A Linear Programming Approach to the Optimal Design of Spline-Based Motion System Inputs

J. De Caigny, B. Demeulenaere, J. Swevers, J. De Schutter
K.U.Leuven, Department of Mechanical Engineering,
Celestijnenlaan 300 B, B-3001, Heverlee, Belgium
e-mail: **jan.decaigny@mech.kuleuven.be**

Abstract

For motion systems such as cam-follower mechanisms and loads driven by servo motors, this paper considers the design of system inputs that are continuous up to their M-th derivative and minimize some design criterion subject to user-defined constraints. This problem is tackled by optimizing a piecewise-linear, continuous parametrization (based on a large number of second-order B-splines) of the M-th derivative of the trajectory, which guarantees the continuity up to derivative M. As an application, the system input of a linear dynamic system is optimized to reduce the residual vibrations. Both non-robust and robust solutions of the resulting linear optimization problem are presented. For high-speed applications a trade-off in the degree of continuity at the boundaries is shown: for robustness against small plant perturbations this degree needs to be low, while for robustness against unmodeled higher dynamics this degree needs to be large. The numerical optimization results furthermore confirm and outperform earlier results for the considered benchmark.

1 Introduction

Designing system inputs is an important issue in the control of motion systems, for example in the reduction of follower residual vibrations in cam design. By taking certain a priori knowledge of the motion system dynamics into account, the system input can be shaped such that the motion output of the system behaves in a satisfactory manner. Several approaches towards system input design to reduce system vibration are presented in literature. Roughly speaking, these approaches can be categorized in two main classes, (i) filter design and (ii) input design.

The design of a filter is independent of the chosen reference trajectory for the motion output, denoted here as *motion reference*. Therefore filter design is a general technique: any motion reference can be convolved with the designed filter to create the appropriate system input. Most filter design methods however need an accurate model of the motion system, while robustness regarding unmodeled dynamics and uncertain or slightly changing system parameters is difficult to incorporate in the design. Singer and Seering [3] explicitly include robustness against uncertain system parameters in the filter design by analytically calculating a series of positive impulses to reduce system vibration. A drawback of this method is that a short move-time penalty is incurred (the length of the filter). This move-time penalty can be reduced by also allowing negative impulses, as shown by Singhose et al. [5], but actuator overcurrenting, which is not present if only positive impulses are used, and high-mode excitation can occur.

Contrary to filter design methods, input design depends on the chosen motion reference: a new system input has to be calculated for every different motion reference. Different approaches towards input design have been proposed. Kanzaki and Itao [2] select a polynomial trajectory for the motion output, based on user-defined specifications (boundary constraints and the characteristics of the residual vibration) and subsequently calculate the system input analytically. Srinivasan and Ge [6] use Berstein-Bézier harmonic curves to analytically design motion outputs, locally insensitive to parameter variations. To the authors' knowledge

Kwakernaak and Smit [1] were the first to formulate the input design as a numerical optimization problem. The jerk of the system input is parameterized and optimized with respect to some optimization criterion, subject to constraints on velocity and acceleration. Similarly, Chew and Chuang [4] parameterize the input acceleration as a polynomial and optimize the coefficients by minimizing the residual vibrations over a range of speeds. The framework presented hereafter is an input design method and generalizes the work of Kwakernaak and Smit.

The outline of the paper is as follows: Sec. 2 discusses the spline-based parametrization of the optimal system input. Section 3 shows how the optimal system input design is formulated as a linear optimization problem. In Sec. 4 the benchmark of Kanzaki and Itao [2] is tackled using the spline-based framework. First the system and the system's equations are given, then non-robust solutions are presented. Section 5 introduces a strategy to ensure robustness. Section 6 states the conclusions.

2 Parametrization of the optimal system input

2.1 General problem statement

The problem considered here is to design a trajectory

$$\theta(\tau) : [0, \tau^*] \rightarrow [\theta_{\text{min}}, \theta_{\text{max}}] \tag{1}$$

that is continuous up to its M-th derivative with respect to τ:

$$\theta(\tau) \in \mathcal{C}^{\text{M}}_{[0,\tau^*]}; \quad \theta(\tau) \notin \mathcal{C}^{\text{M+1}}_{[0,\tau^*]}, \tag{2}$$

and complies with boundary constraints on the position

$$\theta(0) = 0; \quad \theta(\tau^*) = 1; \tag{3}$$

as well as boundary conditions on the higher derivatives $1 \leq m \leq M_{BC} \leq M$:

$$\theta^{(m)}(0) = 0; \quad \theta^{(m)}(\tau^*) = 0, \tag{4}$$

where $M_{BC} \in \mathbb{N}$ is a user-defined constant. τ^* denotes the nondimensionalized time during which the motion has to be accomplished. The constants τ^* and 1 in (1)–(3) can be used without loss of generality: through nondimensionalization, many trajectory design problems can be formulated in this form.

There are, of course, many trajectories $\theta(\tau)$ that comply with (2), (3) and (4). This freedom can be exploited to design the trajectory so as to minimize some performance criterion, as well as to impose additional constraints. That is, the trajectory is found as the numerical solution of an optimization problem. This requires to adopt a parametrization for $\theta(\tau)$, as discussed in Sec. 2.2.

2.2 Parametrization of the optimal system input

One of the key features of the framework is the parametrization of $\theta(\tau)$. $\theta(\tau)$ is parametrized in an indirect manner by adopting a piecewise-linear, continuous parametrization for its M-th derivative $\theta^{(M)}(\tau)$, shown in Fig. 1. We assume that the available *knot locations* τ_k are equidistant[1]:

$$\tau_k = k \cdot \Delta = k \cdot \frac{\tau^*}{K}, \; k = 0 \ldots K,$$

[1] This assumption is made for reasons of convenience. The framework is however still valid if the available knot locations are not equidistant.

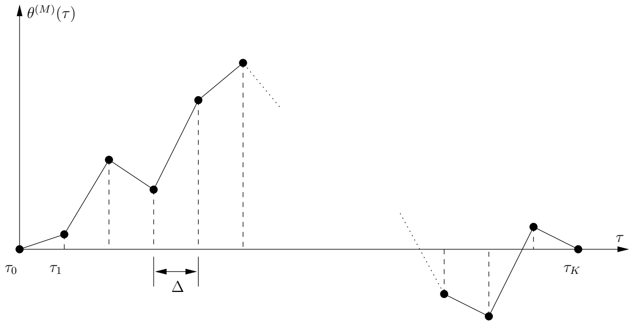


Figure 1: B-spline parametrization of $\theta^{(M)}(\tau)$. $\Delta = \tau^*/K$, where $K + 1$ denotes the number of available knots.

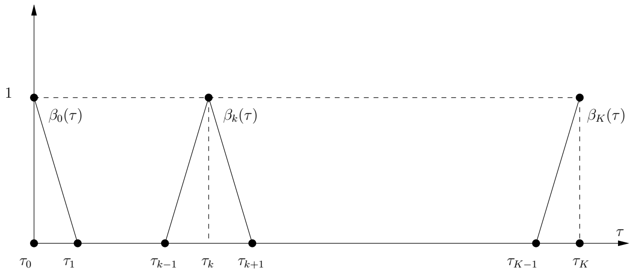


Figure 2: The hat functions $\beta_0(\tau)$, $\beta_k(\tau)$ $(1 \leq k < K)$ and $\beta_K(\tau)$.

where $K + 1$ denotes the number of available knot locations. Given its piecewise-linear parametrization, $\theta^{(M)}(\tau)$ itself is continuous, while its first derivative $\theta^{(M+1)}(\tau)$ is not, and hence it satisfies (2). If K is chosen to be big, say $K = 1024$ or $K = 2048$, this parametrization becomes more and more general in the sense that it becomes a more and more accurate representation of any function that satisfies (2). It is a generalization of the work of Kwakernaak and Smit [1], who restrict M to be 2 and consider very small values of K (typically $K = 30$).

