

Exploiting Actuator Limits with Feedforward Control based on Inverse Models

Manuel Gräber
TLK-Thermo GmbH
Hans-Sommer-Str. 5, 38106 Braunschweig, Germany

Abstract

Feedforward control based on inverse dynamic plant models (linear or nonlinear) is a suitable method to enhance set-point tracking performance of control systems. In reality actuators always have limits, but limiting functions can not be inverted. A common approach to handle this issue is to invert the unlimited plant model and detune the feedforward filter in order to stay always in between the actuator limits. This approach causes a loss in performance for rapid set-point changes, because the actuator range is not entirely used. In this article a rather simple but powerful method is presented, which overcomes this performance issue for many types of plant models. Actuator limits are fully exploited, and the obtained trajectories are close to optimal ones. Simulation and measurement results demonstrate the usability of the proposed feedforward structure.

Keywords: set-point tracking; model inversion; actuator saturation; nonlinear control; model-based control; anti-windup

1 Introduction

Assuming the dynamic behaviour of a plant is known, and a control task is to drive the plant output to a given set-point or track a given trajectory. A reasonable idea is to use the known plant behaviour to compute the needed plant input. Mathematically spoken a dynamic plant model is inverted. The original physical model computes the model output as reaction to the plant input. Whereas the inverse model computes the plant input for a given output. Modelica as equation-based modeling language provides a powerful possibility to automatically generate inverse dynamic models. In many cases inverse models can directly be derived from forward models. Details about model inversion with Modelica can be found in [11].

There are different ideas, where to place the inverse

model inside a control loop. In this paper we concentrate on the most obvious variant: using the inverse model in the feedforward path.

Feedforward control based on inverse dynamic plant models (linear or nonlinear) has been proven to be a suitable method to enhance set-point tracking performance of control systems. Successful applications have been reported in [8, 5].

But there is one issue, that often dramatically decreases the performance of feedforward designs: actuator limits. Every real-world control system has limited actuators. That means the control signal $u(t)$ is constrained by

$$u_{\min} \leq u(t) \leq u_{\max} \quad (1)$$

The problem with Equation (1) is, that it is not invertible. A common approach is, to not include input constraints and invert an unconstrained plant model. But what happens if the unconstrained inverse model in a feedforward path computes plant inputs which violate constraints (1)? A straightforward solution is sketched in Figure 1. A filter F and the inverse plant model \tilde{G}^{-1} is used to compute an input signal out of reference signal r . A limiter finally cuts off the unconstrained input signal v at its limits and feeds the input signal u to the plant G .

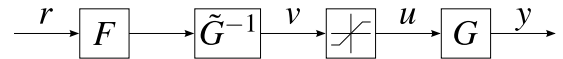


Figure 1: Feedforward control with inverse plant model and actuator limitation.

It's easy to imagine, that if we apply a step jump to r this can lead to very large values of v at the first time points. Its magnitude of course depends on the inverse model and the filter. Theoretically, for most plant models a step jump of r with no filter ($F = 1$) would even lead to an infinite impulse of v . But large values of v generate *actuator saturation*, that means u is equal u_{\max} or u_{\min} . The resulting difference between

u and v causes a inconsistency between the states of the inverse plant model and the plant. Which directly means performance decrease of the feedforward control. This effect can be interpreted as *windup*, which is well known for integral controllers. In Figure 4 example simulation results are plotted, where saturation of u leads to unwanted oscillation of plant output y .

Well-known anti-windup techniques for feedback controllers, which can be described in state-space form, are conditioning [7] or the more general observer-based approach described in [1]. Although these schemes are developed for feedback controllers they could also be used for feedforward control. In fact the proposed method in this article shows similarities to observer-based approaches, but as described in Section 2 there is one important difference.

Straightforward approaches to avoid windup for inverse models as feedforward are: filter detuning or application dependent trajectory planning (e.g. set-point ramps). Drawbacks of these methods are the lack of generality and a loss of performance – at least if simple approaches are used. In [11] handling actuator limits is briefly discussed, although it is not the main scope of this article. As one possibility it is suggested to provide feasible reference trajectories by online or offline solving of dynamic optimization problems. This approach provides high performance potential. Due to its optimal character actuator limits can be fully exploited. But dynamic optimization is usually a non-trivial task.

In [10] actuator saturation is addressed with feedback from the plant according to *internal model control*. This general approach has the advantage to be not application specific. And with only one remaining tuning parameter its simple to parameterize. As a result the output of the inverse model v stays within its limits but does not fully exploit them.

