A Toolbox for Input-Output System Inversion

Antonio Visioli, Aurelio Piazzi

Abstract: In this paper a Matlab-based toolbox for the input-output system inversion of linear systems is presented. Different methods, based either on analytical or numerical approaches, are implemented. The toolbox can be exploited in the design of a feedforward action for control systems in different contexts in order to improve performances in the set-point regulation. The use of a pre-actuation and a post-actuation time can be easily analyzed as well as the role played by the choice of the desired output function.

Keywords: CACSD, input-output inversion, feedforward, set-point regulation, optimization.

1 Introduction

It is well-known that a (properly designed) feedback controller provides robustness to a control system with respect to parameter variations and allows to compensate for external disturbances. On the other side, a high performance in the set-point following task can be achieved by adopting a suitable feedforward action. Indeed, the proper design of a control system consists of suitably combining feedback and feedforward control. Different techniques have been developed for the synthesis of a feedforward controller for a linear system (see e.g. [1, 2, 3]).

When the set-point regulation is of concern, a (noncausal) system inversion approach has been proven to be effective in this context [4]-[13]. Basically, the approach consists of selecting a desired output function in order to achieve a transition from a current output value y_0 to a new one y_1 and then to determine the corresponding input function by applying a stable inversion procedure. Then, the calculated input function is adopted as a reference command input to the (closed-loop) system, instead of the typical step signal.

Actually, while many software packages are available for the synthesis of feedback controllers (for example, via root locus techniques or Bode plots), they are not available for the synthesis of a system inversion based feedforward action. Indeed, the presence of a Computer Aided Control Systems Design tool makes the applicability of these (somewhat complex) techniques much easier and it can be exploited to understand deeply the role of the command function in the context of set-point regulation.

In this paper a Matlab-based toolbox for the input-output system inversion of a linear system is presented. Different techniques in this context are considered, related both to a noncausal and a causal approach. The toolbox allows to evaluate the role of the use of a pre-actuation and a post-actuation time as well as the role of the choice of the desired output function. It can be adopted as a useful tool in different fields, such as robust control, process control and control of mechatronic systems.

The paper is organized as follows. In Section 2 the input-output system inversion approach is briefly reviewed and the different methodologies implemented in the toolbox are presented. The functions implemented in the toolbox are described in Section 3 and application examples are shown in Section 4. Conclusions are drawn in Section 5.

Notation. C^i denotes the set of scalar real functions that are continuous till the ith derivative and BC^i denotes the subset of C^i of the scalar real functions that are bounded. The ith order differential operator is D^i.

2 Input-output system inversion

Consider a general asimptotically stable nonminimum-phase linear system Σ described by the following rational transfer function (note that this might represent the transfer function of a feedback control system):

$$H(s) = K_1 \frac{b(s)}{a(s)} = K_1 \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}, \quad K_1 \neq 0 \tag{1}$$

where it is assumed that polynomials $a(s)$ and $b(s)$ are coprime (no pole-zero cancellations occur) and that Σ has not purely imaginary zeros.

The input and output of Σ are $u \in \mathbb{R}$ and $y \in \mathbb{R}$ respectively and the relative order (or relative degree) of Σ is $\rho := n - m$. The set of all cause/effect pairs associated with Σ is denoted by

$$\mathcal{B} := \{(u(\cdot), y(\cdot)) \in P_c \times P_c : D^n y + a_{n-1}D^{n-1}y + \cdots + a_0 y = K_1(D^m u + b_{m-1}D^{m-1}u + \cdots + b_0 u)\} \tag{2}$$

where P_c denotes the set of piecewise continuous functions defined over $(-\infty, +\infty)$, i.e. the real field $\mathbb{R}$. In the framework of the behavioral approach, $\mathcal{B}$ is the behavior set of Σ that can be rigorously introduced by means of the so-called *weak solutions* of the differential equation associated to Σ [14].

The following proposition [14] is useful in the development of the subsequent analysis.

Proposition 1. *Consider any pair $(u(\cdot), y(\cdot)) \in \mathcal{B}$. Then, $u(\cdot) \in C^l(\mathbb{R})$ if and only if $y(\cdot) \in C^{\rho+l}(\mathbb{R})$ with l being a nonnegative integer.*

The considered regulation problem consists of obtaining an output transition from a previous value y_0 to a new value y_1. Without loss of generality, in the following we will consider $y_0 = 0$. Define $y_d(\cdot) \in BC^k$ with $y_d(t) = 0$ for $t < 0$ as the desired output function to obtain the transition. From a practical point of view, a transition time τ has to be defined, i.e. the desired output function is defined as

$$y_d(t) := \begin{cases} 0 & \text{for} \quad t < 0 \\ y_{01}(t) & \text{for} \quad 0 \leq t \leq \tau \\ y_1 & \text{for} \quad t > \tau. \end{cases} \tag{3}$$