Given its piecewise-linear parametrization, $\theta^{(M)}(\tau)$ is completely determined by its values at the knot locations τ_k. In fact, $\theta^{(M)}(\tau)$ can be written as

$$\theta^{(M)}(\tau) = \sum_{k=0}^{K} \theta^{(M)}(\tau_k) \cdot \beta_k(\tau), \tag{5}$$

where $\beta_k(\tau)$ denotes a B-spline of order 2, the so-called hat function, depicted in Fig. 2. A mathematical description of the hat-function is provided in Appendix A.

Equation (5) implies that $\theta^{(M)}(\tau)$ can be expressed as a linear combination of base functions $\beta_k(\tau)$, with coefficients that are given by $\theta^{(M)}(\tau_k)$. Since these base functions are B-splines of order two, the locations τ_k can be considered as spline knots or knot locations. In order to obtain $\theta(\tau)$ itself, (5) needs to be integrated M times. This implies that M integration constants C_m, $m = 1 \ldots M$ are required to unambiguously determine $\theta(\tau)$. Hence the number of independent parameters that determines $\theta(\tau)$ equals $K + 1 + M$:

- $K + 1$ values $\theta^{(M)}(\tau_k)$, $k = 0 \ldots K$;
- M integration constants C_m, $m = 1 \ldots M$.

These independent parameters are determined by numerically solving an optimization problem, of which a basic version is outlined in Sec. 3.

3 Basic optimization problem

In the previous section, the $K+1+M$ independent variables (optimization variables) are identified to be:

- $K+1$ values $\theta^{(M)}(\tau_k)$, $k=0\dots K$;
- M integration constants C_m, $m=1\dots M$.

Given that K is necessarily high to obtain a general parametrization, the number of optimization variables is high as well. Special measures are therefore needed to keep this problem numerically tractable. The approach chosen here is to only consider performance criteria and additional constraints that result in a linear optimization problem, for which very efficient numerical algorithms exist that are guaranteed to find the global optimum even if the number of variables is large, as is the case here. A linear program is an optimization problem in which a linear goal function is minimized subject to linear equality and inequality constraints.

3.1 Optimization variables

In order to formulate the optimization problem, we consider a set of optimization variables that is larger than the strictly minimal set of $K+1+M$ variables just discussed. That is, the following set of variables, gathered in the optimization variable vector $x \in \mathbb{R}^{(M+1)(K+1)}$ is used:

$$x = \left[\underbrace{\theta^{(M)}(\tau_0)\ \cdots\ \theta^{(M)}(\tau_K)}_{K+1} \mid \cdots \mid \underbrace{\theta^{(0)}(\tau_0)\ \cdots\ \theta^{(0)}(\tau_K)}_{K+1} \right]^T, \tag{6}$$

where $(\cdot)^T$ denotes a matrix transpose. The optimization variable x hence consists of the position $\theta(\tau)$ and all its derivatives up to the M-th derivative[2], evaluated at each of the considered knot locations τ_k, $k=0\dots K$. It is straightforward to show that, in order to impose that the elements of x be each others exact derivatives/integrals, the following set of MK *linear* equality constraints has to be imposed:

$$G \cdot x = 0,$$

where the matrix

$$G \in \mathbb{R}^{(MK)\times(M+1)(K+1)}$$

follows from elementary calculus. Note that the $(M+1)(K+1)$ optimization variables have to satisfy MK linear equality constraints, which results in $K+1+M$ independent optimization variables. This is exactly the number of independent variables identified in Sec. 2.2.

3.2 Optimization criterion

As already explained, the basic version of the optimization problem only considers performance criteria that give rise to linear programs. Two important criteria that are widely used and that give rise to a linear program are:

1. The infinity norm $||\cdot||_\infty$, defined as

$$||Ax-b||_\infty = \max_i |a_i^T x - b_i|, \tag{7}$$

where a_i denotes the i-th row of the $m \times n$ matrix A, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and b_i is the i-th value in b.

[2] Forward or backward finite differences can be used to determine the $M+1$-th derivative from the M-th derivative.

2. The one norm $||\cdot||_1$, defined as

$$||Ax-b||_1 = \sum_{i=1}^{m} |a_i^T x - b_i|, \tag{8}$$

where a_i denotes the i-th row of the $m \times n$ matrix A, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and b_i is the i-th value in b.

Using techniques from linear programming [7], it can be shown that both the infinity norm and the one norm can be formulated as a linear program in x (Appendix B) and an auxiliary, scalar variable w (for the infinity norm) or an auxiliary set of n scalar variables v_i, where i is an integer that satisfies $1 \leq i \leq n$ (for the one norm).

3.3 Constraints

In the basic framework proposed here, only linear equality and inequality constraints are considered, so as to obtain a linear program. A non-restrictive list of constraints that comply with this requirement is given below:

- *Precision point:* at a user-specified time instant $\tilde{\tau_k}$, a user-specified value $\tilde{d}$ is imposed on the m-th derivative of $\theta(\tau)$:
$$\theta^{(m)}(\tilde{\tau_k}) = \tilde{d}, \quad 0 \leq m \leq M+1.$$
Boundary constraints on $\theta(\tau)$ and its derivatives are a special case of a precision point, where $\tilde{\tau_k}$ equals either 0 or τ^*.

- *Bound constraint:* at user-specified time instants $\tau_k = k\Delta$, $k \in \mathbb{N}$, $0 \leq k_1 \leq k \leq k_2 \leq K$, user-specified lower bounds $\tilde{l}_k$ and upper bounds $\tilde{u}_k$ are imposed on the m-th derivative of $\theta(\tau)$:
$$\tilde{l}_k \leq \theta^{(m)}(\tau_k) \leq \tilde{u}_k, \quad 0 \leq m \leq M+1,\ 0 \leq k_1 \leq k \leq k_2 \leq K.$$
Using bound constraints on the system input, overcurrenting can be avoided in a straightforward manner.

