1. Introduction

It is well-known that the ease of implementation is a major issue for an industrial controller, as it is important to achieve the best cost/benefit ratio. In fact, it is for this reason that proportional–integral–derivative (PID) controllers are still largely the most used controllers in industry, despite the fact that many effective control methodologies have been proposed in the past 50 years.

Thus, for a new methodology to be suitable for actual implementation in an industrial environment, it has to satisfy different requirements, in addition to achieving high levels of performance. Basically, the extra tuning effort required from the operator should be kept as low as possible, the design procedure should have a clear physical meaning, and the computational effort should be minimized to allow for the low-cost implementation of the controller.

In this context, we propose a new technique for the improvement of the set-point-following performance of industrial controllers, which consists basically of using a determined command input function instead of a standard step signal. In particular, we consider a single-loop control scheme in which the controller (typically of PID type, but no assumption has to be made regarding its structure) has already been fixed. A step signal is then applied to the set point, and the closed-loop system model is identified. Note that the use of a step-response-based model is widely adopted in the model predictive control framework, particularly in the commercially available dynamic matrix control algorithm. At this point, a desired system output function has to be selected when a transition from one set-point value to another must be performed by the system.
can then be obtained by considering the truncated response (t ∈ {T, 2T, ..., NT})

\[ y(t) = y_0 + g_0r(0) + \sum_{i=1}^{t/T-1} g_i[r(t-iT) - r(t-(i+1)T)] \]

(1)

where \( g_i = g(iT), i = 1, ..., N \), are the sampled output values in response to a unit-step input (see Figure 2) and \( r(t) \) is the system input. In the following discussion, the value of \( y_0 \) will be taken to be 0 without loss of generality. The number \( N \) of parameters has to be taken high enough to allow a sufficiently accurate description of the system, but not so high that the computational effort of the control strategy is unmanageable. From a practical point of view, the sampling of the step response to obtain parameters \( g_i \) should stop when the process output remains close to its steady-state value for a sufficiently long time.

For the presented methodology, it is convenient to write expression 1 in matrix form

\[ Y = GR \]

(2)

where

\[
G = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\
-g_1 + g_2 & g_1 & 0 & \cdots & 0 \\
-g_2 + g_3 & -g_1 + g_2 & g_1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
-g_N-1 + g_N & -g_N-2 + g_N-1 & -g_N-3 + g_N-2 & \cdots & g_1 \end{bmatrix}
\]

and

\[
R = \begin{bmatrix} r(0) \\
r(T) \\
\vdots \\
r((N-1)T) \end{bmatrix}
\]

Remark 1. It should be noted that, in many cases, it might not be necessary to perform an ad hoc identification experiment (i.e., to stop the normal process operations) to apply the devised methodology. In fact, as the model is obtained by evaluating a standard closed-loop step response, data taken from an output transition performed during normal process operations can be used. Obviously, it is important that the collected data be representative of a true step response (and therefore operations such as detrending might be necessary) and, if an unmeasured load disturbance occurs during the transient response, then the data should not be used. In this context, it can be useful to adopt the method proposed in ref 6 to detect load disturbances.

2.3. Output Function Selection. To apply the dynamic inversion procedure, a desired output function has to be chosen to describe a transition from one set-point value to another. Although, in principle, any function can be applied, an effective choice is to adopt a “transition polynomial” that guarantees a smooth and monotonic (i.e., without overshoot) transition between \( y_0 = 0 \) and \( y_1 \) and can be easily parametrized by the transition time \( r \) that can eventually be selected to fully exploit the actuator capability (see remark 4). Formally, we define

\[ y_d(t) = c_{2p+1}t^{2p+1} + c_{2p}t^{2p} + \cdots + c_1t + c_0 \]