Then, the following stable input-output inversion (SIOI) problem can be formulated.
SIOI problem. Determine an input function $u_d(\cdot) \in BC^{k-\rho}$ such that

$$(u_d(\cdot), y_d(\cdot)) \in \mathcal{B}. \tag{4}$$

The general solution to the SIOI problem can be derived as follows [15]. First, express the inverse of the transfer function (1) as

$$H^{-1}(s) = \frac{1}{K_1} \frac{a(s)}{b(s)} = \xi_\rho s^\rho + \xi_{\rho-1} s^{\rho-1} + \cdots + \xi_0 + H_0(s) \tag{5}$$

where $H_0(s)$ is a strictly proper rational transfer function representing the zero dynamics of Σ. By using the fraction expansion, $H_0(s)$ can be decomposed as

$$H_0(s) = H_0^-(s) + H_0^+(s) = \frac{d(s)}{b^-(s)} + \frac{e(s)}{b^+(s)} \tag{6}$$

where $b^-(s)$ and $b^+(s)$ are the monic polynomials containing the roots of $b(s)$ with negative and positive real parts, respectively. Define $\eta_0^-(t)$ and $\eta_0^+(t)$ as the analytic extensions of $\mathcal{L}^{-1}[H_0^-(s)]$ and $\mathcal{L}^{-1}[H_0^+(s)]$ over the space of the Bohl functions for which $\eta_0^-(t)1(t) = \mathcal{L}^{-1}[H_0^-(s)]$ and $\eta_0^+(t)1(t) =$

$\mathcal{L}^{-1}[H_0^+(s)]$ respectively.

Then, the solution of the SIOI problem is derived as:

$$u_d(t) = \xi_\rho D^\rho y_d(t) + \cdots + \xi_1 D y_d(t) + \xi_0 y_d(t) + \int_0^t \eta^-(t-v)y_d(v)dv - \int_t^{+\infty} \eta^+(t-v)y_d(v)dv. \quad (7)$$

It is worth noting that, in general, $u_d(t)$ is defined over the time interval $(-\infty, +\infty)$ and therefore, in order to practically use it, it is necessary to truncate it. Thus, the input function exhibits a pre-actuation (associated with the unstable zeros) and a post-actuation (associated with the stable zeros) time intervals (see for example [16]), denoted as t_p and t_f respectively. They can be calculated with arbitrary precision by selecting two arbitrary small parameters ε_0 and ε_1 and by subsequently determining

$$t_0 := \max\{t' \in \mathbb{R} : |u_d(t)| \le \varepsilon_0 \ \forall t \in (-\infty, t']\} \quad (8)$$

and

$$t_1 := \min\left\{t' \in \mathbb{R} : \left|u_d(t) - \frac{1}{H(0)}\right| \le \varepsilon_1 \ \forall t \in [t', \infty)\right\}. \quad (9)$$

Then, it has to be fixed

$$t_p = \min\{0, t_0\} \qquad t_f = \max\{\tau, t_1\}. \quad (10)$$

Hence, the actual input function to be applied to the system is given by

$$\tilde{u}_d(t) := \begin{cases} 0 & \text{for} \quad t < t_p \\ u_d(t) & \text{for} \quad t_p \le t \le t_f \\ \frac{1}{y_1} & \text{for} \quad t > t_f. \end{cases} \quad (11)$$

Alternatively, the pre-actuation and post-actuation time intervals can be calculated as [4]

$$t_p = -\frac{10}{D_{rhp}} \qquad t_f = \frac{10}{D_{lhp}} \quad (12)$$

where D_{rhp} and D_{lhp} are the minimum distance of the right and left half plane zeros respectively from the imaginary axis of the complex plane.

It is worth noting that the pre-actuation time is zero when there are no unstable zeros and the post-actuation time is zero when there are no stable zeros.

In general, the integrals in expression (7) can not be solved analytically and therefore a numerical solution has to be determined (in this context the Cavalieri-Simpson's rule can be conveniently exploited to found an accurate solution with a small computational time [15]). A notable exception occurs when the selected desired output function is a polynomial function [17], i.e.

$$y_d(t; \tau) = \begin{cases} 0 & \text{for} \quad t < 0 \\ y_1 \frac{(2k+1)!}{k!\tau^{2k+1}} \sum_{i=0}^k \frac{(-1)^{k-i}}{i!(k-i)!(2k-i+1)} \tau^i t^{2k-i+1} & \text{for} \quad 0 \le t \le \tau \\ y_1 & \text{for} \quad t > \tau \end{cases} \quad (13)$$

Note that function $y_d(t; \tau)$, parameterized by the transition time τ is a C^k-function over $(-\infty, +\infty)$ and is strictly increasing in the interval $[0, \tau]$ so that neither overshooting nor undershooting appear in this output planning for set-point regulation. In this case an analytical solution can be found [18] and this fact can be exploited in speeding up the computational time and most of all in avoiding numerical problems. A very interesting application of the analytical stable inversion procedure is for the improvement of the set-point following performance of Proportional-Integral-Derivative (PID) controllers. Specifically, if a PID controller is employed for a first-order plus dead-time (FOPDT) process, described by the following transfer function,

$$P(s) = \frac{K}{Ts+1}e^{-Ls}, \quad (14)$$

or for an integrator plus dead-time (IPDT) process, described by the transfer function

$$P(s) = \frac{K}{s} e^{-Ls}, \qquad (15)$$

then a closed-form solution of the stable input-output inversion applied to the closed-loop system can be exploited (a rational closed-loop transfer function is obtained by adopting a Padè approximation) [13]. Indeed, the actual command signal to be applied for a given process and a given PID controller is determined by substituting the actual value of the process and PID parameters into the resulting closed-form expression. In this context the PID transfer function is expressed as

$$C(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s\right) \frac{1}{T_f s + 1}, \qquad (16)$$

where K_p is the proportional gain, T_i is the integral time constant, T_d is the derivative time constant and T_f is the time constant of the filter that is adopted to render the system proper.