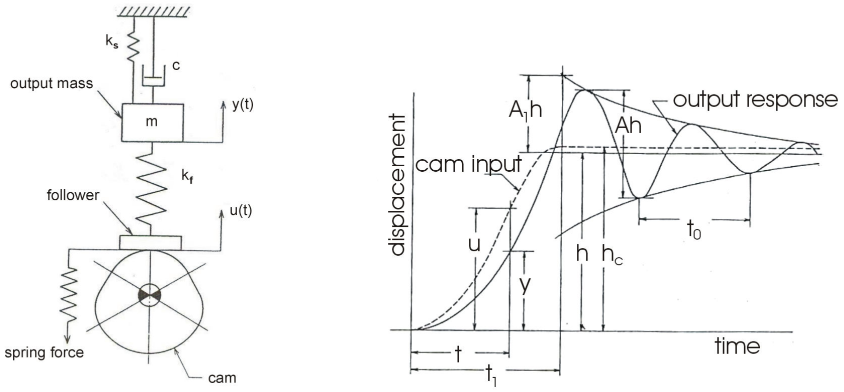
3.4 Resulting Optimization Problem

The resulting optimization problem is:

$$\begin{aligned} &\text{minimize} && ||\theta_{[0,K]}^{(M+1)}(\tau)||_\infty && \\ &\text{subject to} && \theta^{(m_i)}(\tilde{\tau_{k,i}}) = \tilde{d}_i && \forall i; \quad 0 \leq m_i \leq M+1 \\ & && \tilde{l}_{k,j} \leq \theta^{(m_j)}(\tau_{k,j}) \leq \tilde{u}_{k,j} && \forall j; \quad 0 \leq m_j \leq M+1, \\ & && && \quad\quad 0 \leq k_{1,j} \leq k_j \leq k_{2,j} \leq K. \end{aligned} \tag{9}$$

3.5 Numerical solution

Currently, the optimization problem is modeled in `matlab` and solved using `MOSEK`, a commercial code for solving convex programs. Generally, it is important that the numerical algorithm exploits the sparsity of the matrices involved, in order to increase the computational efficiency. For the examples presented hereafter (Sec. 4), the required CPU time is in the order of a few CPU seconds on a Pentium IV centrino 2GHz processor with 1GB RAM.



(a) Cam-follower mechanism with flexible cam-follower.

(b) Response $y(t)$ (solid line) of the cam-follower mechanism with flexible cam-follower to the system input $u(t)$ (dashed line).

Figure 3: Cam-follower. Reproduced from [2]

4 Case-study: input design for a cam-follower system

The case-study considered here is the cam-follower system introduced by Kanzaki and Itao [2] and represented in Fig. 3(a). It is a simplified representation of a typehead driving system to allow fine printing alignment of a high-speed mechanical teleprinter. The goal of the case-study is to design cam profiles such that the transient or residual typehead vibrations are reduced in such a way that the reduction is not affected by variation of the characteristics of the typehead driving system or the cam speed. The system is represented by a one-degree-of-freedom system as shown in Fig. 3(a), since the second natural frequency of the system is much larger than the first. Mass m $[kg]$ represents the equivalent mass of the typehead and of the transmission system, which includes linkages, belts and pulleys. Stiffnesses k_f $[N/m]$ and k_s $[N/m]$ are, respectively, the equivalent stiffness of the transmission system and stiffness of a restoring spring. The viscous damping coefficient c $[Ns/m]$ is due to bearings and guides. To maintain a linear system description, Coulomb friction and backlash are neglected in the system.

4.1 System and system equations

The equation of motion for the system shown in Fig. 3(a) is

$$m \cdot \ddot{y}(t) + c \cdot \dot{y}(t) + (k_f + k_s) \cdot y(t) = k_f \cdot u(t), \tag{10}$$

where

$$\begin{aligned} u(t) &: [0, t_1] \rightarrow [0, h_c] \\ y(t) &: [0, t_1] \rightarrow [0, h]. \end{aligned}$$

The cam profile determines the system input $u(t)$, whereas the follower motion $y(t)$ constitutes the motion output. After the end of the rise portion of the cam ($t \geq t_1$, see Fig. 3(b)), the output $y(t)$ is a free oscillation with viscous damping. A good measure of the amplitude of this residual vibration is the amplitude $A_1 h$ of its exponential envelope at $t = t_1$ (Fig. 3(b)).

After nondimensionalization, we obtain

$$\ddot{\gamma}(\tau) + 2\zeta(2\pi\lambda) \cdot \dot{\gamma}(\tau) + (2\pi\lambda)^2 \cdot \gamma(\tau) = (2\pi\lambda)^2 \cdot \theta(\tau),$$

where

$$\tau = \frac{t}{t_1}, t \in [0, t_1] \;\Rightarrow\; \tau^* = 1;$$
$$\gamma(\tau) = \frac{y(\tau \cdot t_1)}{h};$$
$$\theta(\tau) = \frac{u(\tau \cdot t_1)}{h_c} = \frac{u(\tau \cdot t_1)}{h} \cdot \frac{k_f}{k_s + k_f},$$

and $\omega_0^2 = (k_f + k_s)/m$, $\zeta = c/(2m\omega_0)$, $\lambda = t_1/t_0$ and $t_0 = 2\pi/\omega_0$. In this case, ω_0 equals 0.3808 rad/s (the natural period of the follower $t_0 = 2\pi/\omega_0$ was measured to be $t_0 = 16.5s$ [2]). The corresponding nondimensionalized, continuous-time state-space model equals:

$$\begin{bmatrix} \ddot{\chi}(\tau) \\ \dot{\chi}(\tau) \end{bmatrix} = \begin{bmatrix} -2\zeta \cdot (2\pi\lambda) & -(2\pi\lambda)^2 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \dot{\chi}(\tau) \\ \chi(\tau) \end{bmatrix} + \begin{bmatrix} (2\pi\lambda)^2 \\ 0 \end{bmatrix} \cdot \theta(\tau)$$

$$\gamma(\tau) = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\chi}(\tau) \\ \chi(\tau) \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \cdot \theta(\tau).$$

Here $\theta(\tau)$ is the nondimensionalized system input, $\gamma(\tau) = \chi(\tau)$ the nondimensionalized motion output and $\lambda = t_1/t_0$ the system's dimensionless resonance frequency.

4.2 Non-robust design

The objective in the non-robust design is to extinguish the residual vibration for the nominal system (with parameters λ_n and ζ_n). The nominal, non-robust design is repeated for $3 \times 2 \times 4 = 24$ different settings of the optimization:

- 3 different values of λ_n: $\lambda_n = [1.5, 2.0, 2.5]$;
- with and without damping: $\zeta_n = [0.00, 0.05]$;
- 4 different sets of boundary constraints:

$$\begin{cases} M_{BC} = 1: & \theta(0) = 0, & \theta(1) = 1, & \theta^{(1)}(0) = 0, & \theta^{(1)}(1) = 0, & \\ M_{BC} = 2: & \theta(0) = 0, & \theta(1) = 1, & \theta^{(i)}(0) = 0, & \theta^{(i)}(1) = 0, & i = 1, 2 \\ M_{BC} = 3: & \theta(0) = 0, & \theta(1) = 1, & \theta^{(i)}(0) = 0, & \theta^{(i)}(1) = 0, & i = 1, 2, 3 \\ M_{BC} = 4: & \theta(0) = 0, & \theta(1) = 1, & \theta^{(i)}(0) = 0, & \theta^{(i)}(1) = 0, & i = 1, 2, 3, 4. \end{cases}$$

All optimization problems, both non-robust and robust, are calculated using the following general settings:

- Cost function: the infinity norm: $||\theta_{[0,K]}^{(M+1)}(\tau_k)||_\infty$.
- Continuity $M = 4$: the derivative of the jerk is optimized as a piecewise-linear continuous function.
- Number of points $K = 512$: this implies a nondimensionalized sample-period $\Delta\tau = 1/512$.

