2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009



# Performant Design of an Input Shaping Prefilter via Embedded Optimization

Lieboud Van den Broeck, Moritz Diehl and Jan Swevers

*Abstract*— **Traditional input shaping filters are linear mappings between the reference input and the system input. These filters are often unnecessarily conservative with respect to input and output bounds if multiple references with different amplitudes are applied. This conservatism is due to the off-line computation of the prefilter. This paper presents an on-line input prefilter design approach to overcome this conservatism. The resulting prefilters are called predictive prefilters because the on-line design is based on the model predictive control (MPC) framework. By theoretical considerations, simulation results and experimental results, it is shown that this new prefilter is at least as good as traditional prefilters, and can result in substantial gains in settling time. Tests show that a 30 % decrease in settling time is possible in a classical application.**

## I. Introduction

In the last decades, a lot of effort has been put into developing efficient input shaping prefilters, see e.g. [1]–[4]. These prefilters are computed off-line and applied to the system afterwards. Input or actuator saturation is avoided by considering during the design worst case reference inputs, that is a step reference with maximum amplitude [1], [5]. Hence, these input filters yield conservative results with respect to the input range and response speed if the reference input is more smooth and/or has a smaller amplitude.

Since the nineties, a new control approach has gained a lot of attention, namely model predictive control [6], [7], mostly in control of slow processes. A MPC-controller estimates at every time step the system state, and computes subsequently the optimal input to be applied to the system. This optimal input is computed based on the state of the system, the imposed reference to the system and constraints on input, output, and states. The mapping of the reference to an ideal input is hence computed on-line. An important challenge for MPC in motion control applications is a high sampling rate, in the range of kHz, at which these on-line optimization problems have to be solved.

For some (fast) applications, it is cumbersome to estimate the states, as is necessary for MPC. This can be due to the cost of the sensors, the impossibility to introduce these sensors into the system or because of the delay they introduce. In these cases, it is better to have open loop control like input shaping instead of closed loop control like MPC, especially if the system dynamics are well known.

This paper presents a new approach for the design of prefilters. This approach is a hybrid between the classical prefilters and the MPC-framework. The result is a real prefilter, in the sense that no feedback of the system is incorporated, but the prefilter realizes a nonlinear instead of a linear mapping between the reference and system input because of the on-line optimization. The prefilter takes into account the constraints on inputs and outputs during the real motion of the system and not during a worst case scenario and hence avoids the conservatism of traditional input shapers.

This paper is organized as follows. Section II starts with a short introduction to both input shaping and MPC. Section III describes the design framework of the predictive prefilter and discusses its practical implementation. Section IV compares the new prefilter design with the standard input shaping prefilter for some scenarios to illustrate its advantages. Section IV-D validates the proposed framework with experimental results. The paper ends with directions for future research and conclusions.

## II. Input Shaping and MPC: the basics

This section gives an overview of the basic idea behind input shaping and how it can be extended and efficiently solved. The section continues with basic elements of MPC.

### A. Input Shaping

Traditional input shaping prefilters are finite impulse response (FIR) filters that convert reference point-to-point motion commands such that very little or no residual vibration occurs upon arrival at the endpoint. These filters introduce a short delay, known as move-time penalty, equal to the duration of the prefilter's impulse response. Singer en Seering [1] developed the first popular input shaping design approach. This approach is based on analytic expressions for the response of a continuous-time second-order system to the following sequence of $K+1$ impulses $f_k$ at time locations $t_k$:

$$f(t) = \sum_{k=0}^{K} f_k \delta(t - t_k) \quad (1)$$

Lieboud Van den Broeck is funded by a Ph.D. fellowship of the Research Foundation - Flanders (FWO - Vlaanderen). This work benefits from K.U.Leuven-BOF EF/05/006 Center-of-Excellence Optimization in Engineering and the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office (DYSCO).