Instead of detuning the filter F or using application dependent trajectory planning, a more general method to avoid windup is presented in this article. The basic concepts and ideas are described in Section 2. No theoretical proof of stability is given, but numerical experiments in Section 4 and real-world experiments in Section 5 demonstrate the usability of the proposed feedforward scheme. Benefits of this approach are:

- applicable to nonlinear (SISO) systems
- easy to set up using Modelica
- only one tuning parameter: filter frequency
- full exploitation of actuator limits, almost time-optimal trajectories

2 Feedforward with Actuator Limits

The basic idea of the new control structure is sketched in Figure 2. The original feedforward system in Figure 1 is complemented with an internal feedback loop. Similar to anti-windup schemes for integral controllers [1] the actuator saturation signal $v - u$ is fed back with a constant gain k to the input signal.

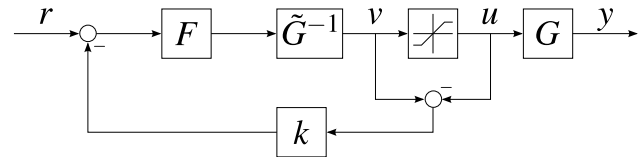


Figure 2: Proposed feedforward control with internal anti-windup feedback.

If the unconstrained signal v is beyond its limits the negative feedback will decrease the windup effect and the constraint violation. The actual performance of this compensation strongly depends on the filter F and the gain factor k . In the following we will design F and k in a way, that there will practically no constraint violation of v .

The basic idea is to use a very high gain k to drive constraints violations to zero. See [12] for details on high gain feedback for nonlinear systems. In order to achieve stability of the feedforward loop with high gains, the open loop $F\tilde{G}^{-1}$ has to fulfill a few requirements:

1. $F\tilde{G}^{-1}$ must be stable.
2. $F\tilde{G}^{-1}$ must have stable zero dynamics (minimum phase for linear systems).
3. Relative degree of $F\tilde{G}^{-1}$ must be equal zero.

Condition 3 actually is stricter than needed. For stability of the high gain feedback loop a relative degree equal one would be sufficient. But numerical experiments show, that the feedforward performance is much better, if we restrict the relative degree to zero. Resulting in a direct algebraic feed through from F 's input to \tilde{G}^{-1} 's output and a nonlinear implicit algebraic equation of the overall feedforward system of equations (see Section 3). This is a crucial feature of the proposed implicit method, which distinguishes this method from observer-based explicit anti-windup schemes described in [1].

The open loop requirements can be transformed into requirements for F and \tilde{G} . Condition 1 and 2 hold also for the classical feedforward control of Figure 1 and

lead to the requirements that plant model G must be stable and must have stable zero dynamics. Condition 3 can easily be fulfilled if we choose the order of the filter F equal to the relative degree of \tilde{G} .

1. \tilde{G} must have stable zero dynamics.
2. \tilde{G} must be stable.
3. Order of Filter F must be equal to the relative degree of \tilde{G} .

It is important to note, that these requirements are not (yet) theoretical proven sufficient conditions. They are motivated by the theory of high-gain feedback and successful numerical experiments with various linear and nonlinear plant models.

3 Feedforward control using Modelica

Modelica provides a convenient possibility to generate feedforward control laws based on the proposed method. Due to its equation-based design the error-prone plant model inversion is handled by the Modelica compiler. Using the free Modelica Standard Library (MSL) and a given forward plant model, this task can even be done graphically. In Figure 3 the graphical representation of an example feedforward control system is shown. The plant is modelled as linear second order system, but it could be any other SISO Modelica model. This structure corresponds exactly to Figure 2. Plant model inversion is easily defined by exchanging input and output with the MSL block `InverseBlockConstraints`. See [11] for further details on model inversion with Modelica.

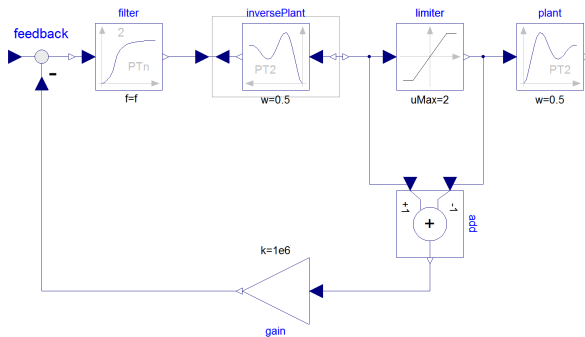


Figure 3: Graphical Modelica implementation of the proposed feedforward structure.