A polynomial output function can be also usefully exploited in determining a causal input-output inversion despite the presence of unstable zeros [9]. In particular, the order of the polynomial function is selected in order to satisfy boundary conditions so that $y_d(\cdot) \in BC^k$ with $k \geq \rho$ and at the same time to have a number of free coefficients equal to the number of the unstable zeros of the system Σ. Then, the free parameters are determined in order to annihilate the unstable modes in the input function determined by the inversion procedure. In this way there is no need of a pre-actuation time interval and the resulting inversion is causal. It can be therefore employed when a preview time is not available in a given application. However, this is paid by the possible presence of undershoots and overshoots in the resulting output function. Note that the approach can be easily extended in order to avoid also the presence of a post-actuation time interval.

3   Toolbox description

The designed Matlab-based toolbox implements the methods described in Section 2. It requires the Control System Toolbox and the Symbolic Math Toolbox of Matlab.
The following main functions are available.

```
[time,command,preaction,postaction]=numdyninvcs(sys,yd,tau,st)
```
This function determines the input command function of a system that causes a desired output function by means of an input-output inversion scheme based on the use of the Cavalieri-Simpson's rule for the determination of the integrals in (7) [15]. In particular, the meaning of the parameters is the following one:

- `sys` is the transfer function of the system expressed in symbolic form (with `s` as a symbolic variable);

- `yd` is the desired output function (for $t \in [0, \tau]$) expressed as an array of numerical values from 0 to τ corresponding to the time instant equally spaced by the sampling time;

- `tau` is the transition time;

- `st` is the sampling time;

- `time` is the output time vector; it starts from the preaction time t_p but for convenience the zero time is shifted to t_p;

- `command` is the determined input function expressed as a numerical array corresponding to the time array `time`;

- `preaction` is the pre-actuation time calculated with formula (12);

- `postaction` is the post-actuation time calculated with formula (12).

`[time,command,preaction,postaction]=outdyninvcs(sys,yd,tau,st)`
This function operates basically as `numdyninvcs` with the difference that the desired output function `yd` is expressed as a symbolic expression with symbolic variable `t`.

`[time,command,preaction,postaction]=numdyninv(sys,yd,tau,st)`
This function is very similar to `numdyninvcs` but it performs the numerical integration by applying a rectangular rule. In order to obtain an accurate result, a small value of the sampling time has to be selected. This might result in a high computational time.

`[time,command,preaction,postaction]=outdyninv(sys,yd,tau,st)`
This function is very similar to `outdyninvcs` but it performs the numerical integration by applying a rectangular rule. Also in this case, in order to obtain an accurate result, a small value of the sampling time has to be selected. This might result in a high computational time.

`[time,command,preaction,postaction]=dyninv(sys,y1,tau,threshold0,`
`threshold1,st)`
This function solves the input-output inversion problem when the desired output function is a polynomial function (13). The order of the polynomial is automatically selected, according to Proposition 1, in order to obtain a continuous input function, i.e. such as $u_d(\cdot) \in BC^0$. Since the input function is determined analytically, the pre-actuation and post-actuation time intervals are conveniently determined by adopting formulae (10). The function parameters that are different from those that have been already described have the following meaning:

- `y1` is the desired new output steady-state value (it is assumed, without loss of generality, that the current input and output steady-state values are zero);

- `threshold0` is the parameter ε_0 in formula (8), which is adopted to calculate the pre-actuation time;

- `threshold1` is the parameter ε_1 in formula (9), which is adopted to calculate the post-actuation time.

`[time,command,taunum,preaction,postaction]=`
`                optdyninv(sys,y1,constraints,threshold0,threshold1,tc)`

This function solves the minimum-time inversion problem that consists of finding the minimum output transition time subject to constraints posed on the input function and its derivatives until an arbitrary order l. Formally, the optimisation problem is posed as follows [18]:

$$\min_{\tau \in \mathbb{R}^+} \tau \tag{17}$$

such that, $i = 0, 1, \ldots, l$,

$$|D^i u_d(t; \tau)| \leq u_M^{(i)} \quad \forall t \in (-\infty, +\infty) \tag{18}$$

where the positive values $u_M^{(i)}$, $i = 0, 1, \ldots, l$, are given bounds of the problem. Note that the problem admits a solution if $u_M^{(0)} > 1/|H(0)|$ and $u_M^{(i)} > 0$, $i = 1, \ldots, l$. The optimisation problem is solved by applying a simple bisection algorithm in conjunction with a gridding of the time axis [17]. With respect to the function `dyninv` there are the following different parameters:

- `constraints` is the array (of $l + 1$ elements) of the constraints for the input derivatives until the lth order; note that the user-chosen dimension of the array automatically determines the order of the constrained derivatives and therefore the order of the output polynomial function (which is determined as $l - 1 + \rho$ so that $u_d(\cdot) \in BC^{l-1}$);

- `taunum` is the resulting optimal transition time.

It is worth noting that, if a rigorous determination of the transition time is sought, the posed optimisation problem should be approached with the tools of global optimisation. In this context the presented input-output inversion toolbox can be easily integrated with the `b4m` toolbox that allows to handle interval arithmetic, which is a well-known effective tool for global optimisation [19].

`[time,command,out,postaction]=causaldyninv(sys,y1,tau,st,hbc,pa)`
This function implements the causal approach proposed in [9]. The resulting pre-actuation time is always zero despite the possible presence of unstable zeros. In particular, the function deals with the following parameters:

- `hbc` is the order h of the boundary conditions to be satisfied for the polynomial output function at time $t = 0$ and $t = \tau$, so that $y_d(\cdot) \in BC^{2h+1}$. Note that it has to be $h \geq \rho$ in order to ensure that the input function is at least continuous, i.e. $u_d(\cdot) \in BC^0$;

- `pa` is a string that, if set to 'nopostaction', avoid also the use of a post-actuation time even if the system has stable zeros. In other words, in this case the system attains an equilibrium point at $t = \tau$. If the parameter is not adopted or if it is set to another value, then a post-actuation time is present and it is determined by means of formula (12).