L. Van den Broeck and J.Swevers are with the Department of Mechinal Engineering, division PMA, K.U.Leuven, 3001 Leuven, Belgium lieboud.vandenbroeck@mech.kuleuven.be

M. Diehl is with the Department of Electrical Engineering, K.U.Leuven, 3001 Leuven, Belgium

978-1-4244-4524-0/09/$25.00 ©2009 AACC

The required zero residual vibration corresponds to the following set of nonlinear equations [1]:

$$\sum_{k=0}^{K} f_k \mathrm{e}^{-\zeta\omega_n(t_K-t_k)} \sin(t_k\omega_n\sqrt{1-\zeta^2}) = 0, \quad (2a)$$

$$\sum_{k=0}^{K} f_k \mathrm{e}^{-\zeta\omega_n(t_K-t_k)} \cos(t_k\omega_n\sqrt{1-\zeta^2}) = 0, \quad (2b)$$

where $\zeta$ and $\omega_n$ are the damping coefficient and the eigenfrequency of the system. The minimum number of impulses, required to solve (2a)–(2b), is two, which corresponds to $K = 1$ in (1). A minimal delay is introduced when $t_0 = 0$ and $t_1 = T_0/2$, where $T_0$ denotes the period of the damped natural eigenfrequency. By requiring that all the impulses of the FIR-filter are positive and sum to one, the design guarantees that if the original reference does not drive the actuators of the system in saturation, the new shaped inputs will not do this either. Numerous extensions to the basic technique exist e.g. to make it more robust [2], to make it faster by allowing negative impulses [5], or including a priori information about the desired trajectory [3].

It has been shown that the input shaping problem can be reformulated in a more general linear programming framework [8], [9]. This generalization considers the system and prefilter as one system of which the total impulse response has to be shaped; i.e., the input shaping prefilter design considers:

$$g(t) = f(t) \otimes h(t), \quad (3)$$

with $f(t)$ the impulse response of the prefilter, $h(t)$ the impulse response of the system, $g(t)$ the impulse response of the system and prefilter together, and $\otimes$ the convolution operator. Consequently, imposing zero vibration with minimal move-time corresponds to the requirement that the impulse response $g(t)$ is as short as possible. This no-vibration constraint has to be acquired without violating input or output constraints. This is usually attained by considering a worst-case scenario, or by including a priori information about the desired trajectory. Based on (3), [8] and [9] show that the input shaping design can be reformulated as a convex optimization problem which has a guaranteed global optimum [10].

### B. Model Predictive Control

MPC is an advanced control technique which is also known as moving or receding horizon control. It determines the system input by solving on-line, at every sampling time, an open-loop optimal control problem, based on the current state of the system. Taking the current state into account, the controller is capable of accounting for disturbances. The optimization generates an optimal input sequence for a specified finite time horizon, however, only the first input $\bar{u}_l = u_0$ is applied to the system. In its simplest setting, the problem to be solved at each time step $l$ has the following structure:

$$\min_{x(.),y(.),u(.)} \sum_{k=0}^{K} [(y_k - y_{\mathrm{ref},k})^T Q (y_k - y_{\mathrm{ref},k}) + u_k^T R u_k], \quad (4a)$$

subject to constraints:

$$x_0 = \bar{x}_l, \quad (4b)$$

$$x_{k+1} = f(x_k, u_k), \quad (4c)$$

$$y_k = h(x_k, u_k), \quad (4d)$$

$$g(x_k, u_k) \geqslant 0 \quad k \in [0, K], \quad (4e)$$

where $K$ is the considered time horizon and $\bar{x}_l$ the measured or estimated system state at time $l$. The optimization is based on a model that predicts the system behavior for a sequence of control variables (4b)–(4d) and acknowledges bounds on inputs, outputs and internal states (4e). The input is optimized considering the system input and output weighted with the matrices $R$ and $Q$ respectively. MPC emerged first in the process industry, due to the less stringent real time requirements [11]. Nowadays there exist algorithms that are able to solve this type of problems at kHz sampling frequencies for systems with up to 10-20 states and prediction horizons $K$ of up to 10-12 time steps ahead yielding that MPC becomes feasible for mechatronic applications [12], [13].