To extinguish the residual vibration it is necessary that $A_1(\lambda_n, \zeta_n) = 0$, where $A_1(\lambda, \zeta)$ is given by (11):

$$A_1(\lambda, \zeta) = \frac{1}{\sin(\delta)} \sqrt{[\gamma(1) - 1]^2 + 2\cos(\delta)[\gamma(1) - 1]\left[\frac{\dot{\gamma}(1)}{2\pi\lambda}\right] + \left[\frac{\dot{\gamma}(1)}{2\pi\lambda}\right]^2}, \tag{11}$$

and δ denotes the arc cosine of ζ. From (11) it follows that a necessary and sufficient condition for $A_1 = 0$ is that

$$\begin{cases} \gamma(1) & = & 1 \\ \dot{\gamma}(1) & = & 0. \end{cases} \tag{12}$$

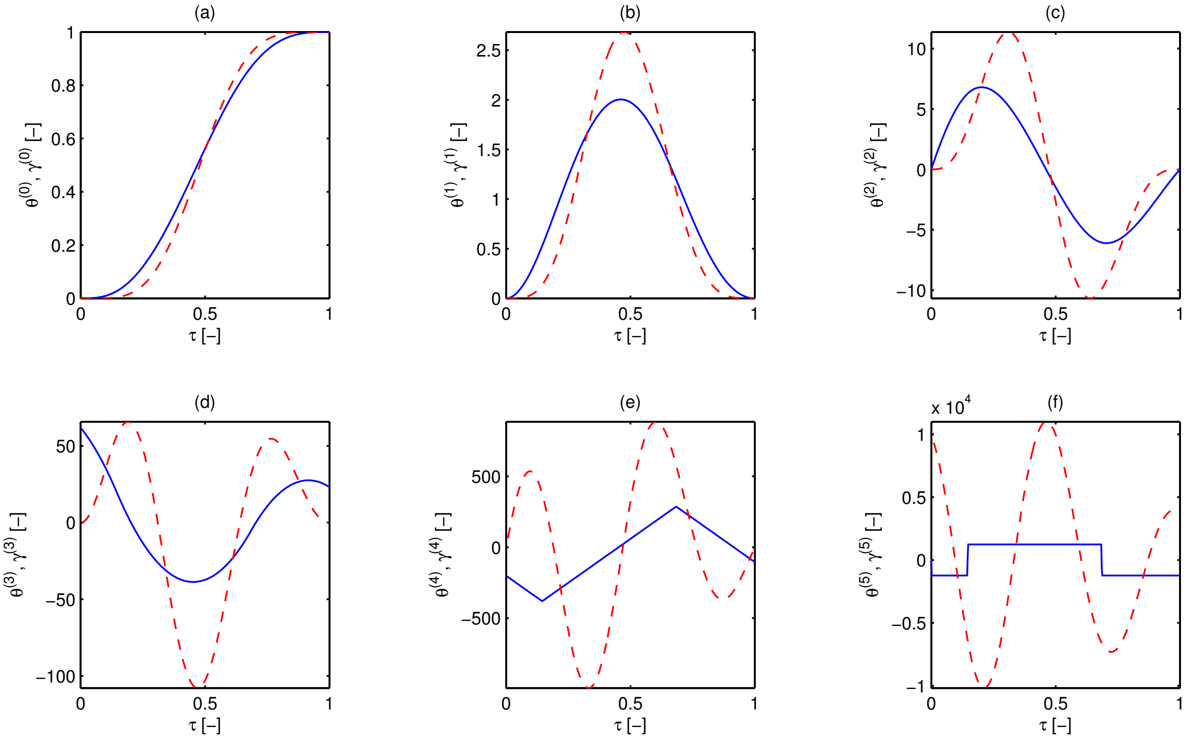


Figure 4: Non-robust design. Optimized input $\theta(\tau)$ (solid line) and output $\gamma(\tau)$ (dashed line) for $\lambda_n = 2$, $\zeta_n = 0.05$ and $M_{BC} = 2$.

Because the system is described by *linear* dynamics, constraints (12) are *linear* in the optimization variable x. Extending the basic, linear optimization problem of Sec. 3 with the constraints (12) therefore still results in a linear program.

Figure 4 shows a typical result from the optimization problem for $\lambda_n = 2$, $\zeta_n = 0.05$ and $M_{BC} = 2$. Fig. 4(a) shows the optimal system input and the motion output; the other figures show the velocity (Fig. 4(b)), acceleration (Fig. 4(c)), jerk (Fig. 4(d)), ping (Fig. 4(e)) and the derivative of the ping (Fig. 4(f)). In this case the optimization results in a spline with only two knots. From Fig. 4(a) and 4(b) it is clear that the residual vibration is extinguished since $\gamma(1) = 1$ and $\dot{\gamma}(1) = 0$.

To check the robustness of the optimal system input against variations of the system parameters λ_n and ζ_n the motion output and the amplitude of the residual vibration are calculated for systems with perturbed parameters: $(\lambda,\zeta) \neq (\lambda_n,\zeta_n)$. Figure 5 shows the amount of residual vibrations, measured by A_1 for λ ranging from 1 tot 10 for the 24 different optimized system inputs presented above. Several conclusions can be drawn from these results. (i) For every optimization, the residual vibration is extinguished for the nominal system: in every plot the amplitude A_1 drops to 0 for $\lambda = \lambda_n$. (ii) All curves have a clear roll-off at higher values of λ which is desirable since this ensures robustness against unmodeled higher dynamics. (iii) Higher values of M_{BC} give rise to a faster roll-off; this is intuitive since a higher value of M_{BC} leads to a smoother input; hence higher modes will be less excited. (iv) As a drawback, inputs with a higher M_{BC} value are, for small values of λ ($\lambda = 1.5$ and $\lambda = 2.0$), less robust against small plant perturbations: for $\lambda \in [\lambda_n; 3.5]$ Fig. 4(a), 4(b), 4(c) and 4(d) clearly show a larger ”bump” in the A_1 curve for the smoother inputs. Similar results are obtained for robustness against changing damping ($\zeta \neq \zeta_n$).

The rather counterintuitive fact that a larger M_{BC} value results in worse behavior for higher cam speeds (smaller values of λ) has already been reported in [4] and [6]. This implies that one must be very cautious when applying general rules of thumb (controlling initial and terminal conditions on velocity, acceleration, jerk, ...) to high-speed cam design. High-speed cam design is defined here as optimal input design for λ-values between 1 and 2.