`[time,command,preaction,ˉpostaction]=piddyninvFOPDT(K,T,L,Kp,Ti,Td,Tf,tau,st)`
This function determines the input command function to unitary feedback closed-loop system in which a process described by a FOPDT transfer function is controlled by a PID controller. The following parameters are adopted:

- `K`, `T`, `L` are the process gain, time constant and dead time respectively (see (14));

- `Kp`, `Ti`, `Td`, `Tf` are the PID parameters (see (16), where the meaning of the different parameters is obvious).

`[time,command,preaction,postaction]=piddyninvIPDT(K,T,L,Kp,Ti,Td,Tf,tau,st)`
This function determines the input command function to unitary feedback closed-loop system in which a process described by a IPDT transfer function is controlled by a PID controller. The meaning of the parameters can be straightforwardly derived by considering those of the previous function together with expression (15).

4   Application examples

In order to better understand the usage of the input-output inversion toolbox, a few examples are given. Consider the system

$$H(s) = 4\frac{(1-s)(s+1)}{(s+2)(s^2+2s+2)}. \tag{19}$$

To insert the system in symbolic form in the Matlab workspace, the following two commands can be applied:

```
syms s
H=4*(1-s)*(s+1)/(s+2)/(s^2+2*s+2)
```

Then, suppose that an output transition from 0 to $y_1 = 1$ is required to be performed in $\tau = 3$ by means of the following output function:

$$y(t) = \frac{4796646617206209}{562949953421312}t^{\frac{63}{25}} - \frac{4658008624178539}{562949953421312}t^{\frac{127}{50}} + \tag{20}$$
$$\frac{6313836048447483}{531691198313966349161522 8241121378304}t^{39}.$$

The output function can be inserted in the Matlab workspace (denote the variable as `yt`) either in symbolic form or as an array whose elements are the values of $y(t)$ for $t = 0, T_s, 2T_s, \ldots, \tau$, where T_s is the sampling time (in this case it has been selected $T_s = 0.001$). In the first case the command to be adopted is:

```
[time,command,preaction,postaction]=outdyninvcs(H,yt,3,0.001)
```

while in the second case it has to be

```
[time,command,preaction,postaction]=numdyninvcs(H,yt,3,0.001)
```

In both cases the pre-actuation time results to be $t_p = -10$ s and the postaction time is $t_f = 10$ s (according to expressions (12)). The resulting input function and the output function obtained by applying it to the actual system are reported in Figure 1 (note that the result is the same in both cases).

Select now a polynomial output function (13) to perform again an output transition from 0 to $y_1 = 1$ and select the parameters $\varepsilon_0 = \varepsilon_1 = 10^{-3}$. In this case the Matlab command to be adopted is

```
[time,command,preaction,postaction]=dyninv(H,1,3,0.001,0.001,0.001)
```

The resulting pre-actuation and post-actuation times `preaction` and `postaction` (determined by means of formula (10)) are respectively $t_p = -6.256$ s and $t_f = 3.547$ s. The determined input and the corresponding system output are plotted in Figure 2. Note that the resulting output function is a cubic polynomial, i.e.

$$y_d(t;\tau) = y_1\left(-\frac{2}{\tau^3}t^3 + \frac{3}{\tau^2}t^2\right) \qquad t \in [0,\tau]$$

as it is $k = \rho = 1$ in formula (13).

Consider now the minimum time problem (17)-(18) and set the limits on the input derivatives as $u_M^{(0)} = 2$, $u_M^{(1)} = 10$ and $u_M^{(2)} = 20$. This means that the following Matlab command has to be executed:

```
limits=[2 10 20]
```

Then, the following function has to be employed (note that the sampling time is 0.001 s as before and again it is $\varepsilon_0 = \varepsilon_1 = 10^{-3}$):

```
[time,command,taunum,pre,post]=optdyninv(H,1,limits,0.001,0.001,0.001)
```

The resulting optimal transition time `taunum` is equal to 0.875 s, while the pre-actuation and post-actuation times are $t_p = -7.111$ s and $t_f = 4.402$ s. The determined command input and the corresponding system output are shown in Figure 3, while the first and second derivatives of the command input are plotted in Figure 4. It can be deduced that the active constraint is the one related to the second time derivative of the input.

If the causal approach is desired, i.e. the pre-actuation time is to be avoided, then the function

`causaldyninv` has to be adopted. In particular, we select the order of the boundary conditions as $h = \rho = 1$ and the desired transition time as $\tau = 3$. Then, if a post-actuation time interval is allowed, the Matlab command to be adopted is:

`[time,command,out,postaction]=causaldyninv(H,1,3,0.001,1)`

In this case the resulting command input and the corresponding output are plotted in Figure 5. Note that the post-actuation time interval is $t_f = 10$ s. Conversely, if a post-actuation is not allowed, the Matlab function to be employed is

`[time,command,out,postaction]=causaldyninv(H,1,3,0.001,1,'nopostaction')`

The resulting command input and the corresponding output function are shown in Figure 6. It can be seen that in both cases the output function is not monotonic. Indeed, the possible overshoots and undershoots are more and more significant when the selected transition time decreases.

