environments*, pp. 144–149, 2008.

[45] J. H. Hilkert, "Inertially stabilized platform technology: Concepts and principles," *IEEE Control Systems Magazine*, vol. 28, no. 1, pp. 26–46, 2008.

[46] Q. Zhou, P. Ben-Tzvi, and D. Fan, "Design and analysis of a fast steering mirror for precision laser beams steering," *Sensors and Transducers Journal*, vol. 5, pp. 104–118, 2009.

[47] D. J. Kluk, M. T. Boulet, and D. L. Trumper, "A high-bandwidth, high-precision, two-axis steering mirror with moving iron actuator," *Mechatronics*, vol. 22, no. 3, pp. 257–270, 2012.

[48] R. J. Watkins, H.-J. Chen, B. N. Agrawal, and Y. S. Shin, "Optical beam jitter control," in *Free-Space Laser Communication Technologies XVI*, vol. 5338, pp. 204 – 213, SPIE, 2004.

[49] R. W. Cochran and R. H. Vassar, "Fast-steering mirrors in optical control systems," in *Advances in Optical Structure Systems*, vol. 1303, pp. 245–251, SPIE, 1990.

[50] J. M. Hilkert, "Kinematic algorithms for line-of-sight pointing and scanning using ins/gps position and velocity information," in *Acquisition, Tracking, and Pointing XIX*, vol. 5810, pp. 11–22, SPIE, 2005.

[51] Y. K. Yong, K. Liu, and R. Moheimani, "Reducing cross-coupling in a compliant xy nanopositioner for fast and accurate raster scanning," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1172–1179, 2010.

[52] H. Ngo, Q. Nguyen, and W. Kim, "Implementation of input shaping control to reduce residual vibration in industrial network motion system," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1693–1698, 2015.

[53] S. Lim, H. D. Stevens, and J. P. How, "Input shaping design for multi-input flexible systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 121, no. 3, pp. 443–447, 1999.

[54] U. Boettcher, R. A. de Callafon, and F. E. Talke, "Data based modeling and control of a dual-stage actuator hard disk drive," in *Proc. 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pp. 8316–8321, 2009.

[55] A. Dautt-Silva and R. A. de Callafon, "Multivariable dynamic input shaping for two-axis fast steering mirror," in *Reference Prefiltering for Precision Motion control*, IFAC World Congress (preprint), 2020.

[56] S. T. Jenkins and J. M. Hilkert, "Line Of Sight Stabilization Using Image Motion Compensation," in *Acquisition, Tracking, and Pointing III*, vol. 1111, pp. 98 – 115, SPIE, 1989.

[57] S. Devasia, E. Eleftheriou, and S. O. R. Moheimani, "A survey of control issues in nanopositioning," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 5, pp. 802–823, 2007.

[58] J. Stergiopoulos and A. Tzes, "H_∞ closed-loop control for uncertain discrete input-shaped systems," *ASME. J. Dyn. Sys., Meas., Control*, vol. 132, no. 4, 2010.

[59] Y. Zhao and S. Jayasuriya, "Feedforward controllers and tracking accuracy in the presence of plant uncertainties," in *Proceedings of 1994 American Control Conference - ACC '94*, vol. 1, pp. 360–364 vol.1, 1994.

[60] A. Dautt-Silva and R. de Callafon, "Application of Dynamic Input Shaping for a Dual-Mirror System," *Journal of Dynamic Systems, Measurement, and Control*, 12 2021.

[61] J. Mattingley and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 50–61, 2010.

[62] V. Kapila, A. Tzes, and Q. Yan, "Closed-Loop Input Shaping for Flexible Structures Using Time-Delay Control1," *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 454–460, 12 1999.

[63] U. Boettcher, D. Fetzer, H. Li, R. A. de Callafon, and F. E. Talke, "Reference signal shaping for closed-loop systems with application to seeking in hard disk drives," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 2, pp. 335–345, 2012.

[64] J. R. Huey and W. Singhose, "Stability analysis of closed-loop input shaping control," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 305–310, 2005. 16th IFAC World Congress.