According to the requirements in Section 2 the chosen filter order is two – identical to the plant model's

relative degree. This means there is a direct algebraic relation between the filter input and the output of the inverse plant model. Using the notation of Figure 2 this relation can be expressed as function f with

$$v = f(r - k(v - u), \mathbf{x}) \quad (2)$$

where \mathbf{x} is the vector of internal states of $F\tilde{G}^{-1}$. Together with Equation (1) we obtain a nonlinear system of equations

$$v = f(r - k(v - u), \mathbf{x}) \quad (3a)$$

$$u = \begin{cases} u_{\max} & v > u_{\max} \\ u_{\min} & v < u_{\min} \\ v & \text{else} \end{cases} \quad (3b)$$

which can be reduced to one unknown v . To conclude this analysis: the resulting anti-windup feedforward control law is a system of differential and algebraic equations (DAE), with at least one nonlinear algebraic equation.

A straight-forward way to test this control law at a real plant is to use Functional Mock-up Unit (FMU) [3] export capabilities of Modelica tools. The resulting FMU for model-exchange comes with an embedded solver for implicit equations. The drawback is, that one has no control of the internal iterative solution process. But at least for prototyping purposes this approach works very well. In Section 5 a FMU exported from Dymola is used as controller at a real plant. With TLK-Thermo's FMI Suite the FMU is imported into the data acquisition and control environment LabView.

4 Simulation Experiments

In this section we look at two different linear example plants, modelled with the Modelica Standard Library. We study the effect of different filter parameters, which is the remaining degree of freedom of the proposed feedforward control. Finally optimal control trajectories are presented and compared to the obtained ones.

4.1 Plant Models and Simulation Setup

The chosen example plant models are two stable but differently damped second order systems. Angular frequency is 0.5 and gain is 1.0 in both cases. The resulting transfer functions are of the form

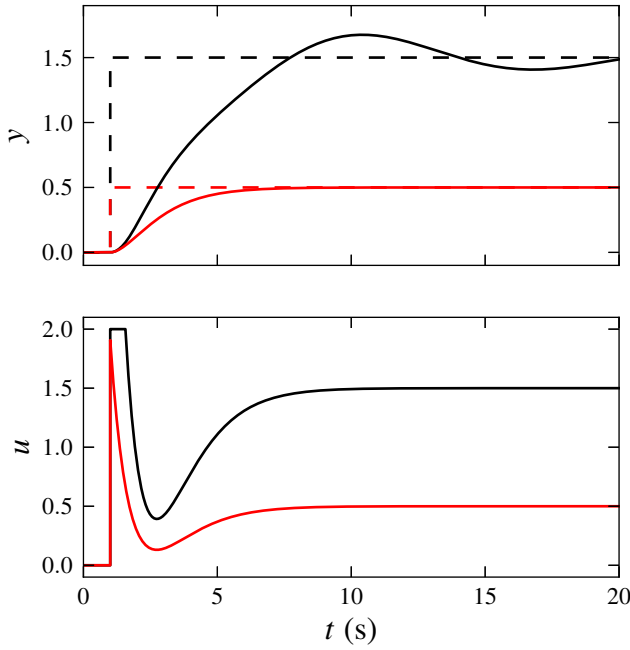


Figure 4: Plant input and output response to reference steps of different height. Conventional feedforward design.

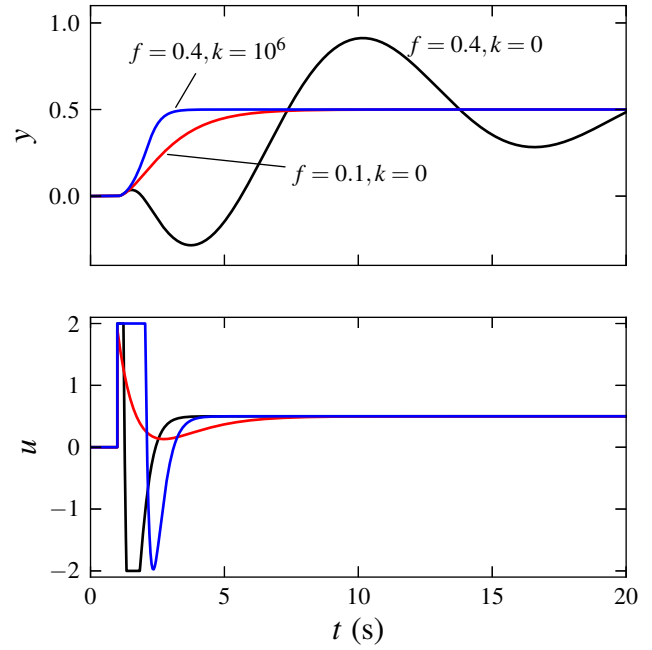


Figure 5: Plant input and output response to a reference step. Conventional feedforward ($k = 0$) with different filter cut-off frequencies f (Hz) compared to proposed anti-windup design ($k = 10^6$).