In order to verify the effectiveness of the inversion approach for PID control, consider first the system

$$P(s) = \frac{2}{10s+1}e^{-5s}, \tag{21}$$

controlled by a PID controller (see (16)) with $K_p = 1.2$, $T_i = 10$, $T_d = 2.5$, $T_f = 0.01$. The transition time is fixed to 15 s and the sampling time to 0.01 s. Then, the following Matlab command is adopted:

`[time,command,pre,post]=piddyninvFOPDT(2,10,5,1.2,10,2.5,0.01,15,0.01)`

The resulting command input (applied to the closed-loop system) and output functions are plotted in Figure 7. The pre-actuation and post-actuation time intervals are $t_p = -16.67$ s and $t_f = 50.01$ s. Note that the output function is slightly different from the desired one because of the Padè approximation.

Then, a IPDT transfer function is considered:

$$P(s) = \frac{0.1}{s}e^{-5s}. \tag{22}$$

In this case the tuning of the PID parameters is $K_p = 0.12$, $T_i = 10$, $T_d = 2.5$, $T_f = 0.01$. The transition time and the sampling time are the same as before, so that the following function is employed:

`[time,command,pre,post]=piddyninvIPDT(0.1,10,5,0.12,10,2.5,0.01,15,0.01)`

Figure 8 reports the determined closed-loop command input and the resulting system output. Also in this case the pre-actuation and post-actuation time intervals are $t_p = -16.67$ s and $t_f = 50.01$ s.

5 Conclusions

A Matlab-based toolbox for the input-output inversion of linear scalar systems has been described in this paper. The toolbox is very useful to understand and to verify the effectiveness of a feedforward action in a control scheme and, in general, of a combined feedback/feedforward synthesis. In this context, the role played by the planned output function can be easily evaluated as well as the influence of the noncausal approach with respect to a causal one. Indeed, all the aspects related to the use of an input-output inversion-based control design can be thoroughly investigated and analyzed.

The toolbox can be exploited in different fields such as motion control, robust control, and process control.

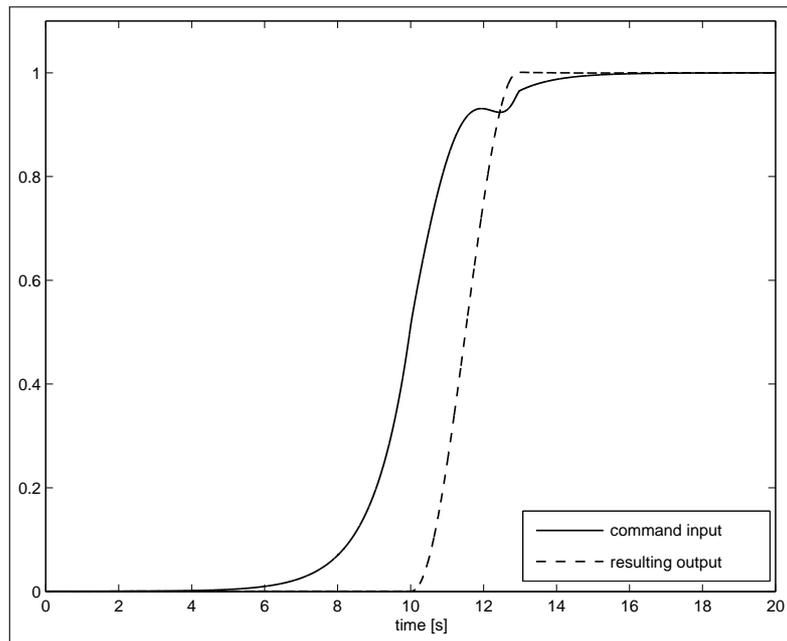It can be downloaded from the website http://www.ing.unibs.it/∼visioli/ioitoolbox.htm

Figure 1: Command input and resulting system output with system (19) and desired output function (20)
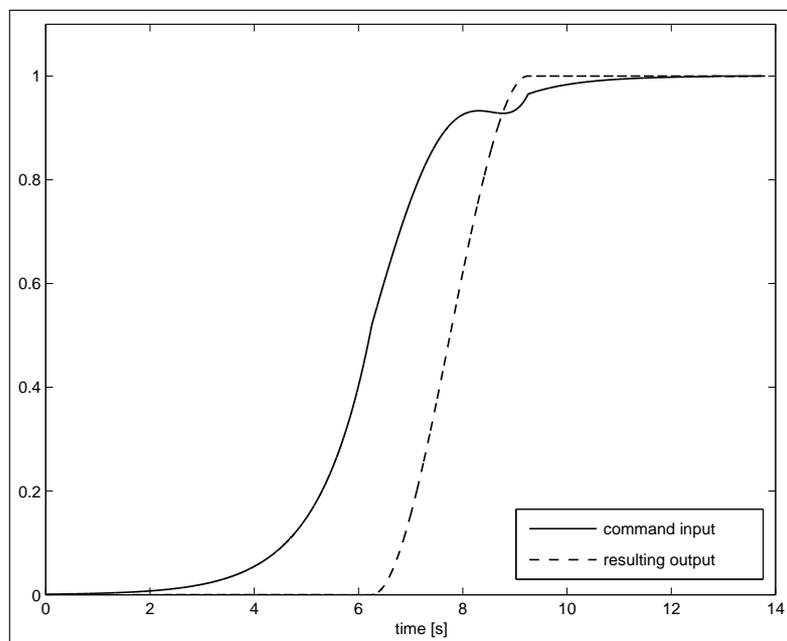

