Backstepping-Based Adaptive PID Control

Abder Rezak Benaskeur* and André Desbiens°

* Decision Support Systems Section
Defence Research Establishment Valcartier (DREV)
2495 Pie-XI Blvd. North, Val-Bélair (Québec), Canada G3J 1X5
Tel: (418) 844-4000 ext. 4396
Fax: (418) 844-4538
E-mail: Abderrezak.Benaskeur@drev.dnd.ca

° Corresponding author
Department of Electrical and Computer Engineering
Université Laval
Pavillon Adrien-Pouliot, Quebec City (Québec), Canada G1K 7P4
Tel: (418) 656-2131 ext. 3408
Fax: (418) 656-3159
E-mail: desbiens@gel.ulaval.ca
Abstract — This paper addresses analysis and design issues in adaptive PID control for linear second order minimal phase processes using the backstepping algorithm. The first step consists in adding an integral action to the basic backstepping algorithm to obtain a zero static error. An integrator is therefore added to the plant model and is then slid back to the controller equation at the end of the design. The control law is made adaptive without using a certainty equivalence design and is robustified even more with nonlinear damping. The resulting adaptive PID control is $u_{ce} + u_{dyn} + u_{nld}$, where $u_{ce}$ is what would be the output of the adaptive PID if a certainty equivalence-based design were used, $u_{dyn}$ compensates for the adaptation dynamics and $u_{nld}$ is a nonlinear damping term added to increase the robustness by bounding the errors, even when the adaptation is off. The resulting PID controller is hence more robust and presents better transients than the basic certainty equivalence PID version. An example compares the proposed PID to a certainty-equivalence PID.
1 Introduction

Backstepping [1] is a recursive procedure for systematically selecting the control Lyapunov functions (clf) that allows the design of adaptive controllers for a class of nonlinear processes. The design approach interlaces the computation of the control and the adaptation laws to compensate for the instabilizing effects of the parameter estimation transients. It has been demonstrated [2] that controlling the linear systems with the adaptive backstepping controllers, compared to the certainty-equivalence controllers, leads to possible significant improvements of the transient performance, without increasing the control effort. Stability analysis, which represents a major drawback of the traditional adaptive controllers, can also be easily performed via the backstepping method. This paper therefore describes how the backstepping algorithm can be used to design a robust adaptive PID controller. Also, an indirect aim of the paper is to gain a better understanding of how backstepping-based controllers work.

Nowadays, most industrial control products offer adaptive algorithms. Among them, the self-tuning and/or adaptive PID, where only a first- or a second-order model is estimated, are still the most popular. In the certainty-equivalence-based adaptive control, the basic idea is that a suitable controller can be designed on-line if, given the input/output measurements, an on-line estimation of the plant model is available. With such an approach, to tune the controller, one has to use the
plant parameter estimates as if they were the true values. The estimation module
adds nonlinear extra dynamics to the control loop that are simply neglected in the
stability and/or convergence analysis, leading to a lack of robustness. Due to the
proved superiority of backstepping over the certainty-equivalence algorithms for
the control of the linear systems [1, 2], the proposed controller yields improved ro-
 bustness properties and better transients. The standard design procedure leads to
a PD adaptive controller, since only a linear second order minimal phase paramet-
ric model is used. It is obvious that if a higher order model were used, the design
would lead to a higher order PD controller. Nevertheless, to take a full advantage
of the PID steady-state performances, it becomes imperative to introduce an inte-
gral action in the obtained controller. Besides its perturbation rejection properties,
such an integral action turns out to improve the parameter estimate convergence [3].

2 Backstepping algorithm with integral action

In this Section, an integral action is added to the backstepping algorithm. The
system input and output are respectively $u$ and $y$, while the reference trajectory
is denoted $y_r$. The following minimal-phase linear system (i.e., $B(s)$ is a Hurwitz
polynomial) is first considered

$$G(s) = \frac{L_y}{L_u} = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)} = \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_0}$$ (1)
An integral action is now inserted into the plant model to derive the PID controller

\[ G^*(s) = \frac{B(s)}{A^*(s)} = \frac{s^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^{n+1} + a_{n-1}s^n + \cdots + a_0s} \]  \hspace{1cm} (2)