## III. The Predictive Prefilter

This section presents the MPC inspired design of an input shaping prefilter. It discusses also the practical implementation of the designed framework, and ends with theoretical considerations. The predictive prefilter is developed in discrete time and is based on a discrete time presentation of the system.

### A. Design of the predictive prefilter

Like all prefilters, the predictive prefilter transforms the reference setpoint $y_{\mathrm{ref},l}$ to a system input. This input has to fulfill the following requirements:

- the system must drive the system output to the desired setpoint as fast as possible,
- the prefilter must prevent residual vibrations at the system output as much as possible,
- constraints on inputs, outputs, and possibly system state variables have to be respected.

The predictive prefilter satisfies these requirement by the on-line solution of an optimization problem which results in a nonlinear mapping between the reference and the inputs of the system.

The considered problem $P(\bar{x}_l, y_{\mathrm{ref},l})$ at time step $l$ is:

$$\min_{x_0,\ldots,x_K,u_0,\ldots,u_K,y_0,\ldots,y_K,K} K \quad (5a)$$

$$\text{subject to: } x_0 = \bar{x}_l, \quad (5b)$$

$$x_{k+1} = f(x_k, u_k), \quad (5c)$$

$$y_k = h(x_k, u_k), \quad (5d)$$

$$0 \leq g(x_k, u_k) \text{ for } k = 0 \ldots K, \quad (5e)$$

$$x_K = f(x_K, u_K), \quad (5f)$$

$$y_K \equiv y_{\mathrm{ref},l} \quad (5g)$$

Constraints (5c)–(5d) describe the system dynamics. For a given input, these constraints completely determine the system behavior for a given initial state, hence constraint

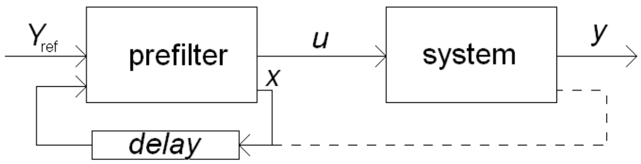


Fig. 1. Comparison of MPC (dashed line) and the predictive prefilter (full line). MPC relies relies on the measured system state, while the predictive prefilter does not feed back the system state, it relies only on the state of the model that is used in the optimization formulation.

(5b). This initial state $\bar{x}_l$ is in the prefilter formulation not obtained by system measurements as in real MPC, but by simulation of the system. This makes the prefilter a real prefilter, which can easily and without additional sensors be added to an existing system. Constraints (5f)–(5g) are the most important ones. They impose the system to arrive at the desired setpoint at time K without residual vibrations. The prefilter will minimize the time $K$ required to arrive at this desired state (5a). Constraint (5e) presents all constraints on the inputs, outputs and states. This allows to impose directly the input constraints of the system, and not indirectly as is done in traditional input shaping. Hence, this avoids to be unnecessarily conservative, see also [3]. For some specific cases, e.g. with fast changing setpoint, this new method allows to be restrictive enough, while traditional input shapers can not handle this and result in over-currenting [5]. This framework can easily be extended to add constraints on the system overshoot, maximum speed, rate of input change, … Also robustness with respect to parameter uncertainty can be included, see paragraph V. Figure 1 illustrates the difference between MPC and the predictive prefilter.

To summarize:

The inputs of the prefilter at each discrete-time step $l$ are:

- state of the model of the system $\bar{x}_l$
- desired setpoint $y_{\text{ref},l}$

The outputs of the prefilter at each discrete-time step $k$ are:

- input of the system $\bar{u}_l$
- state of the system $x_{l+1}$

The following program is solved at each discrete-time step $l$:

1) Solve the parametric optimization problem (5a)–(5g). Obtain $\bar{u}_l$ as first control input of the solution. This problem depends on $\bar{x}_l$ and $y_{\text{ref},l}$.
2) Predict the next state of the model system $x_{l+1} = f(x_l, u_l)$.