$$G(s) = \frac{1}{4s^2 + 4Ds + 1}$$

with

$$D = 0.2 \quad (4a)$$

and

$$D = 1.2 \quad (4b)$$

Plant models and the proposed feedforward control are implemented in Modelica using MSL blocks. The graphical representation of the total simulation model is shown in Figure 3. For both plants actuator limits are -2 and 2 . Gain k of the internal feedback loop is set to a comparable high number of 10^6 , which ensures that v virtually stays in between the actuator limits. According to the requirements in Section 2 the filter order is two.

4.2 Filter Tuning

The remaining degree of freedom for designing of the feedforward control is the filter's cut-off frequency f . In general higher values for f lead to a more aggressive feedforward control. This behaviour is identical

for the proposed anti-windup scheme as well as for the common scheme of Figure 1. In the latter case one would choose the filter frequency, so that the unconstrained plant input v would stay inside the actuator limits. There are two drawbacks with this approach. First, by not exploiting actuator limits we lose performance in terms of rise or settling time. Second, given a filter frequency f_1 works well for a reference step of 1, it is by no means guaranteed, that it does also work for higher step jumps. Instead of using a pure linear filter one would need to implement an application dependent algorithm for *Trajectory Planning*.

The proposed additional internal feedback (see Figure 2) solves both issues. Actuator limits are fully exploited using a general not application specific approach. In the following this is illustrated with comparative simulation results for both feedforward schemes with the slightly damped plant model (4a).

In Figure 4 plant input and output responses to reference steps using the conventional feedforward scheme ($k = 0$) are plotted. In both cases the filter frequency is 0.1 Hz. For a reference step of 0.5 (red) the input signal u stays inside the actuator limits. Whereas a reference step of 1.5 (black) leads temporarily to unconstrained input v above u_{\max} . The plant input u is chopped off at u_{\max} , which leads to a mismatch be-

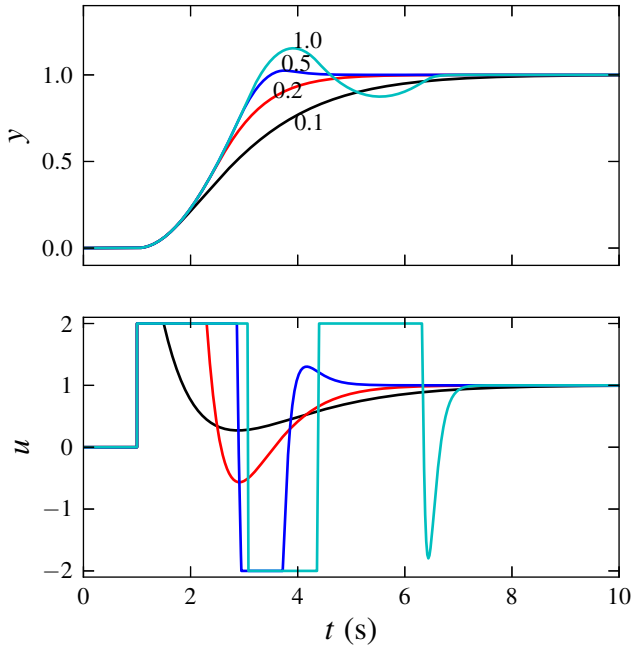


Figure 6: Response to reference unit step at $t = 1$ for different filter cut-off frequencies (in Hz). Slightly damped plant model with $D = 0.2$.

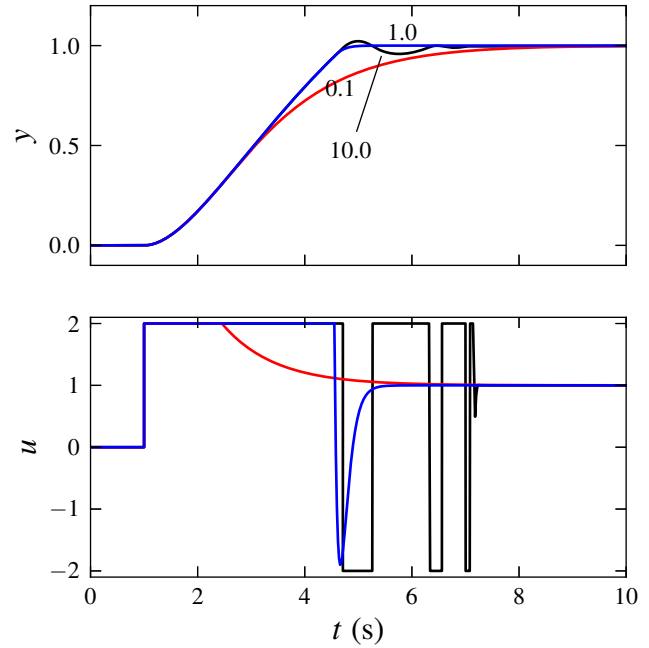


Figure 7: Response to reference unit step at $t = 1$ for different filter cut-off frequencies (in Hz). Well damped plant model with $D = 1.2$.

tween the states of the inverse plant model and the plant. This wind-up effect can be observed at the plant output y in form of unwanted oscillations (black).