The state representation of the augmented model \( G^*(s) \) is

\[ \dot{x}_1 = x_2 \\
\dot{x}_2 = x_3 \\
\vdots \\
\dot{x}_n = x_{n+1} \\
\dot{x}_{n+1} = -a_0x_2 - a_1x_3 - \cdots - a_{n-1}x_{n+1} + w \\
y = x_1 \]

where the Laplace transform of the filtered control \( w \) is

\[ W(s) = sB(s)U(s) \]  \hspace{1cm} (4)

The backstepping procedure is now applied to calculate the control \( w \). At the end of the controller design, the real control \( u \) is obtained by moving the integrator from the augmented plant model to the controller using (4)

\[ U(s) = \frac{W(s)}{sB(s)} \]  \hspace{1cm} (5)

**Step 1:** The first error variable is defined as

\[ \varepsilon_1 = y - y_r = x_1 - y_r \]  \hspace{1cm} (6)

The firstclf is chosen as

\[ V_1 = \frac{1}{2} \varepsilon_1^2 \]  \hspace{1cm} (7)
and its derivative is

$$\dot{\mathcal{V}}_1 = \varepsilon_1 \dot{\varepsilon}_1 = \varepsilon_1 \left[ x_2 - \dot{y}_r \right] \quad (8)$$

To render the later negative, $x_2$ is taken as the first virtual control. Its desired value is

$$\alpha_1 = (x_2)_d = -k_1 \varepsilon_1 + \dot{y}_r \quad (9)$$

where $k_1$ is a positive design parameter. With the above choice, (8) becomes definite negative.

**Step 2:** The new error variable is

$$\varepsilon_2 = x_2 - \alpha_1
\quad (10)$$

$$= x_2 + k_1 \varepsilon_1 - \dot{y}_r$$

An augmented cfl is introduced

$$\mathcal{V}_2 = \frac{1}{2} \varepsilon_1^2 + \frac{1}{2} \varepsilon_2^2 \quad (11)$$

Since

$$\ddot{\varepsilon}_1 = x_2 - \dot{y}_r
\quad (12)$$

$$= \varepsilon_2 - k_1 \varepsilon_1$$

the derivative of (11) is given by

$$\dot{\mathcal{V}}_2 = -k_1 \varepsilon_1^2 + \varepsilon_2 \left[ \varepsilon_1 + \dot{x}_2 - \dot{\alpha}_1 \right]
\quad (13)$$

$$= -k_1 \varepsilon_1^2 + \varepsilon_2 \left[ (1 - k_1^2) \varepsilon_1 + k_1 \varepsilon_2 + x_3 - \dot{y}_r \right]$$
Choosing $x_3$ as the second virtual control, and selecting its value to render $\dot{V}_2$ definite negative, gives

$$\alpha_2 = (x_3)_d$$

$$= (k_1^2 - 1)\varepsilon_1 - (k_1 + k_2)\varepsilon_2 + \ddot{y}_r, \quad k_2 > 0$$

**Step i:** Taking

$$\varepsilon_i = x_i - \alpha_{i-1}$$

$$V_i = \frac{1}{2} \sum_{j=1}^{i} \varepsilon_j^2$$

yields

$$\dot{\varepsilon}_{i-1} = \varepsilon_i - k_{i-1}\varepsilon_{i-1} - \varepsilon_{i-2}$$

$$\dot{V}_i = -\sum_{j=1}^{i-1} k_j \varepsilon_j^2 + \varepsilon_i \left[ \varepsilon_{i-1} + \dot{\varepsilon}_i - \dot{\alpha}_{i-1} \right]$$

The virtual control is then

$$\alpha_i = (\dot{x}_i)_d$$

$$= -k_i \varepsilon_i - \varepsilon_{i-1} + \dot{\alpha}_{i-1}, \quad k_i > 0$$

In the s-plane, this can be rewritten as

$$A_i(s) = |s\mathbf{I}_i - \mathbf{K}_i| Y_r(s) - \left(|s\mathbf{I}_i - \mathbf{K}_i| - s^i\right) X_1(s)$$

where $A_i(s)$, $Y_r(s)$ and $X_1(s)$ are the Laplace transforms of $\alpha_i$, $y_r$ and $x_1$ and where