Figure 2: Command input and resulting system output with system (19) and a polynomial desired output function (13)
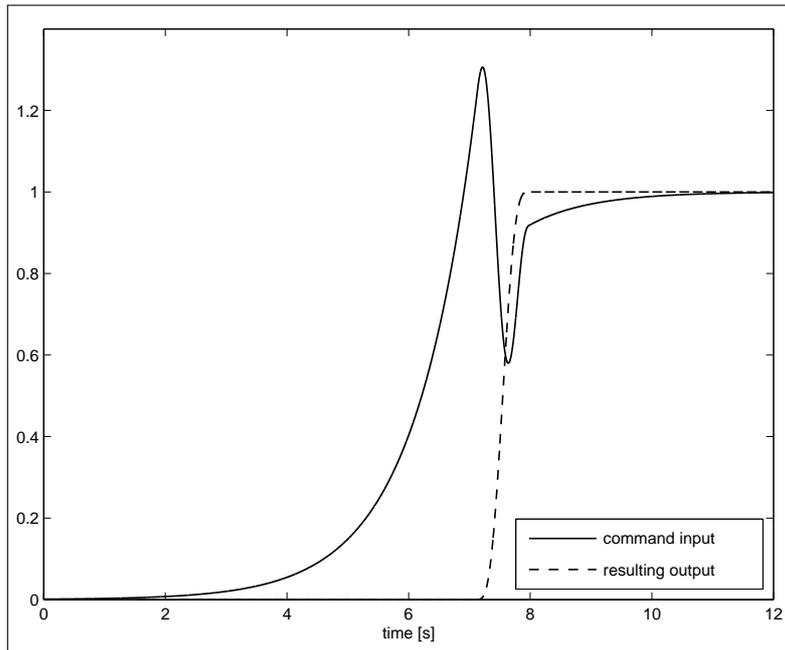
Figure 3: Optimal command input and resulting minimum-time system output with system (19) and a polynomial desired output function (13)
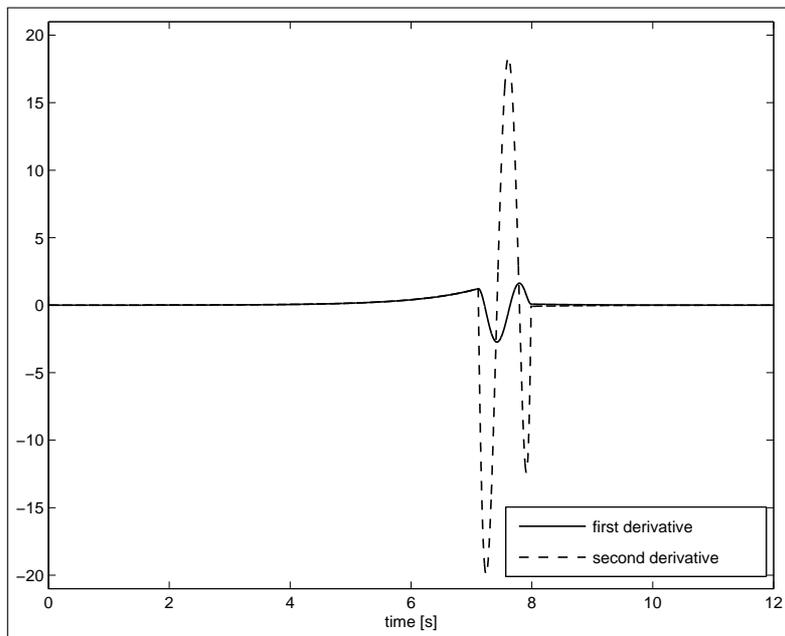

Figure 4: First and second derivative of the optimal command input with system (19) and a polynomial desired output function (13)
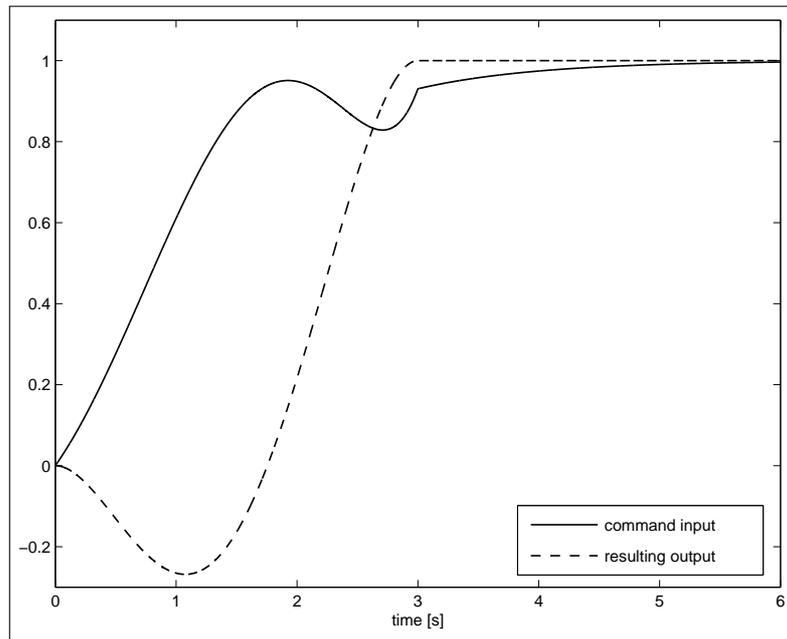
Figure 5: Command input and resulting system output with system (19) and a causal approach with post-actuation
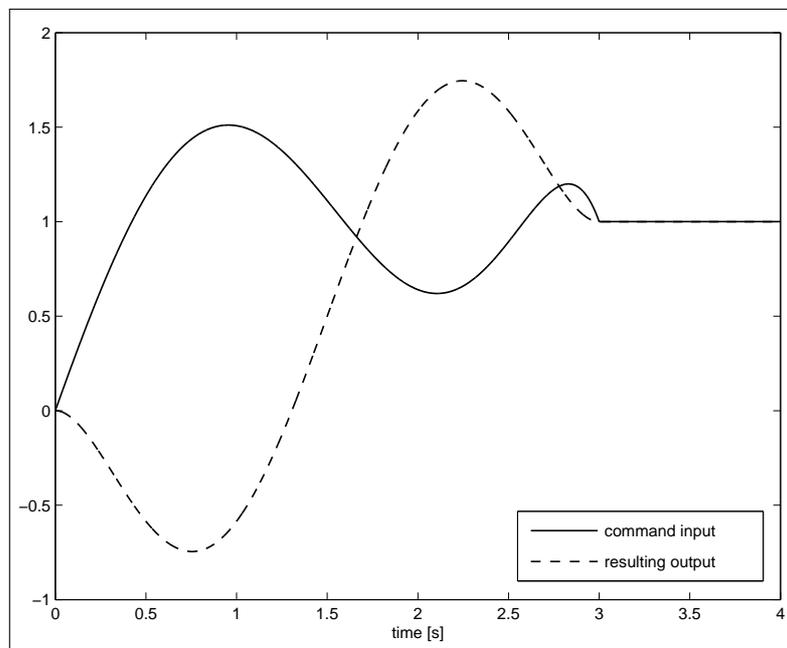