Using internal high-gain feedback wind-up can be avoided. As a result we can use higher filter frequencies and fully exploit actuator limits. In Figure 5 simulation results for a step height of 0.5 are shown. The red curves are identical to Figure 4. If we increase the filter frequency to 0.4 Hz (black) we observe actuator saturation and oscillations of y . Things change dramatically if we turn on feedback with $k = 10^6$ (blue). Now, plant input u also hits its limit, but plant output y behaves perfectly smooth. Compared to feedforward without feedback (red) the settling time is decreased by 63%.

A natural question is, what happens if the filter frequency is further increased? An answer is given in Figure 6. Using identical plant and feedforward setup as in the simulations above, the filter frequency is changed. Increasing f from 0.1 to 0.2 Hz leads to a better feedforward performance. At 0.5 Hz we already see a small overshoot of y , and at 1.0 Hz there are notable oscillations in u and y .

Simulation experiments with the well-damped plant model (4b) give similar results, plotted in Figure 7. Here, we can choose higher filter frequencies. At $f = 1$ Hz there is still no overshoot of y . Further increase

of f finally also leads to oscillations in u and y .

To conclude these observations: Depending on the plant dynamics there seems to be a critical filter frequency f_{crit} . Values of f greater than this critical frequency lead to overshooting or even oscillating plant outputs.

As conclusion from additional simulation experiments, that are not shown in this article: f_{crit} strongly depends on the plant dynamics. Whereas dependencies to actuator limits and the reference signal (e.g. step height) seem to be very weak.

As consequence of these observations we formulate a tuning rule for filter frequency f :

- Simulate a step jump of the reference signal with the given plant model.
- Increase f until overshooting of the plant output is observed. This is f_{crit} .
- Choose f to be less than f_{crit} . Smaller values lead to increased robustness against model uncertainty.

This tuning rule is by no means theoretically founded. It is purely derived from simulation experiments.

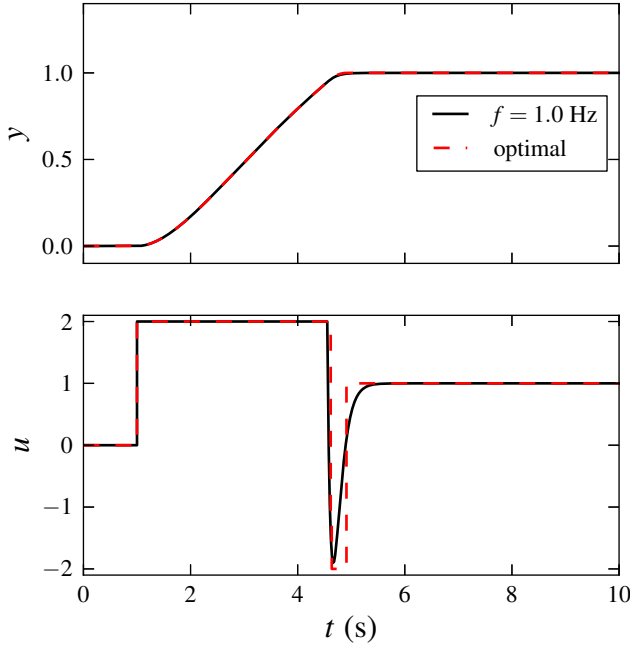


Figure 8: Time-optimal trajectories of plant input and output for rapid set-point changes compared to trajectories obtained with anti-windup feedforward. Well-damped plant model.