$\mathbf{I}_i$ is the identity matrix and

$$\mathbf{K}_i = \begin{bmatrix} -k_1 & 1 & 0 & \cdots & 0 \\ -1 & -k_2 & 1 & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & -1 & -k_i-1 & 1 \\ 0 & 0 & \cdots & -1 & -k_i \end{bmatrix}$$
Step $n + 1$: Defining

\begin{align*}
\varepsilon_{n+1} &= x_{n+1} - \alpha_n \\
V_{n+1} &= \frac{1}{2} \sum_{j=1}^{n+1} \varepsilon_j^2
\end{align*}

(22)

(23)

gives

\begin{align*}
\dot{\varepsilon}_n &= \varepsilon_{n+1} - k_n \varepsilon_n - \varepsilon_{n-1} \\
\dot{V}_{n+1} &= - \sum_{j=1}^{n} k_j \varepsilon_j^2 + \varepsilon_{n+1} \left[ \varepsilon_n + \dot{x}_{n+1} - \dot{\alpha}_n \right]
\end{align*}

(24)

(25)

which leads to

\begin{align*}
\dot{\varepsilon}_{n+1} &= -k_{n+1} \varepsilon_{n+1} - \varepsilon_n
\end{align*}

(26)

and

\begin{align*}
\alpha_{n+1} &= (\dot{x}_{n+1})_d \\
&= -k_{n+1} \varepsilon_{n+1} - \varepsilon_n + \dot{\alpha}_n,
\end{align*}

(27)

$k_{n+1} > 0$

or, in the s-plane

\begin{align*}
A_{n+1}(s) &= \left| sI_{n+1} - K_{n+1} \right| Y_r(s) - \left( \left| sI_{n+1} - K_{n+1} \right| - s^{n+1} \right) X_1(s)
\end{align*}

(28)

Since, from (3),

\begin{align*}
s X_{n+1}(s) &= - \sum_{j=0}^{n-1} a_j s^{j+1} X_1(s) + W(s)
\end{align*}

(29)
the filtered control is

\[ W(s) = \sum_{j=0}^{n-1} a_j s^{j+1} X_1(s) \]

\[ + |sI_{n+1} - K_{n+1}| Y_r(s) - (|sI_{n+1} - K_{n+1}| - s^{n+1}) X_1(s) \]

\[ = \left( \sum_{j=0}^{n-1} a_j s^{j+1} + s^{n+1} \right) Y_r(s) \]

\[ + \left( - \sum_{j=0}^{n-1} a_j s^{j+1} + s^{n+1} \right) (Y_r(s) - X_1(s)) \]

(30)

The error system is given by

\[ \dot{\epsilon} = K_{n+1} \epsilon \]

(31)

where

\[ \epsilon = [\epsilon_1 \; \epsilon_2 \; \cdots \; \epsilon_{n+1}]^T \]

(32)

Since (31) is proved to be stable and to converge to zero by the Lyapunov design, the polynomial

\[ |sI_{n+1} - K_{n+1}| = F(s) \]

(33)

is Hurwitz.

3 From backstepping to PID control

3.1 PID control

For a second order model, i.e., \( m = 1 \) and \( n = 2 \), the polynomial \( F(s) \) will reduce to

\[ F(s) = s^3 + (k_1 + k_2 + k_3)s^2 + (2 + k_1 k_2 + k_2 k_3 + k_1 k_3)s + (k_1 + k_3 + k_1 k_2 k_3) \]

(34)
From (30), this results in the following filtered control

\[ W(s) = (s^3 + a_1 s^2 + a_0 s)Y_r(s) + \left[ (k_1 + k_2 + k_3 - a_1)s^2 \\
+ (2 + k_1 k_2 + k_2 k_3 + k_1 k_3 - a_0)s + (k_1 + k_3 + k_1 k_2 k_3) \right] E(s) \]  

(35)

where the Laplace transform of the tracking error is defined by

\[ E(s) = Y_r(s) - X_1(s) = Y_r(s) - Y(s) \]  

(36)

Since \( W(s) = sB(s)U(s) \), the control action is obtained by sliding back the integrator from the plant to the controller

\[ U(s) = \frac{T_d s + T_i}{B(s)} E(s) + \frac{A(s)}{B(s)} Y_r(s) \]  

(37)

where

\[ T_d = k_1 + k_2 + k_3 - a_1 \]  

(38)

\[ K_c = 2 + k_1 k_2 + k_2 k_3 + k_1 k_3 - a_0 \]  