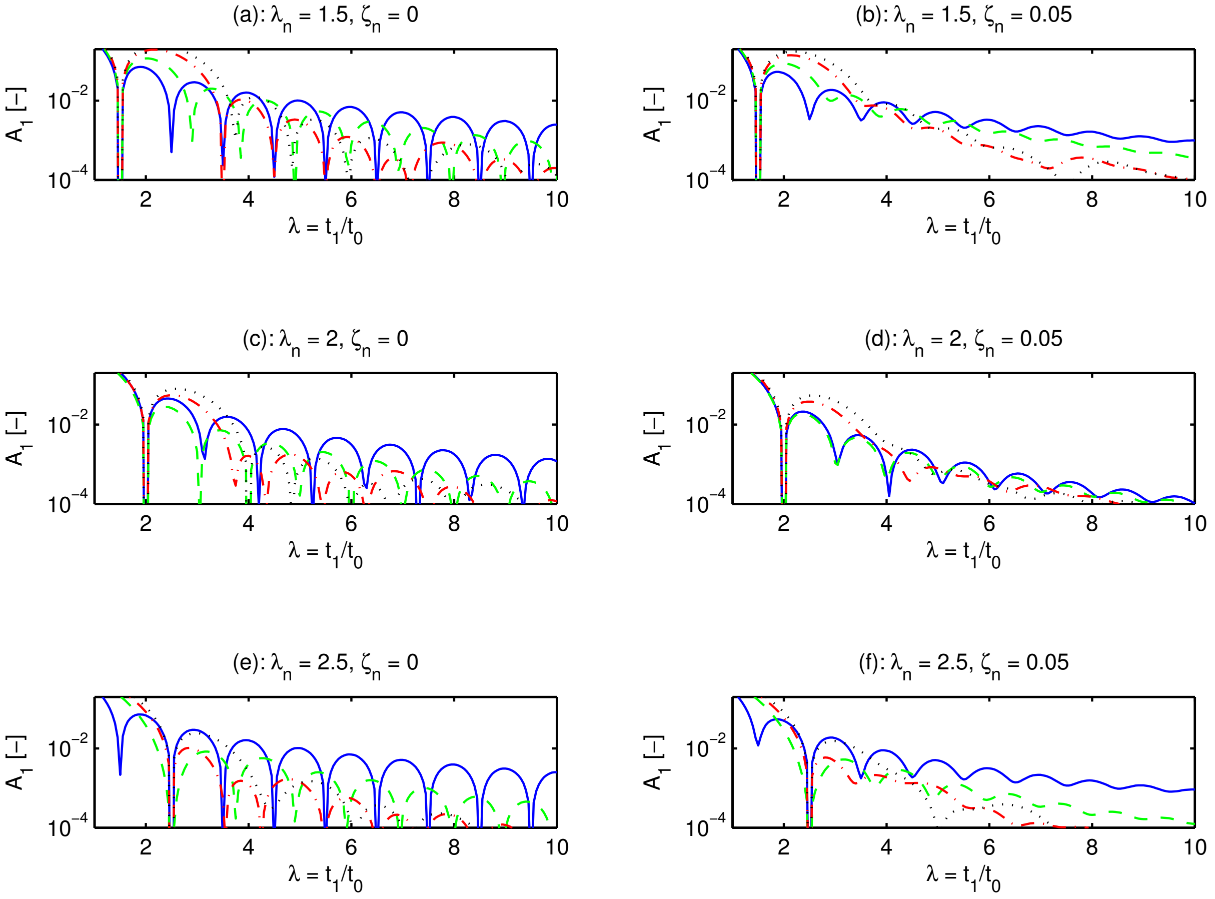


Figure 5: Robustness of non-robust design. Residual vibration for $\lambda_n = [1.5, 2.0, 2.5]$, $\zeta_n = [0.00, 0.05]$ and $M_{BC} = 1$ (solid line), $M_{BC} = 2$ (dashed line), $M_{BC} = 3$ (dash-dotted line), $M_{BC} = 4$ (dotted line).

In the approach developed by Kanzaki and Itao [2], the motion reference is a user-specified polynomial and the system input is analytically calculated so as to obtain the specified motion output. Kanzaki and Itao prove that in the undamped case the residual vibration is extinguished at (several) λ_i, related to the polynomial chosen as the motion reference, but independent of the chosen λ_n . Extinction of the residual vibration at $\lambda_i \neq \lambda_n$ doesn't appear if the system is damped. Figures 5(a), 5(c) and 5(e) show that this behavior also appears for the optimized splines. However, since no a priori specified polynomial is imposed as desired motion output, there is no control over the position of these λ_i values.

5 Robust design

To increase robustness, a natural extension of the non-robust design is to constrain (see Sec. 5.1) or minimize the residual vibrations over some range of λ and/or ζ. From (11) it is clear that $A_1^2(\lambda, \zeta)$ is a quadratic function of the optimization variable x. Here, an approximation for A_1 is used in terms of $\gamma(1)$ and $\dot{\gamma}(1)$ to maintain a linear program. The robust approach is used to optimize the system input for a system with parameters $(\lambda_n, \zeta_n) = (1.8, 0.1)$. Robustness is demanded in a range of $\lambda \in [1.6; 2]$, while ζ is kept to its nominal value. Boundary constraints are only imposed on the position: $\theta(0) = 0$ and $\theta(1) = 1$ ($M_{BC} = 0$). These settings allows us to compare the optimal system input to the dynamically compensated Bézier harmonic motion with second order robustness (developed in [6]).

Section 5.1 presents the linear approximation used to constrain A_1 which results in a sufficient, but not a necessary condition on $\gamma(1)$ and $\dot{\gamma}(1)$ to constrain A_1. Section 5.2 presents the results for the robust approach and shows that they outperform the analytical results by Srinivasan and Ge.

5.1 Constraining A_1

An exact constraint on A_1 for a pair (λ_i, ζ_i) has the form:

$$A_1(\lambda_i, \zeta_i) \leq \epsilon. \tag{13}$$

From (11) it can be shown that in order to guarantee (13) it is sufficient that:

$$\left\{ \begin{array}{lcl} |\gamma(1) - 1| & \leq & \frac{\epsilon}{\eta} \\ \left|\frac{\dot{\gamma}(1)}{2\pi\lambda_i}\right| & \leq & \frac{\epsilon}{\eta}, \end{array} \right. \tag{14}$$

where $\gamma(1)$ and $\dot{\gamma}(1)$ are the end position and velocity for the system with system parameters (λ_i, ζ_i) and η is given by (15) (for a proof, see appendix C).

$$\eta = \frac{1}{\sin(\delta)}\sqrt{(2 + 2\cos(\delta)}. \tag{15}$$

For every considered (λ_i, ζ_i) combination, (14) constitutes an inequality constraint, linear in the optimization variables. Therefore, adding (14), for every considered (λ_i, ζ_i) combination, to the basic optimization problem of Sec. 3 still results in a linear program.

In this case $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$, while all ζ_i are equal to ζ_n. Hence 41 constraints of the general form (14) are added to the basic optimization problem.

5.2 Results

Figure 5.2 shows a trade-off curve: the minimized infinity norm $||\theta^{(5)}(\tau)||_\infty$ is shown as a function of the demanded level of robustness ϵ. The trade-off curve clearly shows two knees (indicated A and B). Knee A is a sharp knee and is located at $\epsilon \approx 0.004$, knee B is less pronounced and is located at $\epsilon \approx 0.04$. A sharp knee implies a strong trade-off at this point: the marginal cost (the increase of $||\theta^{(5)}(\tau)||_\infty$) of increasing the robustness (decreasing ϵ) is large.

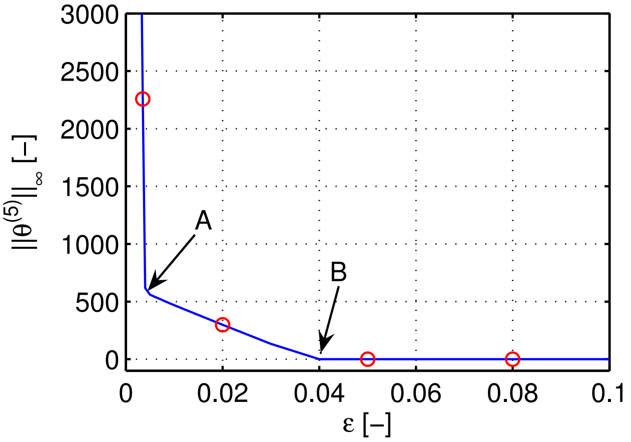


Figure 6: Robust design for $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$: trade-off between ϵ and $||\theta^{(5)}(\tau)||_\infty$. The dots 'o' indicate the ϵ-values selected in Table 1 and Fig. 7 and 8.

