ASYMPTOTIC REFERENCE TRACKING ANALYSIS OF MODEL-FREE AND MODEL-DEPENDENT ADAPTIVE CONTROL METHODS

ALDO ZANELLA JUNIOR*, ANTONIO S. SILVEIRA†, ADEMR NIED‡

*Group of Control and Systems, Federal Institute of Santa Catarina
Jaraguá do Sul, SC, Brazil

†Laboratory of Control and Systems, Federal University of Pará
Belém, PA, Brazil

‡Group of Control and Systems, Santa Catarina State University
Joinville, SC, Brazil

Emails: azanella@ifsc.edu.br, asilveira@ufpa.br, ademir.nied@udesc.br

Abstract—This work aims to describe the analysis of a design method called Adaptive Model-free control, evaluating its results by comparing them to a well established self-tuning adaptive scheme. The Model-free control design is a generalization of the Model Reference Adaptive System, but posed as a direct self-tuner, i.e., the controller is directly identified and not the plant model itself, adapting the control law to guarantee a prescribed closed-loop behavior settled by a reference model. On the other hand, a classical adaptive method by indirect self-tuning is used as a frame of reference for this analysis. Both methods are explored within a digital PID structure capable of guaranteeing asymptotic reference tracking governed by a prescribed first-order reference model. The analysis is conducted by simulations and by practical experiments with a real under-damped electronic circuit. Results are presented in graphics, measured errors and statistical deviation.

Keywords—Model-free control, adaptive control, self-tuning control, PID control.

1 Introduction

Adaptive control has been used more frequently in order to improve the controller performance against structural nonlinearities or temporal changes on plant model parameters. In this work, the self-tuning type through an Adaptive Model-free (MF) technique for guaranteed asymptotic tracking is considered (Silveira et al., 2012), whilst comparing its results to a plant model-dependent self-tuning technique.

To conduct the analysis of the MF and the model-dependent controller design techniques, two similar ways to implement adaptive control systems are used: the first is based on direct self-tuning, i.e., the direct estimation of the controller (Kirecci et al., 2003); the second way uses plant model online identification. For both ways, the Recursive Least-Squares (RLS) algorithm is used as parameter estimator (Åström and Wittenmark, 1995).

2 Plant description and controller syntheses

Design methods used in this analysis are based on classical indirect self-tuning, such as the one presented by Lim (1990), with estimation of plant parameters; and the MF case, that consists in presetting a reference model (RM) for the closed-loop behavior that will govern the adaptation of the control law in order to guarantee the matching between the RM output and the real plant controlled output itself, that may have completely different open-loop characteristics. To give more emphasis to this, an under-damped plant will be put to behave like a first-order RM in order to have asymptotic reference tracking.

2.1 Plant description

The physical plant developed to conduct all considered experiments in this work consists in an active second-order RC filter made of operational amplifiers, resistors and capacitors. Its continuous-time model is given by

\[ G(s) = \frac{1}{s^2 + \frac{1}{(2RC)^2} s + 1/(2RC^2)}, \]

and using the sampling period \( T_s = 0.1 \) seconds, the ZOH equivalent is given by

\[ G(z^{-1}) = \frac{B(z^{-1}) z^{-1}}{A(z^{-1})} = \frac{(b_0 + b_1 z^{-1}) z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}, \]

\[ G(z^{-1}) = \frac{(0.1672 + 0.1438 z^{-1}) z^{-1}}{1 - 1.329 z^{-1} + 0.6399 z^{-2}}. \]

This plant is under-damped in open-loop, with a damping ratio of 0.35 and natural frequency of 6.31 rad/s. The I/O range is of 0V to 5V.

2.2 Self-tuning PID controller design

This first design method presented is based on Lim (1990), which describes a way to force the closed-loop behavior by cancelling the open-loop poles of the plant – which leads to a design technique applicable only to stable systems (Seborg et al., 2004). The goal is to impose a first-order
behavior, settled by the RM where a single tuning parameter is required. This is the closed-loop time constant, \( \tau_{cl} \), of the RM, so that a discrete RM based on a real discrete pole \( z_D \) can be specified as

\[
y(k) = \frac{(1 - z_D) z^{-1}}{1 - z_D z^{-1}},
\]

where \( y(k) \) is the output, \( y_r(k) \) is a reference sequence, and the discrete real pole relationship with \( \tau_{cl} \) is given by \( z_D = e^{-\frac{\tau_{cl}}{T_s}} \).