### B. Problem structure and efficient solution

Problem (5a)–(5g) is in general a nonlinear mixed-integer program. If the system model is linear and the constraints are convex, the total problem is quasi convex and can normally be solved by a sequence of linear feasibility problems which bisect on the time $K$. Every bisection step, a linear feasibility problem with a certain end time $K$ is solved. Depending on the feasibility, the end time $K$ is reduced or increased, until the optimal time $T^*$ is found. However, because a bisection approach consists of a series of LP-problems to be solved at each time step, a reformulation of the problem is proposed. Instead of directly optimizing problem (5a)–(5g), a slightly different problem is defined. By using a weighted sum of $l_\infty$-norms of the tracking error:

$$\min \sum_{k=0}^{K} \|y - y_{\text{ref}}\|_\infty c^k, \; c > 1, \tag{6}$$

instead of the objective (5a) and endpoint constraints (5f)–(5g), one can obtain similar results, at the cost of solving only one optimization problem. The sum of $l_\infty$-norms is used as this results in a relative heavy weighting of the small residuals [10], and hence in an efficient removal of these small residuals, i.e. the residual vibrations. By using an exponential weighting $c^k$, the exponential decay of the residual vibrations is compensated for, and hence the time-optimal behavior can almost perfectly be enforced at a much lower computational cost of only 1 optimization problem instead of multiple feasibility problems. This reformulation should be handled with care though, because it may result in badly scaled problems if $c$ is too large. For the on-line-solution, it is important that the considered time-horizon is not too long as the computation-times of the solution algorithms scale at least linearly but usually worse with the horizon length. A good choice of the horizon is the length of a classical input shaping prefilter.

### C. Theoretical considerations

*Theorem 1:* For any single reference step input, the predictive prefilter gives the fastest possible transition.

*Proof:* The solution of problem $P(\bar{x}_l, y_{\text{ref},l})$ for a single reference step gives a time-optimal trajectory, i.e. the shortest possible. Because the system model in simulation and optimizer are the same, the principle of optimality of subarcs [14] applies to the following problems, so the first computed trajectory, which was optimal, is followed. ■

## IV. Validation

This section discusses the numerical and experimental validation of the developed predictive prefilter. The experimental test system is discussed in section IV-A. Numerical validation in section IV-B and IV-C is based on a linear model of this test setup and clearly shows the advantages of the predictive prefilter. The design is also experimentally validated in section IV-D.

### A. Test setup

First, the predictive prefilter is compared with a classical prefilter for a benchmark problem. The test-setup consists of a two-DOF mass-spring-damper system. Fig. 2 shows a picture and a schematic drawing of the setup. The system is excited by a position controlled hydraulic piston with position $p(t)$. The system input is the reference signal for

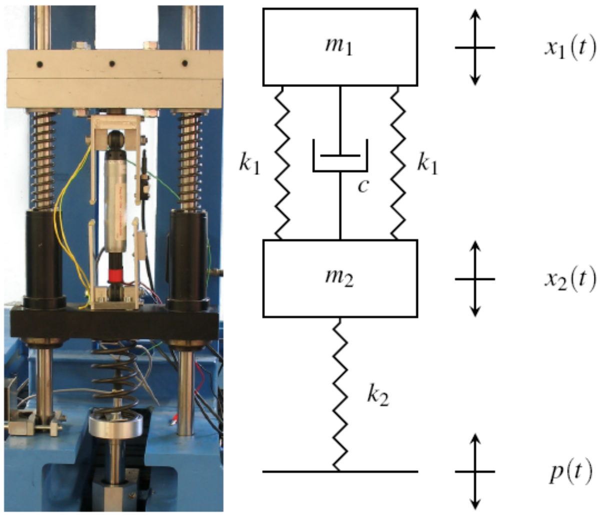


Fig. 2. Picture and schematic drawing of the test-setup.