ϵ	$\|\|\theta^{(5)}(\tau)\|\|_\infty$	$\|\|\theta^{(4)}(\tau)\|\|_\infty$	degree	knots
0.08	0	0	3	0
0.05	0	42.5706	4	0
0.02	299.3260	201.1033	5	0
0.0035	2257.8446	839.0881	5	2

Table 1: Robust design for $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$: numerical results for constraining A_1 with $\epsilon = [0.08, \quad 0.05, \quad 0.02, \quad 0.0035]$.

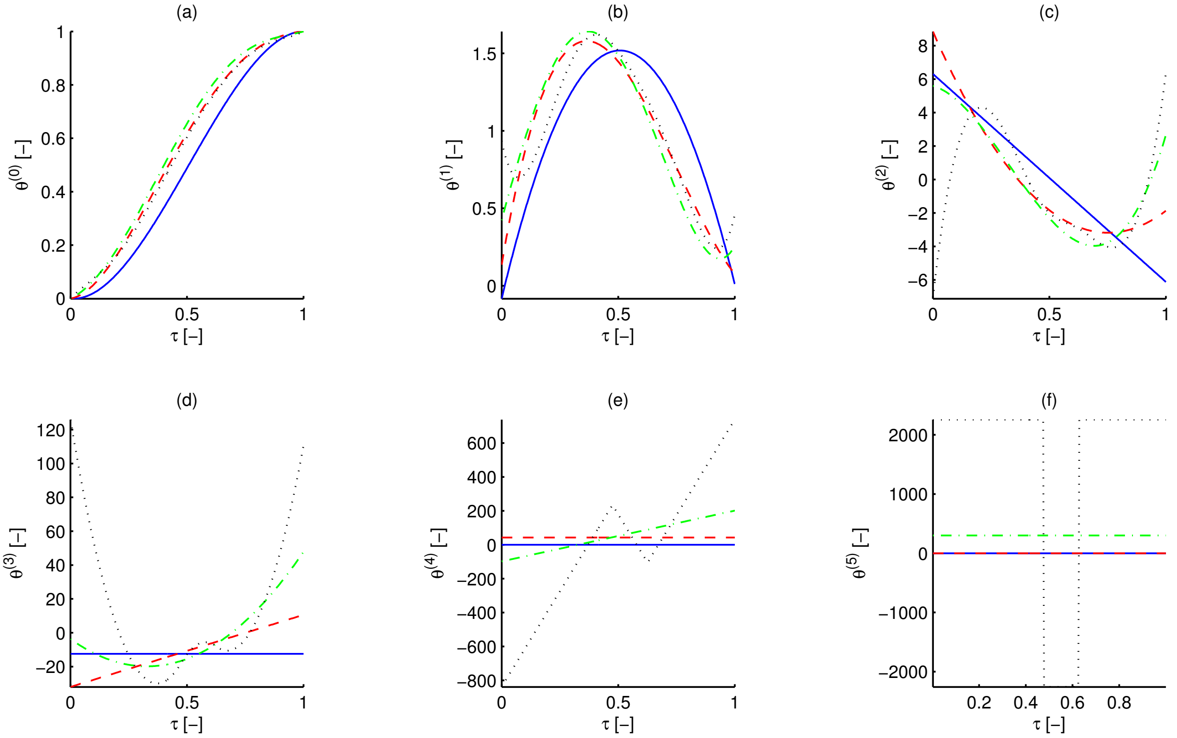
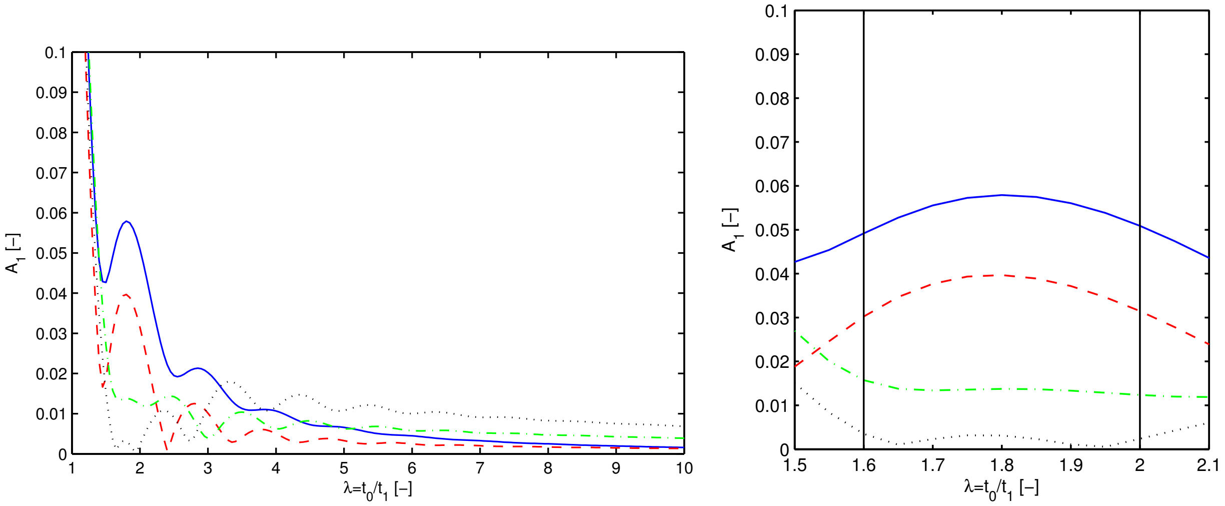


Figure 7: Robust design for $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$: optimized input trajectories for constraining A_1 with $\epsilon = 0.08$ (solid line), $\epsilon = 0.05$ (dashed line), $\epsilon = 0.02$ (dash-dotted line) and $\epsilon = 0.0035$ (dotted line).

To examine the behavior at knees A and B results are presented here for four particular problem instances: $\epsilon \in [0.08, \quad 0.05, \quad 0.02, \quad 0.0035]$. For these four optimization problems, Table 1 contains the numerical results, while Fig. 7 shows the optimized input trajectories and Fig. 8(a) and 8(b) the residual vibrations.

Figure 7 and Table 1 show what happens with the optimal system input as the required robustness increases (i.e. ϵ decreases from 0.08 to 0.0035). For $\epsilon = 0.08$ the minimized infinity norm $||\theta^{(5)}(\tau)||_\infty = 0$. In addition, the fourth derivative of the optimal system input $||\theta^{(4)}(\tau)||_\infty = 0$ too. This implies that to ensure a level of robustness $\epsilon = 0.08$, a third degree polynomial is sufficient. When ϵ is decreased to 0.05, the minimized cost remains $||\theta^{(5)}(\tau)||_\infty = 0$. The fourth derivative however has a constant non-zero value; hence to ensure the required robustness $\epsilon = 0.05$, a fourth degree polynomial is necessary. For $\epsilon = 0.02$ the optimal system input becomes a fifth degree polynomial, with a constant, non-zero fifth derivative $||\theta^{(5)}(\tau)||_\infty \neq 0$. Finally, for $\epsilon = 0.0035$, the optimal system input becomes a spline with two knots. This result generalizes as follows: knee B marks the transition from a fourth degree to a fifth degree polynomial, whereas knee A marks the transition from a fifth degree polynomial to a spline with two knots.