The polynomial coefficients can be uniquely found by solving the following linear system, in which boundary conditions at the endpoints of interval \([0, r]\) are imposed

\[
\begin{cases}
y_d(0) = 0, & y_d(r) = y_1 \\
y_d^{(1)}(0) = 0, & y_d^{(1)}(r) = 0 \\
\vdots & \vdots \\
y_d^{(p)}(0) = 0, & y_d^{(p)}(r) = 0
\end{cases}
\]

Thus, the desired output function can be expressed in closed form as

\[
y_d(t; r) = \begin{cases}
(2p + 1)! & \frac{(-1)^{p-i}}{p!}\left(\frac{t^{2p+1}}{2p+1}\right)^{p-i} \\
& \frac{(-1)^{p-i}}{p!}\left(\frac{t^{2p+1}}{2p+1}\right)^{p-i} if 0 \leq t \leq r \\
& \frac{(-1)^{p-i}}{p!}\left(\frac{t^{2p+1}}{2p+1}\right)^{p-i} if t > r
\end{cases}
\]

(3)

If the algorithm described in subsection 2.4 is employed, then the order of the transition polynomial can be chosen arbitrarily. (Note that this does not apply when the dynamic inversion is performed analytically, based on the continuous-time transfer function of the system. In that case, the order of the polynomial has to be chosen depending on the relative degree of the system to guarantee the continuity of the command function.) Indeed, the order of the polynomial can be chosen to handle the tradeoff between the need to decrease the rise time and the need to decrease the control effort, taking into account that the rise time decreases and the control effort increases as the order of the polynomial increases. In general, a good choice in this context is to select \( p = 2 \), i.e., to use the desired output function

\[
y_d(t; r) = \begin{cases}
\frac{6t^5}{r^5} - \frac{15t^4}{r^4} + \frac{10t^3}{r^3} & if 0 \leq t \leq r \\
\frac{6t^5}{r^5} - \frac{15t^4}{r^4} + \frac{10t^3}{r^3} & if t > r
\end{cases}
\]

(4)

Expression 4 is employed in all of the simulation examples presented in section 3.

2.4. Dynamic Inversion. Once the desired output function has been selected, i.e., the array \( Y_d \) has been constructed, then the corresponding closed-loop system input \( r(t) \) that causes \( y_d(t; r) \) can easily be determined by simply inverting the system using eq 2. For matrix
\( \mathbf{G} \) to be invertible by a standard numeric algorithm, it should be well-conditioned; for example, there must not be a row (or a column) where all of the elements are very small with respect to the elements of the other rows (or columns). This happens when the process has a true dead time or an apparent dead time (i.e., when the process is of high order), which causes some of the first sampled output values \( y \) of the step response to be null or almost null. Thus, we denote by \( k \) the number of first rows of \( \mathbf{G} \) in which all of the elements are less than a selected threshold \( \epsilon \). Then, we obtain matrix \( \mathbf{G} \) by removing the first \( k \) rows and the last \( k \) columns from \( \mathbf{G} \). Subsequently, by evaluating \( y_0(\tau;\tau) \) at the first \( N-k \) sampling time intervals, the array \( \mathbf{Y}_d = [y_0(T;\tau) \ y_{d_1}(2T;\tau) \cdots \ y_{d_l}(N-kT;\tau)]^T \) can easily be constructed. The first \( N-k \) values of the command reference input are then determined by applying the expression

\[
\mathbf{R} = [r(T) \ r(2T) \cdots \ r((N-k)T)]^T = \mathbf{G}^{-1}\mathbf{Y}_d \quad (5)
\]

In this way, the input function can be calculated simply by determining the inverse of a matrix, which can be performed by using different algorithms (see, e.g., ref 7).

Remark 2. Because the first \( k \) rows and the last \( k \) columns have been removed from matrix \( \mathbf{G} \), the obtained output function is delayed by \( kT \) with respect to the desired output function. Actually, the dead time is removed in the model of the closed-loop system transfer function employed in the dynamic inversion.