TABLE I
POLES OF THE FIFTH ORDER SYSTEM

| frequency [rad/s] | damping [-] |
|---|---|
| $\omega_0 = 2.6205 \times 2\pi$ | $\zeta_0 = 0.157\%$ |
| $\omega_1 = 7.7926 \times 2\pi$ | $\zeta_1 = 0.293\%$ |
| 1 real pole at 214 | / |

the piston position controller. The position of the upper mass $x_1(t)$ is chosen as the system output. A fifth-order continuous-time state space model is identified for this system based on frequency response function measurements that are obtained from a multisine excitation with a frequency content between 0.1Hz and 10Hz [15]. To apply the developed framework, this model is transformed to discrete time with a sample period of $T_s = 0.01s$. This model contains two pairs of complex conjugated poles originating from the two flexible modes of this system, and one real pole that is introduced by the band limited piston position controller (Table I). The input of this system is limited to 1 V, and the output is limited to a displacement of 1cm.

### B. Single step benchmark

For benchmarking, the predictive prefilter is compared with the results of a classical prefilter, for a maximal step, i.e. a reference step of $1cm$. The considered prefilter is designed following the procedure of [8]. In this procedure, negative impulses are allowed as in the predictive prefilter, and such that no saturation takes place for a reference step of $1cm$. Figure 3 shows this reference (full line) and also the resulting output, both with a classical input shaping prefilter (dashed line) as with a predictive prefilter (dotted line). This figure shows that the predictive prefilter can reproduce the classical input shaping prefilters for a maximal allowable step. Fig. 4 shows this even more clearly by showing the outputs of the inputshaping prefilter (full line) and the predictive prefilter (dashed line) for this reference. These filter outputs are not

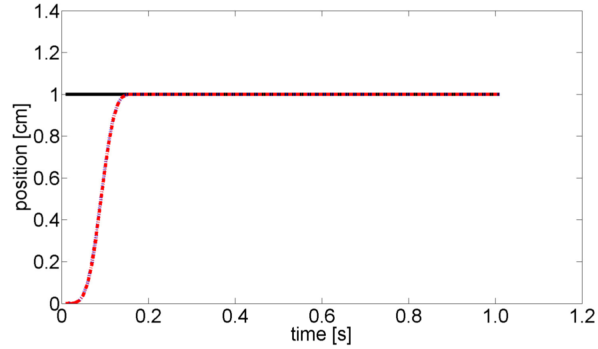


Fig. 3. Output of the system with a classical input shaping prefilter (dashed line) and a predictive prefilter (dotted line), for a full step reference (full line). Remark that both outputs coincide.

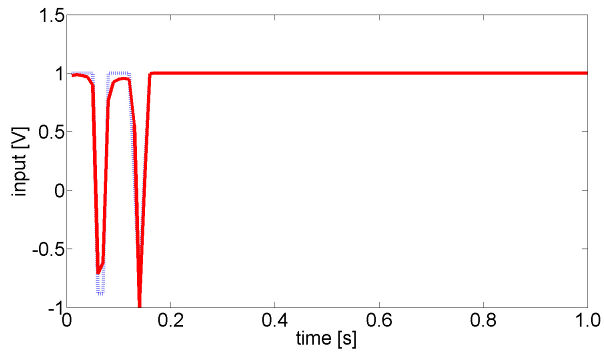


Fig. 4. Input of the system with a classical input shaping prefilter (full line) and a predictive prefilter (dotted line).

completely the same. This is due to the freedom left in the optimization problem, as the real optimal time can only be approximated from above in a discrete setting.