Figure 6: Command input and resulting system output with system (19) and a causal approach without post-actuation
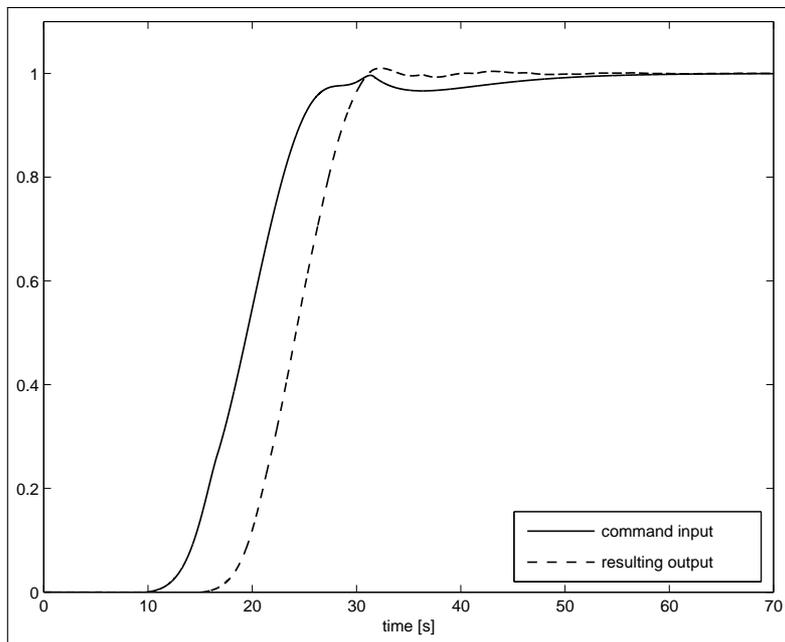
Figure 7: Command input and resulting system output with system (21) with a PID controller
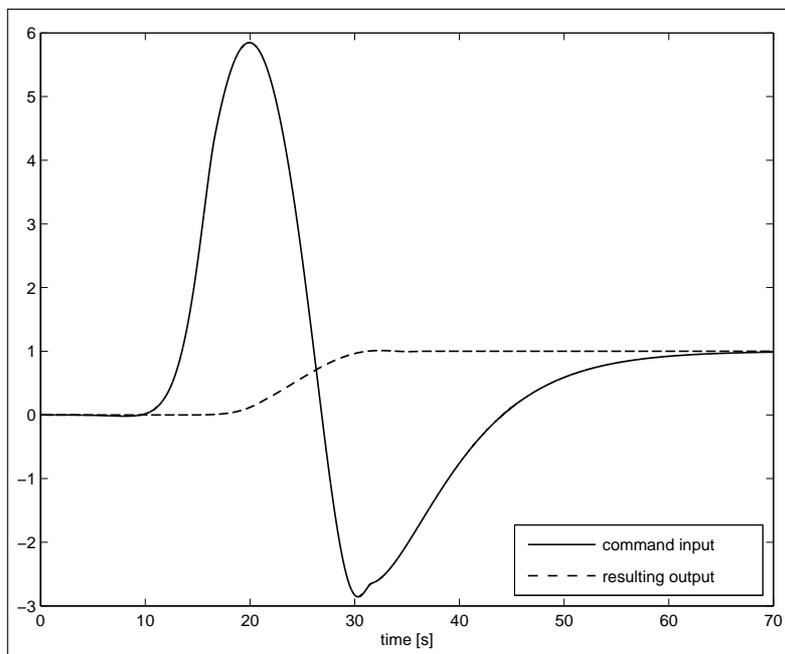

Figure 8: Command input and resulting system output with system (22) with a PID controller

Bibliography

[1] B. C. Kuo, *Automatic Control Systems*, Prentice Hall, Englewood Cliffs (NJ), 1995.

[2] A. Wallen, K. J. Åström, Pulse-step control, *Preprints of the 15th IFAC World Congress on Automatic Control*, Barcelona (Spain), 2002.