The robustness against small plant perturbations ($\lambda \in [1.5; 2.1]$) is shown in Fig. 8(b). Evidently, for a smaller ϵ the robustness is better in the interval $\lambda \in [1.6; 2]$, considered in the optimization. As a drawback, the robustness against unmodeled higher dynamics is worse: for lower values of ϵ, the residual vibrations at



(a) A_1 for $\lambda \in (1; 10)$

(b) A_1 for $\lambda \in (1.5; 2.1)$

Figure 8: Robust design for $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$: residual vibration for for constraining A_1 with $\epsilon = 0.08$ (solid line), $\epsilon = 0.05$ (dashed line), $\epsilon = 0.02$ (dash-dotted line) and $\epsilon = 0.0035$ (dotted line).

higher values of λ in Fig. 8(a) are shifted upwards. The reason for this behavior can be found in Fig. 7(d), 7(e) and 7(f): while all four optimal system inputs have the same degree of continuity at the boundaries ($M_{BC} = 0$), the spline clearly shows larger amplitudes for the higher derivatives. This results in an optimal system input that is less smooth, and therefore more prone to exciting unmodeled higher dynamics.

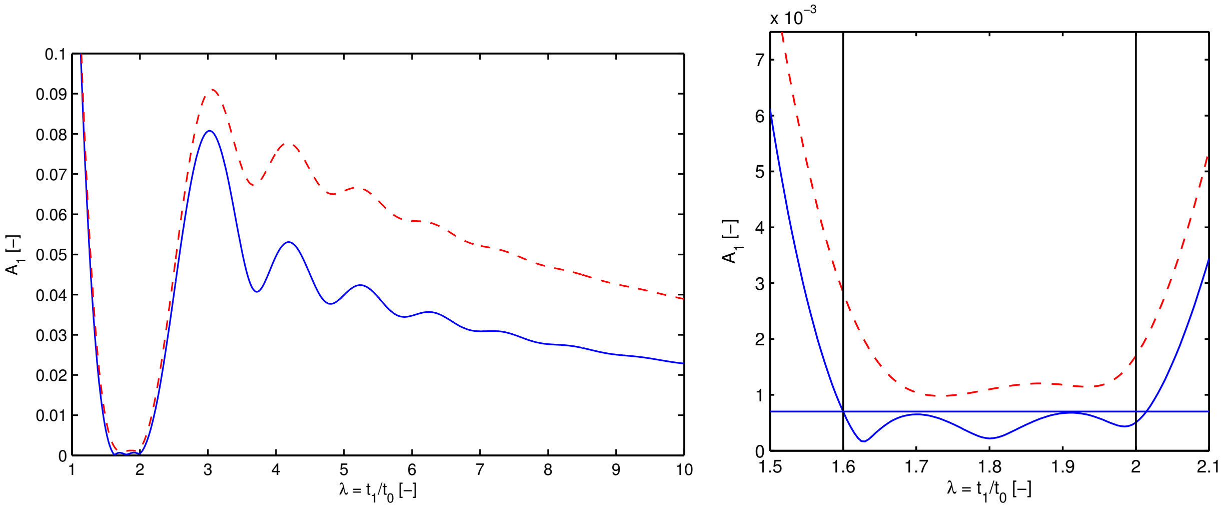
At a certain point, while further decreasing ϵ, the constraint on A_1 becomes too tight and the resulting problem is infeasible. For this benchmark the limit case is $\epsilon = 0.0007$; i.e. for $\epsilon < 0.0007$ the optimization is infeasible. For $\epsilon = 0.0007$ Fig. 9(a) and 9(b) compare the residual vibrations of the optimal system input to residual vibrations of the 2^{nd} order Bernstein Bézier harmonic (designed in [6]). The approach presented in [6] is to increase robustness by imposing $\partial^k A_1(\lambda, \zeta)/\partial\lambda^k = 0$ at $\lambda = \lambda_n$, for $k = 1, \ldots, n$. The 2^{nd} order Bernstein Bézier harmonic is calculated with $k = 1, 2$. Figure 9(a) shows that around the nominal λ value 1.8, the optimal system input and the Bézier harmonic give rise to little or no residual vibrations[3]. For higher λ, the optimal system input shows a much faster roll-off. The reason can be found in Fig. 10, which compares the optimal system input to the 2^{nd} order Bernstein Bézier harmonic. The higher derivatives (Fig. 10(c), 10(d), 10(e) and 10(f)) of the optimal system input all have lower maximal amplitudes compared to the Bernstein Bézier harmonic. Since the level of continuity at the boundaries is the same ($M_{BC} = 0$), the lower amplitudes guarantee a smoother input, which results in a system input that excites higher modes to a lesser extent. It is therefore concluded that the spline-based optimal system input perform better than the Bernstein Bézier harmonic curves proposed in [6]. This is mainly due to the fact that the spline-based parametrization is more general.

6 Conclusions

The spline-based framework proposed here proves to be an efficient and versatile tool in the design of system inputs for motion systems. Problems such as overcurrenting, which occur when negative impulses are used to shape the system input [5], can be dealt with in a straightforward manner: user-defined bound constraints can be added to the optimization problem to guarantee an input without overcurrenting.

For the considered benchmark non-robust and robust solutions are presented. Using the results of the non-robust design the importance of the degree of continuity at the boundaries (M_{BC}) is shown. A higher degree

[3] REMARK: the A_1-curve of the 2^{nd} order Bernstein Bézier doesn't become exactly 0 at $\lambda = 1.8$ in Fig. 9(b), as it should. This may be due to the finite precision of the coefficients published in [6].



(a) A_1 for $\lambda \in (1; 10)$ (b) A_1 for $\lambda \in (1.5; 2.1)$

Figure 9: Robust design for $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$: residual vibration for constraining A_1 with $\epsilon = 0.0007$ (solid line) and for 2^{nd} order Bézier harmonic (dashed line), reproduced from [6].