### C. Multiple step

This paragraph shows the benefits of the predictive prefilter for some other references. The desired setpoint change is not a maximal step, but a sequence of smaller steps. Figure 5 shows the desired setpoints for the system (solid line). Also the resulting behavior with a classic input shaping prefilter (dashed line) and a predictive prefilter (dotted line) is shown. This shows clearly the superior behavior of the predictive prefilter. Reductions of the settling time up to 30% are obtained. Figure 6 shows the respective inputs of the system, i.e. with input shaping (full line) and predictive prefiltering (dotted line), and this shows that the reduction is due to the more efficient use of the allowable input range. The classic prefilter scales its input for the maximal step with the desired step, while the predictive prefilter makes full use of the available actuator possibilities. This allows faster reaction times and hence shorter settling times. If there are unmodelled higher modes in the system however, they will be more excited than with classical input shaping prefilters.

The next example illustrates this advantage even more clearly. Figure 7 shows the same behavior if the next reference set point is requested before the previous is fully

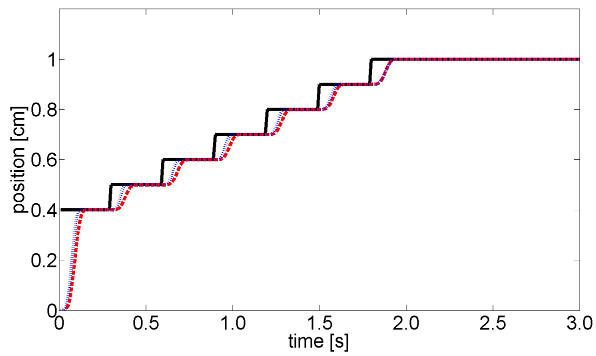


Fig. 5. Output of the system with a classical input shaping prefilter (dashed line) and a predictive prefilter (dotted line). The reference is not one big step, but a series of smaller steps (solid line).

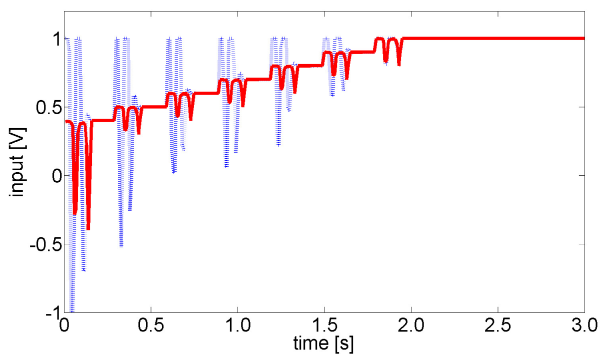


Fig. 6. Input of the system with a classical input shaping prefilter (solid line) and a predictive prefilter (dotted line).

executed. Both the behavior with the predictive prefilter (dotted line) and with a classical input shaping prefilter (dashed line) are shown for a reference trajectory of 3 steps (full line).

Figure 8 presents a last simulation example which shows the better performance of the new prefilter. This example requests a reference trajectory (full line) with a step up from 0 to 1 at time $0s$ followed by a step down to 0.3 at time $0.1s$. A traditional input shaping prefilter drives the actuators of the system into saturation which results in unacceptable behavior of the system (dashed line). The predictive prefilter (dotted line) which can take into account the real constraints of the system has no problems with this saturation, and hence is also for this case more efficient.

### D. Experimental

After numerical validation, the new input shaping prefilter design is also experimentally validated. The considered test setup is the same as presented in section IV. A dSPACE board DS1103 performs the computations and controls the input signals of the system. To apply this prefilter, it is of the utmost importance, that the presented optimization problem be solved in the available sampling-time of $0.01s$. To execute the computations, the dSPACE board contains a 1GHz processor with 90Mb RAM.

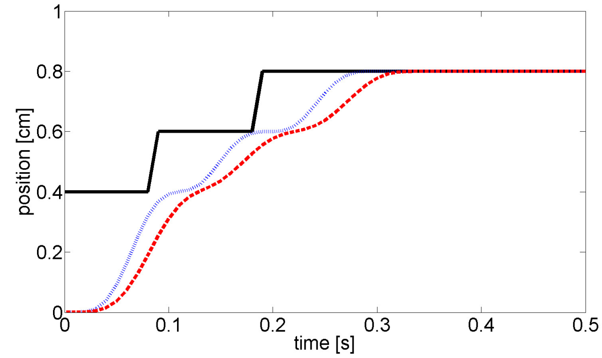


Fig. 7. Output of the system with a classical input shaping prefilter (dashed line) and a predictive prefilter (dotted line). The reference is a series of 3 steps (solid line), where a new setpoint is requested before the previous is attained.