The RLS estimator is used to determine the plant parameters in this model-dependent approach, in order to produce controller zeros to cancel the plant open-loop poles. For this, consider the digital PID controller,

\[
U(z^{-1}) = \frac{s_0 + s_1 z^{-1} + s_2 z^{-2}}{1 - z^{-1}},
\]

where \( E(z^{-1}) \) denotes the error between \( y_r(k) \) and \( y(k) \), and the PID model-based tuning is given by:

\[
s_0 = \frac{(1 - z_D)}{B(1)} = \frac{(1 - z_D)}{b_0 + b_1},
\]

\[
s_1 = \hat{a}_1 s_0,
\]

\[
s_2 = \hat{a}_2 s_0,
\]

using the hat (‘) symbol to denote estimated parameters of a model as shown in (2).

The RLS algorithm needs to be structured to identify a second order system – which is readily available in (Aström and Wittenmark, 1995).

### 2.3 Model-free control

The objective is to produce a control sequence in order to guarantee the matching of the measured plant output with the RM output, similar to Model Reference Adaptive System, MRAS (Aström and Wittenmark, 1995), but in the MF technique the RM output is not simulated (the RM output is not generated), but an analytical approach is considered. Another interesting feature is that the controller can assume a PID or I+PD form, whereas the indirect self-tuning PID shown in Section 2.2 cannot.

The analytical approach of RM in the MF design considers that the RM in (4) is rewritten in the following form (Silveira et al., 2012):

\[
\frac{Y_m(z^{-1})}{Y_r(z^{-1})} = \frac{B_m(z^{-1}) z^{-1}}{A_m(z^{-1})},
\]

where \( Y_m(z^{-1}) \) is the RM output.

The RM shown in (9) is now generalized in such a way it could cope with the requirements of a large number of linear discrete systems.

### 2.3.1 PID Model-free control synthesis

The MF control problem, using a digital PID controller, is to adapt the control law shown in (5) to eliminate the error \( E(z^{-1}) \) and simultaneously guarantee that \( Y(z^{-1}) \rightarrow Y_m(z^{-1}) \). Considering \( \Delta = 1 - z^{-1} \), the PID control law could be rewritten into an adaptive form, where

\[
\Delta U(z^{-1}) = (\hat{s}_0 + \hat{s}_1 z^{-1} + \hat{s}_2 z^{-2}) E(z^{-1}),
\]

\[
\Delta U(z^{-1}) = \hat{S}(z^{-1}) \left[Y_r(z^{-1}) - Y(z^{-1})\right].
\]

In order to guarantee that the error \( E(z^{-1}) \) will vanish with time and simultaneously \( Y(z^{-1}) \rightarrow Y_m(z^{-1}) \), i.e. \( Y(z^{-1}) = Y_m(z^{-1}) \), Eq. (9) is reorganized to

\[
Y_r(z^{-1}) = \frac{A_m(z^{-1})}{B_m(z^{-1}) z^{-1}} Y(z^{-1}),
\]

which is substituted into (11), leading to

\[
\Delta U(z^{-1}) = \hat{S}(z^{-1})\left[\frac{A_m(z^{-1})}{B_m(z^{-1}) z^{-1}} Y(z^{-1}) - Y(z^{-1)}\right],
\]

\[
B_m(z^{-1}) z^{-1} \Delta U(z^{-1}) = \hat{S}(z^{-1}) \left[A_m(z^{-1}) - B_m(z^{-1}) z^{-1}\right] Y(z^{-1}),
\]

\[
U_{bm}(z^{-1}) = \hat{S}(z^{-1}) Y_{abm}(z^{-1}).
\]

It is important to observe that if we find \( \hat{S}(z^{-1}) \) that satisfies the equality shown in (15), we also satisfy (11) by forcing \( Y(z^{-1}) \rightarrow Y_m(z^{-1}) \) through (12). Since \( U_{bm}(z^{-1}) \) and \( Y_{abm}(z^{-1}) \) in (15) are realizable, they can be considered as generalized outputs based on past and up to present data of the plant input and output sequences. That is,

\[
U_{bm}(z^{-1}) = B_m(z^{-1}) z^{-1} \Delta U(z^{-1}),
\]

\[
Y_{abm}(z^{-1}) = \left[A_m(z^{-1}) - B_m(z^{-1}) z^{-1}\right] Y(z^{-1}).
\]

