Indirect and Implicit Adaptive Predictive Control of the Benchmark Plant*†

L CHISCI,‡ L GIARRÈ‡ and E MOSCA‡

The performance of two adaptive predictive controllers—one of an indirect type the other of an implicit type—are compared, particularly in terms of their different ability to work with or without low-pass prefiltering the data entering the identifiers

Key Words—Adaptive control, predictive control, benchmark problem, identification, indirect and implicit adaptive control

Abstract—Two different adaptive predictive controllers are used for controlling the benchmark plant. One is of an indirect type, the other of an implicit type. In both cases, the only a priori knowledge used is that the nominal process order equals three.

1 INTRODUCTION

Two different adaptive predictive controllers, one of an indirect type and the other of an implicit type, have been used to control the benchmark plant proposed by Graebe (1994). The indirect adaptive controller is based on the Stabilizing Input–Output Receding Horizon Control (SIORHC) (Mosca et al., 1990, Clarke and Scattolini, 1991, Mosca and Zhang, 1992, Chisci and Mosca, 1993, 1994). SIORHC improves the stability properties of the Generalized Predictive Controller (GPC) of Clarke et al. (1987) guaranteeing closed-loop stability for any stabilizable plant. This is achieved by adding terminal output constraints to the input terminal constraints of GPC. The implicit adaptive controller, known under the acronym MUSMAR was first reported in Menga and Mosca (1980). Thanks to its implicit nature, MUSMAR was shown to be robustly self-optimizing against undermodelling (Mosca et al., 1989).

In applying the two above adaptive techniques to control the benchmark plant, only knowledge of the nominal process order was used in order to set the regressor size. In both controllers the plant deadtime was assumed to possibly equal the intrinsic one, being the two adopted control laws capable of dealing with any deadtime equal to or larger than the assumed one and smaller than the prediction horizon. In fact, no effort was made to guess a plant deadtime and a model from preliminary experiments, our approach having been instead to use directly each of the two adaptive control techniques under the stress.

1 case to autotune a controller to be used at the startup of each subsequent adaptive experiment.

2 INDIRECT ADAPTIVE PREDICTIVE CONTROL

An indirect adaptive predictive control algorithm, RLS + SIORHC, obtained by the combination of a recursive least-squares (RLS) identifier with the SIORHC predictive controller, is considered SIORHC is capable of stabilizing any stabilizable linear plant provided that the prediction horizon \(T\) is chosen larger than or equal to the plant order. Further, it ensures unit dc gain from the reference to the output provided that the loop has integral action.

Plant model

It is assumed that, in the frequency band of interest, the time-discretized plant is adequately described by the locally-linearized model

\[
A(d)y(t) = B(d)u(t) + \eta(t),
\]

where \(y(t)\) and \(u(t)\) are the output and, respectively, input sequences sampled every \(T_s\), \(\eta(t)\) is a disturbance term, \(d\) is the unit delay.
operator, viz. \( dy(t) = y(t-1), \) \( A(d) \) and \( B(d) \) are polynomials in \( d \)

\[
A(d) = 1 + a_1 d + \cdots + a_n d^n, \\
B(d) = b_1 d + \cdots + b_m d^m.
\]

**SIORHC**

Consider the problem of tracking a reference sequence \( \{r(t)\} \) known by the controller up to \( T > 0 \) steps in advance. The SIORHC criterion amounts to selecting a control sequence \( u(t), \ldots, u(t+T-1) \) which minimizes the quadratic cost

\[
f = \frac{1}{T} \sum_{k=0}^{T-1} \{y_H(k) - r_C(k)\}^2 + \rho u_H^2(k)
\]

subject to the terminal constraints

\[
u_H(t+T+k) = 0, \quad k = 0, 1, \ldots, n-2
\]

\[
\varepsilon \{y_H(t+T+k)\} = r_C(t+T), \quad k = 0, 1, \ldots, n-1,
\]

where \( n = \max\{n_a, n_p\}, \) \( y_H, u_H \) and \( r_C \) are suitably filtered versions of \( y, u \) and \( r \).

\[
y_H(t) = H(d)y(t), \quad u_H(t) = H(d)u(t), \quad r_C(t) = G(d)r(t)
\]

\( H(d) \) is a strictly Hurwitz polynomial of degree \( n_a \) and \( G(d) \) is a stable transfer function, with \( H(1) = G(1) = 1 \). In (3)–(4) \( \varepsilon \{ \cdot \} \) denotes conditional expectation with respect to the past data \( \{y(k), u(k-1), k \leq t\} \) and under the assumption that \( \{\eta(t)\} \) in (1) are independent identically distributed random variables. From the optimal solution of (1)–(4), a time-invariant two degree of freedom control law

\[
R(d)u_H(t) = -S(d)y_H(t) + Z(d)r_C(t+T)
\]

is obtained by using, according to the receding-horizon control philosophy, as input to the plant the first element of the optimal control sequence at each time. In (6) \( R(d), S(d) \) and \( Z(d) \) are polynomials in \( d \) of degrees

\[
n_r = \max(n_a-1, n_h), \quad n_s = \max(n_a-1, n_h-1), \quad n_s = T - 1 + n_h,
\]

with \( R(d) \) monic.