Remark 3. Because the proposed methodology is capable of ensuring good set-point-following performance despite the tuning procedure adopted for the controller, it is convenient to design the controller to guarantee good load disturbance performance, even if this implies that the predicted closed-loop step response is unsatisfactory. In other words, in the case that the noise measurements, it is sensible to redefine parameter \( \epsilon \) as a noise band \( NB \), i.e., a threshold value that determines, as before, whether the sampled value \( y \) has to be discarded. Specifically, if \( |y| < NB \), then \( y \) is considered to be 0 in the construction of matrix \( \mathbf{G} \). The value of \( NB \) can easily be selected by monitoring that process output for a sufficiently long time when the process is at steady state. Note that the concept of a noise band has already been applied successfully in the implementation of autotuning procedures in industrial regulators.\(^9\)

### 3. Simulation Results

Some simulation examples are presented to illustrate the proposed methodology and to demonstrate its effectiveness. For the sake of clarity, measurement noise will be taken into account only in the last example. For all of the presented examples, the controller has a PID structure described by the following expression

\[
U(s) = K_p \left[ E(s) + \frac{1}{T_i} E(s) - T_d s \ Y(s) \right] \quad (6)
\]

where \( U(s) \), \( E(s) \), and \( Y(s) \) are the Laplace transforms of the control variable, system error, and process output, respectively. The required output transition is from \( y_0 = 0 \) to \( y_1 = 1 \).

#### 3.1. FOPDT Process

As a first example, we consider the first-order plus dead-time (FOPDT) process

\[
P_1(s) = \frac{1}{10s + 1} e^{-5s} \quad (7)
\]

with the following values of the PID parameters: \( K_p = 2.61 \), \( T_i = 10.05 \), and \( T_d = 2.51 \). The sampling time is fixed at 0.5 s. To obtain the closed-loop system model, \( N = 161 \) samples of the step response are evaluated, and matrix \( \mathbf{G} \) is constructed accordingly. By fixing \( \epsilon = 0.01 \), one obtains \( k = 11 \). Thus, the first 11 rows and the last 11 columns of \( \mathbf{G} \) are removed, yielding matrix \( \mathbf{G} \) of dimensions \( 150 \times 150 \). The desired output array is constructed by selecting \( \tau = 5 \) s. The resulting command function \( r(t) \), obtained by applying the dynamic inversion, is reported in Figure 3. A comparison of the process outputs obtained by using \( r(t) \) and the standard step signal is shown in Figure 4, and the corresponding control variables are plotted in Figure 5. It is evident that the use of the calculated command function provides a great improvement in the set-point-following performance. In any case, to better compare the two approaches, the resulting overshoot \( O(\%) \), rise time \( T_r \) (defined as the interval time in which the process output passes from 10 to 90% of its steady-state output), settling time \( T_s \) (defined as the time at which the regulated output remains within a 2% range of \( y_1 \)), and integrated absolute error

\[
IAE = \int_0^{T_r} |e(t)| \, dt
\]
were calculated and are reported in Table 1. It appears that the overshoot is significantly reduced without a increase in the rise time, so that the settling time is greatly reduced as well.

<table>
<thead>
<tr>
<th></th>
<th>O (%)</th>
<th>T_r (s)</th>
<th>T_s (s)</th>
<th>IAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>step input</td>
<td>39</td>
<td>3.07</td>
<td>49.6</td>
<td>10.82</td>
</tr>
<tr>
<td>dynamic-inversion-based</td>
<td>7</td>
<td>2.54</td>
<td>28.8</td>
<td>8.86</td>
</tr>
</tbody>
</table>