The RLS algorithm can be used to estimate \( \hat{S}(z^{-1}) \) based on (16) and (17) if the RLS problem is posed in the following manner:

\[
\hat{u}_{bm}(k) = \phi^T \theta,
\]

\[
\phi^T = \left[ \begin{array}{ccc} y_{abm}(k) & y_{abm}(k-1) & y_{abm}(k-2) \end{array} \right],
\]

\[
\theta^T = \left[ \begin{array}{ccc} \hat{s}_1 & \hat{s}_2 & \hat{s}_3 \end{array} \right],
\]

where \( \hat{u}_{bm}(k) \) is an estimate of \( u_{bm}(k) \) based on the regressors vector \( \phi^T \) and parameters vector \( \theta \). When \( \hat{u}_{bm}(k) \rightarrow u_{bm}(k) \), not only the servo problem is guaranteed, but also the closed-loop dynamics would match the RM dynamics.

Specifically for the first-order model shown in (4), equations (16) and (17) could be implemented by the following two recursive equations:

\[
u_{bm}(k) = (1 - z_d) \Delta y(k-1),
\]

\[
y_{abm}(k) = y(k) - y(k-1) = \Delta y(k).
\]
The RLS estimation error is observed by looking to the measurable \( u_{bm}(k) \) computed by (21) and comparing it to the estimated one shown in (18). In this way, the algorithm would converge and adapt the digital PID control law with innovations of the estimated \( S(z^{-1}) \) polynomial.

### 2.3.2 I+PD Model-free control synthesis

The MF control problem for the I+PD case changes only due to the differences posed by the structure of the controller (Jesus and Barbosa, 2015). The I+PD, from the discrete case viewpoint, merges the benefits of the digital PID to the regulatory problem with the benefit of the incremental structure avoids the derivative kick, generally leading to a more conservative tuning. Due to the differences, a common ground between these control structures was defined within its parameters, based on what they share in the continuous case: the Proportional, Integral and Derivative gains.

To ensure the same initial conditions to both design techniques, a common nominal controller was used during initial iterations. The used nominal controller in the continuous case was manually tuned by trial and error with: \( K_c = 0.1, \ T_i = 0.1, \ T_d = 0.05 \), where these parameters were mapped to the discrete PID or I+PD controller by the following (backward approximation) equations (Åström and Wittenmark, 1995):

\[
\begin{align*}
    s_0 &= K_c \left( 1 + \frac{T_d}{T_i} + \frac{T_d}{T_s} \right), \\
    s_1 &= -K_c \left( 1 + \frac{2T_d}{T_s} \right), \\
    s_2 &= K_c \frac{T_d}{T_s}.
\end{align*}
\]

The sampling period \( T_s = 0.1s \) was used in all simulations and practical experiments in this work.

### 3 Simulation tests and results

#### 3.1 Without load disturbance

To initialize all control methods closer to the same initial state, the PID described in Section 2.4 was used during the first 19 seconds of every test. Also, the step response of the desired RM in (4) was used to generate a standard output response in order to measure errors with respect to the RM.

Simulations of the algorithms were run to compare theoretical and practical results. In Figure 1 it is shown the simulation of the indirect self-tuning case, with a desired \( \tau_{cl} = 0.8s \), that resulted in a discrete pole \( z_d = 0.88 \).

![Figure 1: Simulated indirect self-tuning.](image-url)
The self-tuning PID method presented a first-order behavior with nearly 0.78 seconds of $\tau_{ci}$. This slight deviation from the theoretical desired 0.8 seconds occurred because the pole cancellation method used, considers an static approximation of the estimated plant $\hat{B}(z^{-1})$ polynomial, i.e. $\hat{B}(1) = b_0 + b_1$, as presented in (6). The measured error between desired RM response and the control-loop response was 0.65% and the standard deviation was 1.78%. It is also interesting to notice the small derivative kick in the control signal of Figure 1, happening during every setpoint change after the 19th second, when the indirect self-tuning was started.