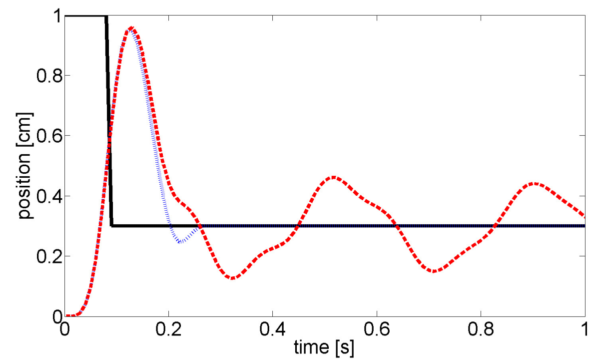


Fig. 8. Output of the system with a classical input shaping prefilter (dashed line) and a predictive prefilter (dotted line). The reference is a series of 2 steps (solid line), where the second step is reversed compared with the first.

The considered optimization problem is formulated with a horizon of 15 time steps. The input is limited to $\pm 1V$ and the slew rate of the input is limited to $\pm 0.1V$. This last constraint is necessary due to nonlinear oil-flow in the hydraulic actuators. The overshoot is limited to 5%. This results in a problem formulation with 30 variables, 120 constraints and 60 bounds. This problem can not be easily solved in the available time, and specialized algorithms have to be employed which exploit the problem formulation. Therefore, the problem is modeled in Simulink® and solved with qpOASES [13]. To exploit this algorithm fully, a light quadratic weighting is added to both the inputs and the error. Furthermore, the hotstarting ability is exploited. For this formulation, the maximal CPU time for computation is $2.9ms$, and the mean CPU time is $0.5ms$. The maximal time of $2.9ms$ is required when a reference step is requested, but still safely below the $10ms$ sampling time, making real-time application of the method possible.

Fig. 9 shows the system response, i.e. the position $x_1$, if the reference (solid line) is applied with a traditional input shaping prefilter (dashed line) and with the designed predictive prefilter (dotted line). This shows that in the experimental setup the same gains, i.e. 30% less settling time, can be attained as in the simulations. The apparent

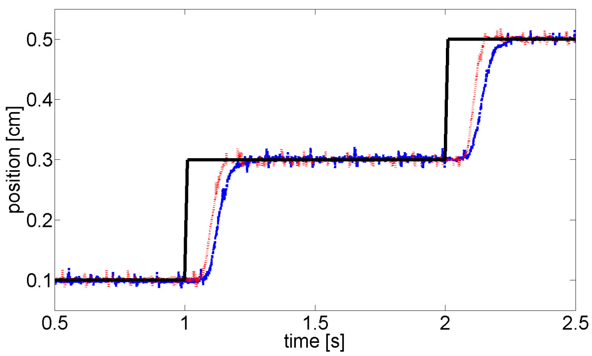


Fig. 9. Validation of the predictive input shaping prefilter design: desired motion (full line), and system response with a classical input shaping prefilter (dashed line) and with the predictive prefilter (dotted line).

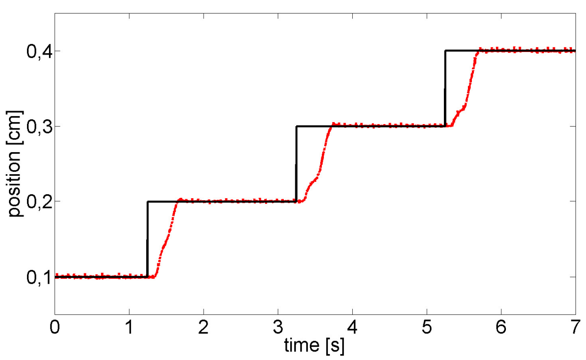


Fig. 10. Robust design: a reference (full line) is applied to the perturbed system. This reference is robustly prefiltered and the resulting response (dashed line) is still very good.

error around the endpoints is accountable to noise.