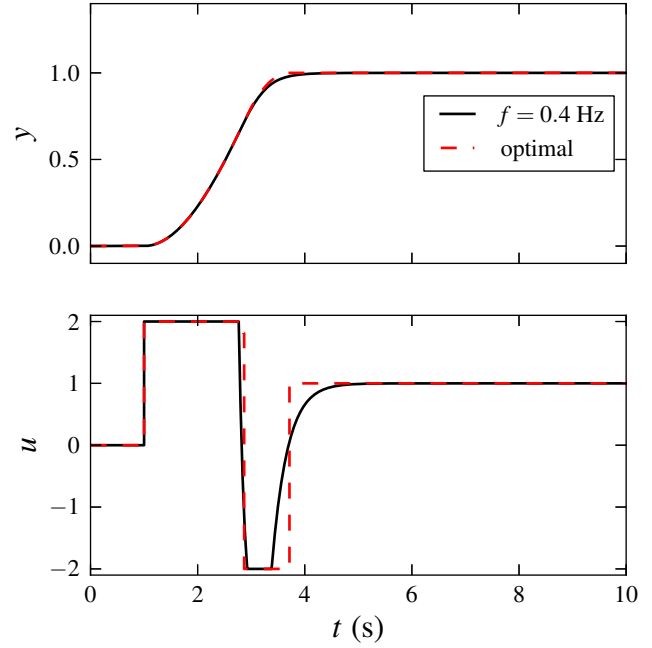


Figure 9: Time-optimal trajectories of plant input and output for rapid set-point changes compared to trajectories obtained with anti-windup feedforward. Slightly-damped plant model.

4.3 Comparison with Optimal Trajectories

In Section 4.2 several control trajectories are obtained with the proposed feedforward algorithm. Now we want to find out, how good they are. More precisely, we compare them with optimal trajectories, defined by the solution of the *Optimal Control Problem*

$$\min_{\mathbf{x}(\cdot), u(\cdot)} t_e \quad (5a)$$

$$\text{s.t.} \quad \frac{d\mathbf{x}}{dt}(t) = \mathbf{f}(\mathbf{x}(t), u(t)) \quad (5b)$$

$$-2 \leq u(t) \leq 2 \quad (5c)$$

$$\mathbf{x}(0) = \mathbf{0} \quad (5d)$$

$$\frac{d\mathbf{x}}{dt}(t_e) = \mathbf{0} \quad (5e)$$

$$y(t_e) = 1 \quad (5f)$$

\mathbf{x} denotes the vector of state variables and \mathbf{f} is the corresponding ODE function of plant model (4). Optimization task is to drive the plant from $\mathbf{x}(0) = \mathbf{0}$ as fast as possible to steady state with $y(t_e) = 1$.

OCP (5) is solved numerically using *Direct Multiple Shooting* within the software package MUSCOD-II [4, 6, 9]. The obtained optimal trajectories of u and

y are plotted for both plants in Figure 8 and 9 together with feedforward trajectories. Frequencies of feedforward filter are chosen according to simulation results from Section 4.2. One can see that there are differences of the feedforward control inputs u to its optimal trajectories. But at the output y of the well-damped plant Figure 8 there is practically no deviation between optimal and feedforward values. In Figure 9 deviations at the output of the slightly-damped plant can be detected, but they are comparable small. For both plant models the proposed feedforward control provides input trajectories for an almost time-optimal set-point change.

5 Real-World Experiments

In this section the proposed feedforward algorithm is tested on a real plant. To eliminate set-point tracking errors from model/plant mismatch feedforward control is enhanced with additional feedback. Measurement results are presented for pure feedforward and combined feedback/feedforward control.

5.1 Electric Water Heater

The experimental plant is a water heater as part of an existing test stand, where it is used to provide water flow at a given temperature. As sketched in Figure 10 a water mass flow \dot{m} with inlet temperature T_{in} is heated by electrical power P_{el} and leaves the heater with outlet temperature T_{out} . The electrical heating power is thyristor controlled with an analog input signal u . Control task is set-point tracking of plant output $y = T_{out}$ using input u . \dot{m} and T_{in} can be seen as disturbances.

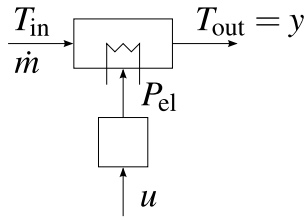


Figure 10: Schematic view of the controlled plant.

5.2 Plant Model

In Modelica a mixed physical/empirical model of the electric water heater is developed. The static behavior is modelled with the energy balance

$$0 = mc_p (T_{in} - T_{out}) + P_{el} \quad (6)$$

and a nonlinear empirical relation

$$P_{el} = f(x) \quad (7)$$

where x is an internal state. Equation (7) describes the nonlinear static behavior of the phase cutting thyristor controller.

The overall plant dynamics are modelled as second order time delay model. The identified transfer function from u to x is

$$G_{ux}(s) = \frac{1}{(1 + 5.804s)(1 + 5.809s)} e^{-1.885s} \quad (8)$$

In Figure 11 the measured step response used for identification together with the model step response is shown. Good agreement between fitted model and measured dynamics can be noticed.