In Figure 2, obtained results with the PID MF method are presented, also depicting a first-order behavior. However, $\tau_{ci} = 0.89s$ was observed. The measured error to the reference model response was of 2.22% and standard deviation was of 4.12%, demonstrating by these statistical selected metrics and by simulation, that the performance of the PID MF method was around 2-times worse than the indirect self-tuning method, but there is no derivative kick in the control signal, even with a PID structure in use.

![Figura 2: Simulated PID MF method.](image)

In Figure 3, the results of the I+PD MF method are presented. The $\tau_{ci} = 0.74s$ was better than the previous PID MF method result. The measured error was of 1.37% and standard deviation of 3.82%. Near 2-times worse than the indirect self-tuning, but 7% better than the PID MF method.

![Figura 3: Simulated I+PD MF method.](image)

### 3.2 Under load disturbance

Disturbance influence was tested only by simulations, since the real physical plant do not have this test feature. The disturbance $d(k) = 0.1V$ is applied at the instant of 32 seconds. The plant simulation model was augmented with a load disturbance model of the form

$$Y(z^{-1}) = \frac{B(z^{-1})z^{-1}}{A(z^{-1})}U(z^{-1}) + \frac{1}{A(z^{-1})}D(z^{-1}),$$

(37)

where $D(z^{-1})$ is the load disturbance model input. Note that the load disturbance model shares the same undamped characteristic polynomial $A(z^{-1})$ in order to verify if asymptotic disturbance recovery occurs.

In Figure 4, the indirect self-tuning adapted asymptotically to the disturbance, as desired, but presented a high derivative kick during the setpoint change that followed the disturbance. It is also interesting to observe that adapting to the disturbance led the control-loop to a high derivative kick that vanished later on due to the convergence of the estimation algorithm. However, using the same PID structure, but in the model-free approach, the simulation result do not present the derivative kick (cf. Figure 5).

![Figura 4: Simulated indirect self-tuning under load.](image)

The result observed in Figure 5, regarding the absence of the derivative kick for the PID MF case, may seem intriguing if we think only about the PID structure and abrupt setpoint changes. But by looking further into the MF algorithm, specifically to equations (13-15), it is evident that the estimated $S(z^{-1})$ polynomial is working with filtered
versions of $u(k)$, $y(k)$, $y_r(k)$, and these filters are given by design, based on the RM alone.

In Figure 6 the results of the simulated I+PD MF under load disturbance are shown. Since the I+PD, by structure, is already insensitive to abrupt setpoint changes regarding the derivative kick, this algorithm simulation confirms this premise and also keeps its place as the best performance against the other two methods, when $\tau_{cl}$ is considered to be as close as possible to the desired value and the avoidance of the derivative kick is guaranteed by the I+PD structure plus the MF approach and its filters given by design.

![Figure 6: Simulated I+PD MF under load.](image)

The drawback of the MF approach to both PID and I+PD is that the disturbance recovery observed was not perfectly damped. In figures 5 and 6 it is evident the oscillatory recovery.

4 Results of experimental tests

An Arduino\textsuperscript{1} Uno was used as the data acquisition (DAQ) device along with the DaqDuino\textsuperscript{2} library for Matlab. The control algorithms were handled at the Matlab side, while the Arduino board was merely handling A-D and D-A conversion.

The indirect self-tuning experimental results are shown in Figure 7. Comparing it to the desired RM response, it was found a measured error of 1.50% and standard deviation of 2.73%. Around 50% worse than observed by simulation. However, if we consider that in practice a noisy signal is corrupting the measured output data, as can be observed in Figure 7, this result is within expectations. This is proved by the observed $\tau_{cl} = 0.82$s, which is close enough to the desired value of 0.8s.

Another interesting result within the practical test with the Self-tuning PID controller, is the absence of the derivative kick in the control signal during step setpoint changes (cf. Figure 7), contradicting the simulated results (cf. Figure 1). The only explanation we have for this difference is a Plant-Model-Mismatch (Seborg et al., 2004), meaning that during the online identification procedure, the identified model did not led the Self-tuning PID to behave in a similar manner as in the simulated controller with a simulated model. However, it does not mean that the derivative kick effect would never happen again in practice with our tested plant. A power source change, an Arduino board change or a PC change may produce a slight difference to the identified model that may lead to a different tuning of the Self-tuning PID.