**Design knobs**

SIORHC is equipped with two design knobs, viz. the prediction horizon \( T \geq n \) and the control effort weight \( \rho > 0 \). In Mosca et al. (1990), it is shown that by varying these knobs SIORHC covers a wide family of different control laws, ranging from state-deadbeat to LQG control. In addition, the shaping filters \( H(d) \) and \( G(d) \) have been included in the SIORHC formulation (1)–(5) to allow suitable frequency shaping of the loop and, respectively, reference signals.

The ‘loop’ FIR filter \( H(d) \) can be used to penalize the high-frequency components of \( u(t) \) and \( y(t) \). To this end, the polynomial \( H(d) \) must possess high-pass characteristics. Conversely, the filter \( G(d) \) is used to provide an independent low-pass filtering action on the reference signal. In fact, it can be shown that the resulting closed-loop characteristic polynomial divides \( H(d) \), viz. \( A(d)R(d) + B(d)S(d) = n(d)Q(d) \) for some strictly Hurwitz polynomial \( Q(d) \) and that also \( Z(d) = H(d)V(d) \) for some polynomial \( V(d) \).

Consequently, the output of the closed-loop system turns out to be

\[
y(t) = \frac{B(d)V(d)}{Q(d)} \left[ \frac{G(d)}{H(d)} \eta(t) \right]
\]

The above equation shows that the filters \( H^{-1}(d) \) and \( G(d)H^{-1}(d) \) can be designed so as to suitably low-pass filter the disturbance \( \eta(t) \) and, respectively, the reference signal \( r(t) \), in an independent way.

**Plant identification**

An RLS algorithm, implemented in square-root factorized form, with directional forgetting (Kulhavy and Karny, 1984) has been used for estimating the parameters \( \theta \) in the linear regression

\[
y_L(t) = q'(t-1)\theta + \xi(t),
\]

where

\[
q'(t-1) = [-y_L(t-1) - y_L(t-2) - y_L(t-3)
\]

\[
u_L(t-1)u_L(t-2)u_L(t-3)1']^T,
\]

\[
\theta = [a_1 a_2 a_3 b_1 b_2 b_3 ]^T,
\]

\( \xi(t) \) is an equation error, and \( y_L(t) = L(d)y(t) \), \( u_L(t) = L(d)u(t) \), are the input, output, data filtered via a stable transfer-function \( L(d) \). The filter \( L(d) \) allows to low-pass filter the data entering the identifier so as to counteract the effects of unmodeled dynamics and high-frequency noise. In SIORHC, but not in MUSMAR, a one has been included in the regressor \( q'(t-1) \) so as to remove possible external low-frequency disturbances.

### 3 IMPLICIT ADAPTIVE PREDICTIVE CONTROL

**MUSMAR** is an implicit adaptive predictive controller. The word ‘implicit’ refers to the fact that in MUSMAR several multistep-ahead output predictors are directly identified on-line instead of being indirectly computed from an
explicitly identified plant model. The MUSMAR version, which was found most suitable for controlling the benchmark plant, adaptively selects the one degree of freedom control law

$$\Phi(d)u_I(t) = - \Phi(d)[y_I(t) - r_G(t)]$$  \hspace{1cm} (10)$$

so as to minimize the quadratic cost (3) under the constraints that over the $T$-steps prediction horizon all controls, except the first, are given by a constant control law. Since MUSMAR, thanks to its direct closed-loop estimates of the multistep-ahead output predictors, turns out to possess robust self-optimizing properties, it can allow one to work with no $L(d)$ prefiltering. Therefore, in all MUSMAR simulation experiments we set $L(d) = 1$, and the control weight $\varphi$ was chosen so as to achieve on the average the best possible performance. For a description of MUSMAR the reader is referred to the pertinent references quoted in the introductory section.

4 EXPERIMENTS

For both controllers it was decided to use the lowest possible sampling time equal to $T_s = 50$ ms, even if slower sampling rates could have been used to get similar results.

RLS + SIORHC

In order to work with a reduced-order controller which assumes a plant order equal to three, it was essential to low-pass prefilter the data as in (8) and (9). A suitable choice for the $L(d)$-filter is to use a third-order Chebyshev low-pass filter with normalized cutoff frequency $\omega_c T_s = 0.4$ rad, obtained by the MATLAB call CHEBY1(3, 0.5, 0.4/ps). With no data prefiltering in the identification, it was impossible to operate. The RLS identifier used a directional forgetting with forgetting factor $\lambda = 0.98$. The other controller design knobs were set, according to the simulation results, as follows.

$$\Phi(d)u_I(t) = - \Phi(d)[y_I(t) - r_G(t)]$$  \hspace{1cm} (10)$$

For all stress conditions, $G(d)$ was set equal to one. It is to be pointed out that the highpass FIR filter $H(d)$ turned out to yield a significant noise rejection on the closed-loop output. The prediction horizon $T$ was set as large as 30 steps. Smaller values of $T$ yield, in fact, an appreciable performance degradation. A preliminary auto-tuning under stress 1 was carried out to suitably initialize the estimates of the identifier.