**Figure 3.** Command function for the example with $P_1(s)$.

**Figure 4.** Process outputs for the example with $P_1(s)$.

**Figure 5.** Control variables for the example with $P_1(s)$.

**High-Order Process.** As a second example, we consider the high-order process

$$P_2(s) = \frac{1}{(s+1)^8e^{-5s}}$$

(8)

with the following values of the PID parameters: $K_p = 0.37$, $T_i = 19.93$, and $T_d = 4.98$. In this case, the sampling time is fixed at 2 s, and a number $N = 101$ of output samples is used to model the closed-loop system. By again fixing $\epsilon = 0.01$, we obtain $k = 5$, and therefore, $G$ is of dimensions $96 \times 96$. With the selection of $r = 20$ s, the command input function $r(t)$ plotted in Figure 6 was determined. A comparison between the process outputs obtained with the standard and the new approach is shown in Figure 7, and the corresponding control variables are plotted in Figure 8. As in the previous example, the dynamic-inversion-based systems outperforms the one with the step signal. The results are summarized in Table 2.

**3.3. Process with Lag-Dominant Dynamics.** Processes with lag-dominant dynamics, which are frequently encountered in industry, especially in fluid processes, are likely to present a large overshoot in the
set-point response when the controller is tuned for good performance in load rejection. Thus, as a third example, we consider a distillation column in which the top composition (or temperature) is measured. The response to a change in reflux flow can be modeled as an FOPTD process in which the time constant is

\[ \theta = \frac{n^2 + n'}{2} \]

where \( n \) is the number of trays and \( n' \) is the time constant of a single part of the process, and the dead time can be expressed by

\[ L = 0.14\theta. \]

If \( n = 100 \) and \( n' = 5 \) s, one obtains \( \theta = 7 \) h and \( L = 0.98 \) h. Thus, the following transfer function is considered

\[ P_3(s) = \frac{1}{7s + 1} e^{-0.98s} \] (9)

The following values of the PID parameters were selected: \( K_p = 8.26, T_i = 2.01, \) and \( T_d = 0.5 \). In this case the sampling time is fixed to 0.08 h, and \( N = 126 \) output samples are employed to model the closed-loop system. By again fixing \( \epsilon = 0.01, \) we obtain \( k = 13, \) and therefore, the resulting matrix \( G \) dimensions are 113 \( \times \) 113. Selecting \( r = 1.6 \) h, the command input function \( r(t) \) shown in Figure 9 is determined. The process outputs obtained with both the standard and the new approach are plotted in Figure 10, and the corresponding control variables are plotted in Figure 11. The results are summarized in Table 3. It appears that, also in this case, the dynamic-inversion-based approach outperforms the one with the step signal.

3.4. Effects of Measurement Noise. To evaluate the effects of measurement noise on the performance obtained with the proposed dynamic-inversion-based technique, we again considered process \( P_2(s) \) (see eq 8) with the output corrupted by white noise. In fact, in the simulations, a random number in the interval \([-0.1, 0.1]\) was added to the process output at each sample time. The process output signal was filtered using a sixth-order Butterworth filter with a cutoff frequency of 0.06 Hz. (Note that the sampling time is 0.5 s in this case.) Then, NB was set to 0.1, and the same procedure as in the noise-free case was applied. The resulting command input signal is plotted in Figure 12, and the corresponding output is shown in Figure 13. It can be observed that measurement noise does not prevent the achievement of high performance, provided that appropriate
filtering is applied to the process output signal before the devised algorithm is employed.

4. Conclusions

In this paper, a dynamic-inversion-based methodology for improving the set-point-following performance of a control scheme has been proposed. The methodology relies on a fast and simple algorithm that is based on the calculation of the inverse of a matrix. Actually, because of its features, the method appears to be suitable for application as an add-on functionality in single-station industrial controllers and also in distributed control systems, as the performance of many control loops can be significantly improved by applying the determined command functions instead of the step signals without the need to retune the controllers.

Acknowledgment

This work was partially supported by MIUR scientific research funds.

Literature Cited

(10) Shinskey, F. G. PID—deadtime control of distributed processes. Presented at the IFAC Workshop on Digital Control PID’00, Terrassa (E), Terrassa, Spain, Apr 5–7, 2000.

Received for review September 18, 2002
Revised manuscript received January 15, 2003
Accepted February 10, 2003

IE020734F