[65] M. Kenison and W. Singhose, "Concurrent design of input shaping and proportional plus derivative feedback control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 124, pp. 398–405, 07 2002.

[66] Z. Mohamed and M. A. Ahmad, "Hybrid input shaping and feedback control schemes of a flexible robot manipulator," *IFAC Proceedings Volumes*, vol. 41, pp. 11714–11719, 2008.

[67] T. Singh, "Minimax input shaper design using linear programming," *Journal of Dynamic Systems Measurement and Control-transactions of The Asme*, vol. 130, p. 051010, 2008.

[68] O. L. Mangasarian, "Uniqueness of solution in linear programming," *Linear Algebra and its Applications*, vol. 25, pp. 151–162, 1979.

[69] G. Appa, "On the uniqueness of solutions to linear programs," *Journal of the Operational Research Society*, vol. 53, no. 10, pp. 1127–1132, 2002.

[70] L. Van den Broeck, M. Diehl, and J. Swevers, "Embedded optimization for input shaping," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1146–1154, 2010.

[71] J. Scruggs, S. Lattanzio, A. Taflanidis, and I. Cassidy, "Optimal causal control of a wave energy converter in a random sea," *Applied Ocean Research*, vol. 42, pp. 1–15, 2013.

[72] T. Conord and T. Singh, "Robust input shaper design using linear matrix inequalities," in *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 1470–1475, 2006.

[73] O. N. Stamnes and R. A. De Callafon, "Time-optimal input shaping for discrete-time lti systems with application to seek profiles of a hdd system," in *Proceedings ASME ISPS Confeference*, pp. 146–148, 2007.

[74] D. W. Miller, S. R. Hall, and A. H. von Flotow, "Optimal control of power flow at structural junctions," in *1989 American Control Conference*, pp. 212–221, 1989.

[75] G. E. Dullerurd and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*. Springer, 2000.

[76] K. J. Keesman, *System Identification, An Introduciton*. Springer, 2011.

[77] J. Doyle, "Robust and optimal control," in *Proceedings of 35th IEEE Conference on Decision and Control*, vol. 2, pp. 1595–1598 vol.2, 1996.

[78] M. C. Pai and A. Sinha, "Increasing Robustness of Input Shaping Method to Parametric Uncertainties and Time-Delays," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, 02 2011.

[79] R. K. Yedavalli and C. R. Ashokkumar, "Time response bounds for linear parametric uncertain systems," *International Journal of Systems Science*, vol. 31, no. 2, pp. 177–188, 2000.

[80] A. Dautt-Silva and R. A. de Callafon, "Reference signal shaping for closed-loop systems with causality constraints," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021.

[81] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.

[82] A. Dautt-Silva and R. A. de Callafon, "Robust input shaping design for uncertain second-order dynamics systems," in *[Manuscript submitted for publication]*.

[83] D. K. de Vries and P. M. J. V. den Hof, "Quantification of uncertainty in transfer function estimation: a mixed probabilistic-worst-case approach," *Automatica*, vol. 31, no. 4, pp. 543–557, 1995.

[84] L. Ljung, *System Identification Theory for the User*. USA: Prentice-Hall, Inc., 1999.

[85] T. Söderström and P. Stoica, *System Identification*. USA: Prentice-Hall, Inc., 1988.

[86] R. A. de Callafon, D. Miller, J. Zeng, and M. Brenner, *Step Based Experiment Design and System Identification for AeroElastic Dynamic Modeling*.

[87] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[88] L. Y. Pao, "Strategies for shaping commands in the control of flexible structures," in *Proceedings Japan-USA-Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering*, 2000.

[89] L. Rupert, P. Hyatt, and M. D. Killpack, "Comparing model predictive control and input shaping for improved response of low-impedance robots," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 256–263, 2015.

[90] X. Jian-Xin, "A survey on iterative learning control for nonlinear systems," *International Journal of Control*, vol. 84, no. 7, pp. 1275–1294, 2011.

[91] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2017.