5.3 Feedforward Measurement Results

Measurement results of the proposed feedforward control applied to the real plant are plotted in Figure 13.

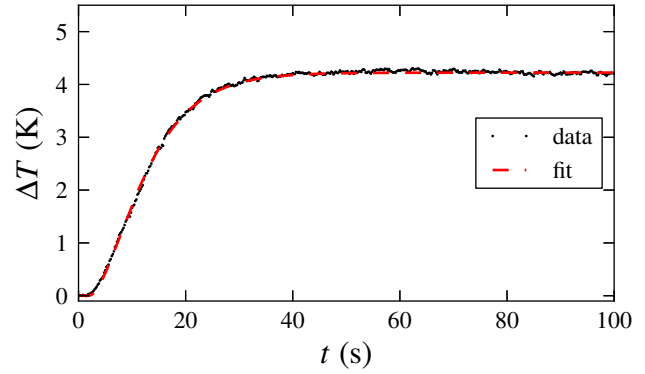


Figure 11: Measured input step response compared to the response of the identified second order time delay model.

The shown experiment spans set-point steps of varying height. Similar to the simulation experiments the feedforward reacts very aggressively to set-point changes and exploits actuator limits (0-10V). Due to model plant mismatch the output does not exactly track the given set-points. The static error is 1-2 K. Compared to the full control range of 27 K this error is relatively small.

If one would still like to drive this static error to zero, the feedforward control can be combined with measurement feedback, as it is done in the next section.

5.4 Combined Feedforward and Feedback Control

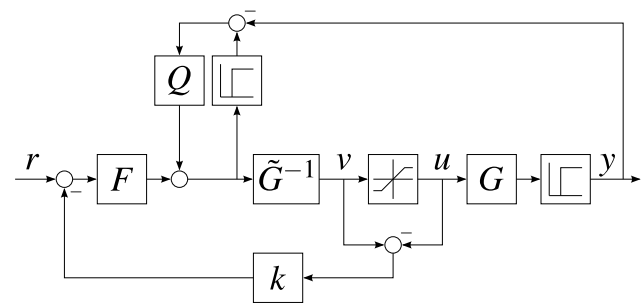


Figure 12: Feedforward and feedback control structure.

There are different possibilities, how to combine feedforward and feedback in one control system. See for example [11] for a brief overview. Following the approach suggested in [2] we add the output of an feedback filter Q to the input of the inverse plant model. The input of Q is the difference between measured and desired plant output. The resulting control

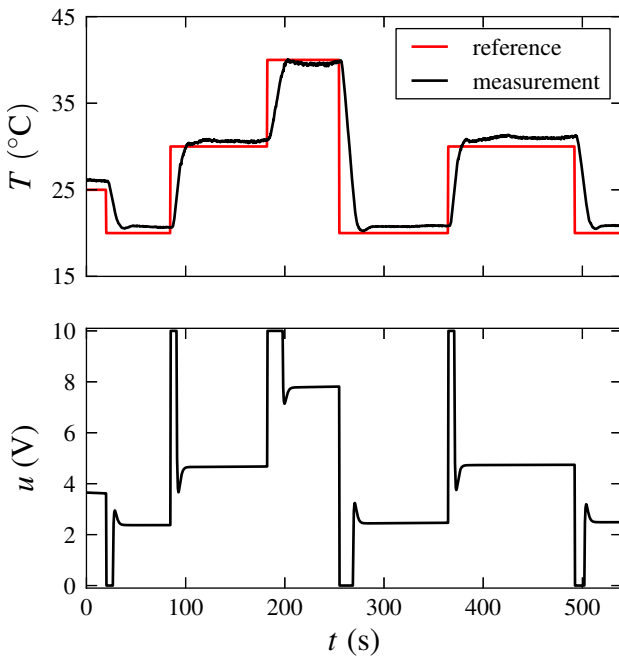


Figure 13: Measurement results from feedforward control of a liquid heater.