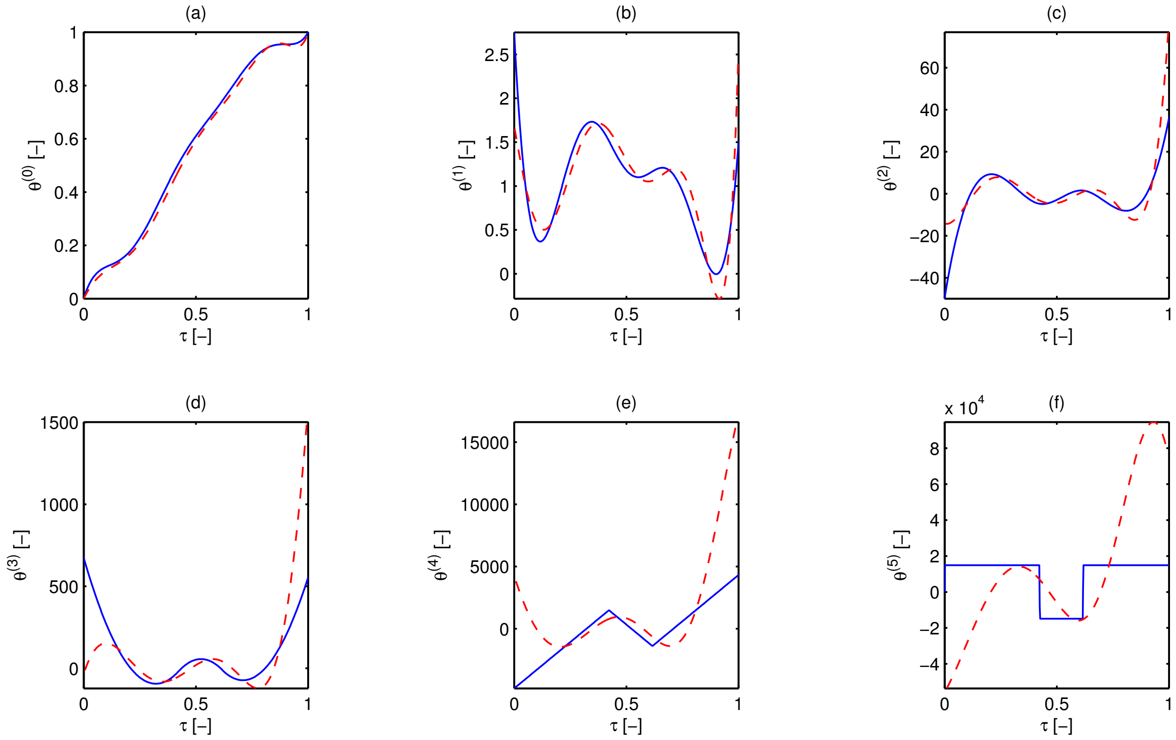


Figure 10: Robust design: $\lambda_i \in \Lambda = [1.6 : 0.01 : 2]$. Optimal system inputs for constraining A_1 with $\epsilon = 0.0007$ (solid line) and for 2^{nd} order Bézier harmonic (dashed line), reproduced from [6].

of continuity gives rise to a faster high-frequency roll-off, which implies more robustness against unmodeled higher dynamics. However for high-speed applications a lower degree of continuity is shown to be better. The results published here confirm the results of [4] and [6].

A constraint on the residual vibrations over a discrete set of system parameters is proposed as an approach to increase robustness. A clear trade-off is present between increased robustness and increased higher derivatives. The cost for increased robustness for system parameters close to the nominal values is a decrease in robustness against unmodeled higher dynamics. However, compared to the Bernstein Bézier harmonic curves proposed in [6], the optimal system input still shows more robustness against unmodeled higher dynamics and a similar robustness against small plant perturbations.

Acknowledgments

This work has been carried out within the framework of projects G.0446.06 and G.0462.05 of the Research Foundation - Flanders (FWO - Vlaanderen). Bram Demeulenaere is a Postdoctoral Fellow of the Research Foundation - Flanders. This work also benefits from K.U.Leuven-BOF EF/05/006 Center-of-Excellence Optimization in Engineering.

References

[1] H. Kwakernaak and J. Smit, *Minimum Vibration Cam Profiles*, Journal of Mechanical Engineering Science, Vol. 10, No. 3, (1968), pp. 219–227.

[2] K. Kanzaki and K. Itao, *Polydyne cam mechanisms for typehead positioning*, Transactions of the ASME, Journal of Engineering for industry Vol. 94 No. 1, (1972), pp. 250–254.

[3] N. C. Singer and W. P. Seering, *Preshaping command inputs to reduce system vibration*, Transactions of the ASME, Journal of Dynamic Systems, Measuremnt and Control Vol. 112, (1990), pp. 76–82.

[4] M. Chew and C. H. Chuang, *Minimizing residual vibrations in high-speed cam-follower systems over a range of speeds*, Transactions of the ASME, Journal of Mechanical Design Vol. 117, (1995), pp. 166–172

[5] W. E. Singhose, W. P. Seering and N. C. Singer, *Time-optimal negative input shapers*, Transactions of the ASME, Journal of Dynamic Systems, Measuremnt and Control Vol. 119, (1997), pp. 198–205.

[6] L. N. Srinivasan and Q. J. Ge, *Designing dynamically compensated and robust cam profiles with bernstein-bézier harmonic curves*, Transactions of the ASME, Journal of Mechanical Design Vol. 120, (1998), pp. 40–45.

[7] S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge University Press (2004).

A The hat function

In the case of equidistant knot locations τ_k, the hat function is defined by $(1 \leq k < K)$

$$\beta_k(\tau) = \begin{cases} 0 & \tau \leq \tau_{k-1} \\ \frac{\tau-\tau_{k-1}}{\Delta} & \tau_{k-1} \leq \tau \leq \tau_k \\ \frac{-\tau+\tau_k}{\Delta}+1 & \tau_k \leq \tau \leq \tau_{k+1} \\ 0 & \tau \geq \tau_{k+1} \end{cases} .$$

β_0 is defined as

$$\beta_0(\tau) = \begin{cases} 0 & \tau \leq 0 \\ \frac{-\tau}{\Delta}+1 & 0 \leq \tau \leq \Delta \\ 0 & \tau \geq \Delta \end{cases} .$$

β_K is defined as

$$\beta_k(\tau) = \begin{cases} 0 & \tau \leq \tau_{K-1} \\ \frac{\tau-\tau_{K-1}}{\Delta} & \tau_{K-1} \leq \tau \leq \tau_K \\ 0 & \tau \geq \tau_K \end{cases} .$$

B Infinity and one norm minimization

- The infinity norm minimization (also known as Chebyshev or minimax minimization):

  $$\text{minimize} \quad ||Ax-b||_\infty = \max_i |a_i^T x - b_i| \ ,$$

  where $x \in \mathbb{R}^n$, is equivalent to the following linear program:

  $$\begin{array}{ll} \text{minimize} & w \\ \text{subject to} & -w \leq a_i^T x - b_i \leq w, \quad \forall i, \quad 1 \leq i \leq m \end{array}$$

  where a_i denotes the i-th row of the $m \times n$ matrix A, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, b_i is the i-th value in b and $w \in \mathbb{R}$.

- The one norm minimization:

  $$\text{minimize} \quad ||Ax-b||_1 = \sum_{i=1}^{m} |a_i^T x - b_i|,$$

  where $x \in \mathbb{R}^n$, is equivalent to the following linear program:

  $$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{m} v_i \\ \text{subject to} & -v_i \leq a_i^T x - b_i \leq v_i, \quad \forall i, \quad 1 \leq i \leq m, \end{array}$$

  where a_i denotes the i-th row of the $m \times n$ matrix A, $x \in \mathbb{R}^n$, b, $v \in \mathbb{R}^m$ and b_i, v_i the i-th value in b, resp. v.

C An approximation for $A_1(\lambda_i, \zeta_i) \leq \epsilon$

In this section it is proven that constraints (14) form a sufficient, but not necessary condition for (13).

Proof: If both inequality constraints in (14) hold then from (11) it is clear that:

$$A_1 \leq \frac{1}{\sin(\delta)} \sqrt{\left(\frac{\epsilon}{\eta}\right)^2 + 2\cos(\delta)\left(\frac{\epsilon}{\eta}\right)^2 + \left(\frac{\epsilon}{\eta}\right)^2} = \frac{\epsilon}{\eta}\frac{1}{\sin(\delta)}\sqrt{2+2\cos(\delta)}. \tag{16}$$

With η defined as in (15), this implies the original constraint (13): the linear approximation is sufficient.

When the original constraint holds, the linear constraints (14) do not necessary hold: for example when $A_1 = \epsilon$ and $\dot{\gamma}(1) = 0$, the position $\gamma(1)$ can be easily calculated to be:

$$|\gamma(1) - 1| = \frac{\epsilon}{1/\sin(\delta)} \geq \frac{\epsilon}{\eta}.$$

This proves that the linear approximation is not necessary.