Figures 1-3 show the typical behavior of the plant output and input for RLS + SIORHC under different stress conditions. The full-order controller, which assumes a plant order equal to seven, was found to behave similarly to the above discussed reduced-order controller, even if for the latter is essential the preliminary search for and the constant use of a suitable low-pass data prefiltering RLS + SIORHC was used also to adaptively autotune a constant controller. This was found acceptable for stress 1 but not for stresses 2 and 3, where it was impossible to fulfill the requirement $|y(t)| < 1.5$. A typical behavior of the autotuned SIORHC constant controller is reported in Fig 4.

MUSMAR

A convenient identification algorithm in MUSMAR is the RLS with constant trace and data normalization (CT-RLS) analyzed in (Lozano-Leal and Goodwin, 1985) under ideal conditions. This algorithm, implemented in a square-root factorized form, was adopted for all the simulation experiments reported hereafter. Simulations of the MUSMAR controlled plant over the three stress conditions suggested the adoption of the following choices $T = 7, H(d) = 1$, and $G(d) = 0.08/(1 - 0.92 d)$. In particular, here the use of $\varphi = 1$, while making the controller action as regular as required, yields a steady-state offset which on the average can be compensated by changing the $G(d)$ filter into the filter $gG(d)$ with $G(d)$ as above and $g = 1.43$. With the above choices, MUSMAR autotuning under stress 1 was carried out. This yielded the one degree of freedom 'nominal' control law

$$\Phi(d)u(t) = - \Phi(d)[y_G(t) - r_G(t)]$$  \hspace{1cm} (12)$$

$$\epsilon_G(t) \triangleq y(t) - gr_G(t)$$

This control law was used to initialize the algorithm for each subsequent simulation. During each simulation MUSMAR provided an on-line tuning of the initial polynomials $\Phi(d), \Phi(d)$ on different time-varying polynomials $\Phi(d), \Phi(d)$ in accordance to the time variations exhibited by the identified predictors. The only design knob that was varied according to the stress operating conditions was the value of the constant trace ($7 \times tr$) of the 'covariance matrix' in the CT-NRLS algorithm (here seven denotes the dimension of the regression vector $\varphi(t) \triangleq [\epsilon_G(t) \epsilon_G(t - 1) \epsilon_G(t - 2) u(t) u(t - 1) u(t - 2) u(t - 3)]'$).

To sum up, the following choices were adopted for all
Indirect and implicit adaptive predictive control

Fig 3: SIORHC + RLS output and input (stress 3)

Fig 4: AUTOTUNED SIORHC constant controller output (stress 3)
MUSMAR experiments

<table>
<thead>
<tr>
<th>Stress</th>
<th>$T$</th>
<th>$q$</th>
<th>$g$</th>
<th>$G(d)$</th>
<th>$tr$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>1</td>
<td>1.43</td>
<td>0.08/(1 - 0.92$d$)</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>1</td>
<td>1.43</td>
<td>0.08/(1 - 0.92$d$)</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>1</td>
<td>1.43</td>
<td>0.08/(1 - 0.92$d$)</td>
<td>200</td>
</tr>
</tbody>
</table>

Figures 5–7 show the output and the input of the plant controlled by MUSMAR according to the choices discussed above. A typical behaviour of the autotuned MUSMAR constant controller is reported in Fig 8.

5 CONCLUSIONS

Both adaptive controllers were found capable of controlling the plant by fulfilling the given specifications, except for the zero steady state tracking error under stress 3. In order to fulfill this requirement, control versions using input increments $\delta u(t) = u(t) - u(t - 1)$ instead of $u(t)$ should be used. However, an appropriate selection of both the design knobs and the various filters for the $\delta u(t)$-case turned out to be critical and time-consuming.

As expected, a different behaviour was experienced in using the two controllers. In particular, MUSMAR, the implicit one, turned out to require neither prefiltering for identification nor dynamic weights in the cost. This makes
its use more direct and immediate. However, in contrast with SIORHC, in a two degree of freedom version MUSMAR performance degrades significantly under stress 3.

**REFERENCES**


Chisci, L and E Mosca (1994) Stabilizing I/O Receding Horizon Control of CARMA plants IEEE Trans Aut Control


Graebe, S F (1994) Robust and adaptive control of an unknown plant a benchmark of new format Automatica, 30, 567–575


Mosca, E, J M Lemos and J Zhang (1990) Stabilizing I/O receding-horizon control Proc 29th IEEE CDC, pp 2518–2523 Honolulu, U S A