[3] A. Visioli, A new design for a PID plus feedforward controller, *Journal of Process Control*, Vol. 14, No. 4, pp. 455-461, 2004.

[4] H. Perez, S. Devasia, Optimal output transitions for linear systems, *Automatica*, Vol. 39, pp. 181-192, 2003.

[5] Q. Zou, S. Devasia, Preview-based optimal inversion for output tracking: application to scanning tunneling microscopy, *Proceedings IEEE International Conference on Decision and Control*, Las Vegas (USA), pp. 79-85, 2002.

[6] D. Iamratanakul, H. Perez, S. Devasia, Feedforward trajectory design for output transitions in discrete-time systems: disk-drive example, *Proceedings of the American Control Conference*, Denver (USA), pp. 3142-3147, 2003.

[7] A. Piazzi, A. Visioli, Minimum-time system-inversion-based motion planning for residual vibration reduction, *IEEE/ASME Transactions on Mechatronics*, Vol. 5, No. 1, pp. 12-22, 2000.

[8] A. Piazzi, A.Visioli, Optimal inversion-based control fo the set-point regulation of nonminimum-phase uncertain scalar systems, *IEEE Transactions on Automatic Control*, Vol. 46, No. 10, pp. 1654-1659, 2001.

[9] M. Benosman, G. Le Vey, Stable inversion of SISO nonminimum phase linear systems through output planning: an experimental application to the one-link flexible manipulator, *IEEE Transactions on Control Systems Technology*, Vol. 11, No. 4, pp. 588-597, 2003.

[10] C. Guarino Lo Bianco, A. Piazzi, A servo control system design using dynamic inversion, *Control Engineering Practice*, Vol. 10, No, 8, pp. 847-855, 2002.

[11] A. Piazzi, A. Visioli, Optimal dynamic inversion based control of an overhead crane, *IEE Proceedings - Control Theory and Applications*, Vol. 149, No. 5, pp. 405-411, 2002.

[12] A. Visioli, A. Piazzi, Improving set-point following performance of industrial controllers with a fast dynamic inversion algorithm, *Industrial Engineering and Chemistry Research*, Vol. 42, pp. 1357-1362, 2003.

[13] A. Piazzi, A. Visioli, A noncausal approach for PID control, *Journal of Process Control*, Vol. 16, pp. 831-843, 2006.

[14] J. W. Polderman, J. C. Willems, *Introduction to Mathematical Systems Theory*, Springer-Verlag, New York, 1998.

[15] D. Pallastrelli, A. Piazzi, Stable dynamic inversion of nonminimum-phase scalar linear systems, *Preprints of the 16th IFAC World Congress on Automatic Control*, Prague (CZ), 2005.

[16] Q. Zou, S. Devasia, Preview-based inversion of nonlinear nonminimum-phase systems: VTOL example, *Proceedings of the IEEE International Conference on Decision and Control*, Paradise Island (The Bahamas), pp. 4350-4356, 2004.

[17] A. Piazzi, A. Visioli, Optimal noncausal set-point regulation of scalar systems, *Automatica*, Vol. 37, No. 1, pp. 121-127, 2001.

[18] A. Piazzi, A. Visioli, Using stable input-output inversion for minimum-time feedforward constrained regulation of scalar systems, *Automatica*, Vol. 41, No. 2, pp. 305-313, 2005.

[19] E. Hansen, G. W. Walster, *Global optimization using interval analysis* - 2nd edition, Marcel Dekker, 2003.

Antonio Visioli
University of Brescia
Dipartimento di Elettronica per l'Automazione
Via Branze 38, I-25123 Brescia, Italy
E-mail: antonio.visioli@ing.unibs.it

Aurelio Piazzi
University of Parma
Dipartimento di Ingegneria dell'Informazione
Parco Area delle Scienze 181A, I-43100 Parma, Italy

Antonio Visioli received the Laurea degree in electronic engineering from the University of Parma, Parma, Italy, and the Ph.D. degree in applied mechanics from the University of Brescia, Brescia, Italy, in 1995 and 1999 respectively. His Ph.D. dissertation was on control strategies for industrial robot manipulators. He is currently an Associate Professor of Automatic Control, Department of Electronics for Automation, University of Brescia, Italy. His research interests include industrial robot control and trajectory planning, dynamic-inversion-based control and process control. He is the author or co-author of more than 100 papers in international journals and refereed conference proceedings and he is the author of the book *Practical PID Control* published by Springer. Dr. Visioli is a senior member of IEEE and a member of IFAC and Anipla (Italian Association for the Automation).

Aurelio Piazzi received the Laurea degree in nuclear engineering in 1982 and the Ph.D. degree in system engineering in 1987, both from the University of Bologna, Italy. From 1990 to 1992, he was Research Associate in System Theory, DEIS, University of Bologna. Since 1992 he has been at the University of Parma where he is currently Full Professor of Automatic Control at the Department of Informatics Engineering. His main research interests are in system and control theory and related engineering applications. His recent research activities have focused on optimization and dynamic inversion techniques for autonomous vehicle guidance and for the design of high-performance control systems. Scientific coordinator of bilateral research programs in collaboration with various industries (among them CNR, ENEL, RFI -Ferrovie dello Stato), in 2002 and 2003 he has directed the European Project COOKIES within the EU cluster Eutist-IMV (Integrated Machine Vision) in collaboration with Gruppo Colussi (Perugia) for the artificial vision-based control of food industrial ovens. He is a member of IEEE and SIAM and has published over 90 scientific papers in international journals and conference proceedings.