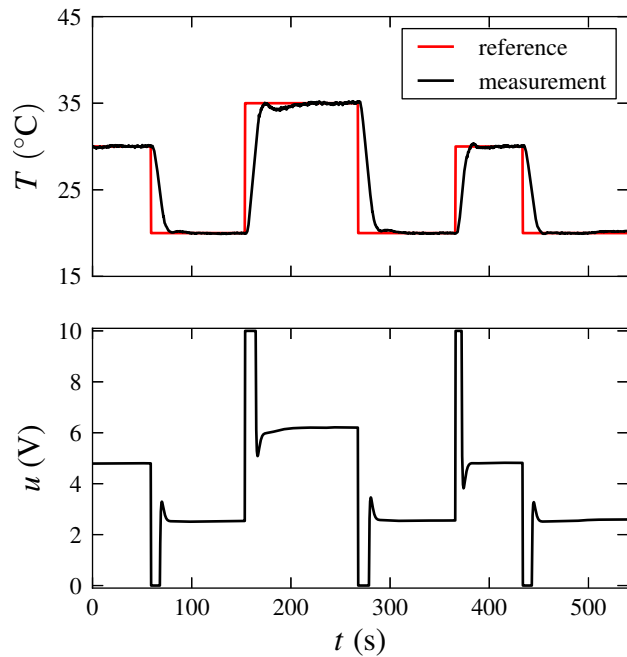


Figure 14: Measurement results from combined feed-forward and feedback control of a liquid heater.

structure is sketched in Figure 12. For $Q = 1$ infinity-gain feedback is obtained. Usually Q is implemented as low pass filter with unity gain. Filter order and frequency are tuning parameters of the feedback loop. Compared to [2] the control structure in Figure 12 has two additional elements. One is the internal feedback loop of the feedforward path as it is discussed in the previous sections. The second addition is a delay block. It is necessary because, plant model (8) includes delay, which is not invertible. This issue can be solved by splitting the model in a continuous block without delay and an additional delay block. Only the continuous block is inverted. The delay block is added afterwards to the desired plant output, before it is compared with the measured real output.

Measurement results obtained with this control structure are shown in Figure 14. As in Figure 13 the closed loop shows fast reaction to set-point changes. Additionally the static control error is eliminated.

6 Conclusion

The proposed anti-windup scheme for feedforward control with inverse models leads to near optimal control trajectories for a broad class of plant models: stable nonlinear SISO systems with stable zero dynamics. Because of its equation-based nature Modelica provides an user-friendly possibility to generate ap-

propriate feedforward control laws based on (forward) system models of the controlled plant. With one remaining tuning parameter – filter cut-off frequency – the proposed method is easy to configure. Experiments with a real plant show good performance of the anti-windup feedforward control. Control errors due to model plant mismatch are successfully avoided by extending feedforward control with feedback loop.

Acknowledgments

This work is part of the research project "Reflex Thermo" funded by the German Federal Ministry of Education and Research (reference EM1STR002).

References

- [1] Karl Johan Åström and Lars Rundqwist. Integrator windup and how to avoid it. In *Proc. of the American Control Conference*, 1989.
- [2] N. Bajcinca and T. Bünte. A novel control structure for dynamic inversion and tracking tasks. In *Proc. of the 16th IFAC World Congress*, Prague, 2005.
- [3] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss,

- M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf. The Functional Mockup Interface for Tool independent Exchange of Simulation Models. In *Proc of the 8th International Modelica Conference*, Dresden, 2011.
- [4] H. G. Bock and K. J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proc. of the 9th IFAC World Congress Budapest*, pages 243–247. Pergamon Press, 1984.
- [5] G.M. Clayton, S. Tien, a.J. Fleming, S.O.R. Moheimani, and S. Devasia. Inverse-feedforward of charge-controlled piezopositioners. *Mechatronics*, 18(5-6):273–281, June 2008.
- [6] Moritz Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. Phd thesis, Universität Heidelberg, 2001.
- [7] R. Hanus, M. Kinnaert, and J.-L. Henrotte. Conditioning technique, a general anti-windup and bumpless transfer method. *Automatica*, 23(6):729–739, 1987.
- [8] Stephen W John, Gursel Alici, and Christopher D Cook. Inversion-based feedforward control of polypyrrole trilayer bender actuators. *IEEE/ASME Transactions on Mechatronics*, 15(1):149–156, 2010.
- [9] D B Leineweber, I Bauer, A A S Schäfer, H G Bock, and J P Schlöder. An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-Scale Dynamic Process Optimization (Parts I and II). *Computers and Chemical Engineering*, 27:157–174, 2003.
- [10] Chung Seng Ling, Michael D Brown, Paul F Weston, and Clive Roberts. Gain Tuned Internal Model Control for Handling Saturation in Actuators. In *Proc. of the American Control Conference*, Boston, 2004.
- [11] Gertjan Looye, Michael Thümmel, Matthias Kurze, Martin Otter, and Johann Bals. Nonlinear Inverse Models for Control. In *Proc. of the 4th International Modelica Conference*, Hamburg, 2005.
- [12] Riccardo Marino. High-gain feedback in nonlinear control systems. *International Journal of Control*, 42(6):1369–1385, 1985.