![Figure 7: Practical indirect self-tuning.](image)

In Figure 8 it is shown the experimental results using the PID MF controller. The time constant was 1.0 second. This is the worst result among all experimental tests and simulated ones. The measured error to the RM was 2.67% and standard deviation was 5.54%. Around 36% worse than what was observed by simulation.

![Figure 8: Practical PID MF method.](image)

Experimental results of the I+PD MF case are shown in Figure 9, with a $\tau_{cl} = 0.88$s. The measured error was 2.20% and standard deviation of 4.79%. It was 25% worse than the simulated case.

![Figure 9: Practical I+PD MF method.](image)

All tests, simulated and experimental, have been summarized in Table 1, presenting some of the metrics used to compare how every control method analyzed deviated from the RM model objective.
5 Conclusions

The MF method has presented equivalent performance when compared to model-dependent self-tuning method, since the concept of MR is what drives both methods. However, based on measured errors and standard deviations presented in Table 1, the Indirect Self-tuning (ST) PID presented better results. On the other hand, if the load disturbance is considered, the derivative kick will compromise this controller ability to guarantee the closed-loop objective (cf. Figure 4).

The advantage of the presented PID and I+PD MF method is its application in different practical situations, as in unknown and time-varying plant systems. Although the results with MF were not better than the model-dependent case, within chosen comparison metrics of measured error and standard deviation to the reference model response, the MF results were acceptable from the viewpoint of the adaptation rather than an exact match to the prescribed time constant.

Another drawback of the MF case when compared to the indirect Self-tuning method is that its analytical and direct form of development is less intuitive than the indirect adaptive case. However, when compared to the MRAS case, the presented MF synthesis for PID or I+PD, seems to be simpler to be changed in order to synthesize new MF controllers based on any other control structure, since all the designer needs to do is to substitute the model reference into the control law to find the filtered versions of the input, output and reference signals given, by design, by the MF direct self-tuning synthesis covered in this article.

With that thought on synthesizing more controllers based on the MF method presented, in terms of controller structural complexity versus plant model complexity, the MF is simpler, since it does not need to have the number of zeros equal to the number of the open-loop poles of the plant. However, it may lead to poor performance when the plant is of greater order and the selected controller topology is not capable to handle the problem imposed by the plant, whilst the indirect Self-tuning method by pole cancellation would demand a controller structure of same complexity and consequently ideal cancellation of the open-loop poles, just by increasing the order of the $S(z^{-1})$ polynomial.

It is important to remark that the MF method presented, despite structured to work as a PID or I+PD controller, is more complicated than it seems to be, due to its filters added by the substitution of the reference model within the control law. However, this form of construction leads to an overall control system similar to a dynamic compensator, i.e., control based on estimated data feedback, obeying the separation principle (Åström and Wittenmark, 2011).

The MF as a dynamic compensator may be further investigated in a future work in order to prove, analytically or empirically, its robustness to sensor failures and noise, as well as the capability to handle different classes of plants, such as open-loop unstable or type-1 (integral) systems.

We finish our conclusion depicting the major advantage of the three algorithms tested in this work, which is the fact that a single parameter is necessary in order to tune the asymptotic desired response: the closed-loop time constant, $\tau_{cl}$. We also propose as a future work, the hybridization of this muscular MF technique with the intelligent monitoring and adaptation of $\tau_{cl}$ in order to best match a desired performance criterion.

Referências


Table 1: Errors and standard deviations.

<table>
<thead>
<tr>
<th>Test</th>
<th>Method</th>
<th>Error</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulat.</td>
<td>Indirect ST</td>
<td>0.65%</td>
<td>1.78%</td>
</tr>
<tr>
<td>Simulat.</td>
<td>PID MF</td>
<td>2.22%</td>
<td>4.12%</td>
</tr>
<tr>
<td>Simulat.</td>
<td>I+PD MF</td>
<td>1.37%</td>
<td>3.82%</td>
</tr>
<tr>
<td>Practical</td>
<td>Indirect ST</td>
<td>1.50%</td>
<td>2.73%</td>
</tr>
<tr>
<td>Practical</td>
<td>PID MF</td>
<td>2.67%</td>
<td>5.54%</td>
</tr>
<tr>
<td>Practical</td>
<td>I+PD MF</td>
<td>2.20%</td>
<td>4.79%</td>
</tr>
</tbody>
</table>