## V. Robustness

As an extension to the basic framework, robustness is introduced in the developed framework. By requiring that not only the nominal system, but also perturbed systems arrive at the desired setpoint, the resulting prefilter gains robustness. This can be introduced by taking the $l_\infty$ norm over the outputs of $X$ systems instead of 1 system, see (6). To stabilize the designed prefilter, a limited amount of vibration should be allowed e.g. 0.5% of the total displacement. This is necessary to allow for numerical rounding errors. This can be imposed by requiring the $l_\infty$ norm in (6) to have a minimal value. Fig. 10 shows the experimental behavior (dashed line) of a perturbed test setup, i.e. masses are added to the setup and the system is controlled by a robust prefilter. This clearly demonstrates that the proposed prefilter design is also capable of producing robust prefilters, which are solvable in real-time, i.e. $10ms$ sampling time, on embedded hardware.

## VI. Conclusion

This paper presents a new design method for input shaping prefilters. This new design produces a prefilter by the online solution of an optimization problem which takes into account the real constraints of the system and hence results in a nonlinear mapping between the sequence of setpoints and the sequence of inputs for the system instead of a linear mapping. Theoretical considerations show that this predictive prefilter is at least as good as the linear input shaping prefilter but normally outperforms this prefilter in terms of time optimality. Simulations show that the predictive prefilter reproduces a benchmark problem. Furthermore, it is also shown that the prefilter is more efficient than a classical input shaping prefilter if the requested setpoint does not correspond to a worst case scenario. Gains of 30% in settling-time are easily achieved. The simulation results are confirmed by experimental results on a quarter-car test setup. Hereby it is shown that sampling times of $0.01s$ are no problem with the presented design. It is also shown that robustness easily can be introduced in the developed framework.

## References


[1] N. C. Singer and W. P. Seering. Preshaping command inputs to reduce system vibration. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 112:76–82, 1990.

[2] W.E. Singhose, W. P. Seering, and N.C. Singer. Residual vibration reduction using vector diagrams to generate shaped inputs. *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, pages 654–659, 1994.

[3] M. D. Baumgart and L. Y. Pao. Discrete time-optimal command shaping. *Automatica*, 43:1403–1409, 2007.

[4] O.J.M. Smith. Posicast control of damped oscillatory systems. In *Proceedings of the IRE*, pages 1249–1255, 1957.

[5] W.E. Singhose, W. P. Seering, and N.C. Singer. Time-optimal negative input shapers. *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, 119:198–205, 1997.

[6] J.M. Maciejowski. *Predictive control with constraints*. Prentice Hall, 2000.

[7] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Schockaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[8] L. Van den Broeck, J. De Caigny, G. Pipeleers, and al. A linear programming approach to robust input shaping. In *Proceedings of the AMC*, pages 80–85, Trento, 25-27 March 2008.

[9] L Sungyung, H. D. Stevens, and J. P. How. Input shaping design for multi-input flexible systems. *Journal of dynamic systems, measurement, and control*, 121:443–447, 1999.

[10] S. Boyd and L. Vandeberghe. *Convex Optimization*. Cambridge University Press, 2004.

[11] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[12] A.G. Wills, D. Bates, A.J. Fleming, B. Ninness, and S.O.R. Moheimani. Application of mpc to an active structure using sampling rates up to 25khz. In *Proceedings of the Conference on Decision and Control and European Control Conference ECC'05*, Seville, 2005.

[13] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, in press.

[14] R. Bellman. *Dynamic Programming*. Princeton Univ. Press, 1957.

[15] R. Pintelon and J. Schoukens. *System Identification: A Frequency Domain Approach.* IEEE Press, 2001.