(39)

\[ T_i = k_1 + k_3 + k_1 k_2 k_3 \]  

(40)

It is worth noting that if \( b_1 \) is zero, there will be no filtering effect in the PID equation. Consequently, a filter will be required for practical implementations. In [3, 4], it is shown how to add a filtering effect to achieve smoother control action while preserving the overall stability via a Lyapunov-based derivation approximation. For higher order plants, the proposed approach leads to higher order PID controllers.
If the reference trajectory $y_r$ in (37) is generated by filtering the setpoint $y_s$ ($\mathcal{L}y_s = Y_s(s)$) with the reference model $G_r(s)$, then (37) can be written as (see Figure 1, where $d$ and $d_i$ are output and input disturbances)

$$U(s) = G_{PID}(s) \left[ G_r(s)Y_s(s) - Y(s) \right] + G_r(s)G^{-1}(s)Y_s(s)$$

which represents a two-degree of freedom PID controller. In the pure tracking context, i.e., without modeling errors and disturbances, the plant output is exactly equal to the reference trajectory and the PID is therefore not needed. In presence of modeling errors and/or disturbances, the PID makes the appropriate correction. Its tuning depends on $k_1$, $k_2$ and $k_3$ and can be seen as a pole-placement problem.

Indeed, the regulation dynamics is ($\mathcal{L}d = D(s)$)

$$\frac{Y(s)}{D(s)} = \frac{s^3 + a_1s^2 + a_0s}{F(s)}$$

Of course, the regulation poles will correspond to the poles of the error dynamics (31). For instance, to place all three poles at the real value $p$ ($p < 0$), possible values for the tuning parameters are

$$k_1 = -p + \sqrt{2}$$

$$k_2 = -p$$

$$k_3 = -p - \sqrt{2}$$

If it is desired to have the poles larger than $-\sqrt{2}$ (time units)$^{-1}$, the time scale must be changed to insure the positivity of $k_3$. 
The control law (37) could be obtained without using the backstepping algorithm. However, as will be shown in the next section, the backstepping technique allows one to make the above PID control adaptive, while being robust by taking into account the adaptation dynamics and using nonlinear damping terms.

3.2 Adaptive PID control

In the adaptive context, the introduction of the integral action is also exploited to design robust self-tuning PID controllers. In addition to the elimination of the steady-state errors, this modification turns out to enhance the convergence properties of the unknown parameter estimates. Nevertheless, sliding the added integrator from the plant equation to controller equation is not as obvious as it is in the non-adaptive case. The variable nature of the parameter estimates does not allow direct integration of the control expression. The assumptions of this section are the following: \( m = 0, \ n = 2 \), the plant parameters \( c = b_0^{-1}, a_0 \) and \( a_1 \) are unknown but constant, the sign of \( c \) is known and the estimate errors are

\[
\tilde{c} = \hat{c} - c
\]  
(46)

\[
\tilde{a}_0 = \hat{a}_0 - a_0
\]  
(47)

\[
\tilde{a}_1 = \hat{a}_1 - a_1
\]  
(48)

where \( \hat{c}, \hat{a}_0 \) and \( \hat{a}_1 \) are the parameter estimates. The parameterization from \( b_0 \) to \( c \) is introduced to avoid a division, in further developments, by \( \hat{b}_0 \) which could occasionally be equal to zero. Three new terms are added to the last clf to take
into account the adaptation mechanism

\[ V_3^* = V_3 + \frac{\dot{a}_0^2}{2\gamma_0} + \frac{\dot{a}_1^2}{2\gamma_1} + \frac{\dot{c}^2}{2\gamma_c|c|} \]

\[ = \frac{1}{2}\dot{e}_1^2 + \frac{1}{2}\dot{e}_2^2 + \frac{1}{2}\dot{e}_3^2 + \frac{\dot{a}_0^2}{2\gamma_0} + \frac{\dot{a}_1^2}{2\gamma_1} + \frac{\dot{c}^2}{2\gamma_c|c|} \]  

(49)

where \( \gamma_0, \gamma_1 \) and \( \gamma_c \) are positive adaptation gains. The clf derivative is expressed as

\[ \dot{V}_3^* = -k_1\dot{e}_1^2 - k_2\dot{e}_2^2 - k_3\dot{e}_3^2 
+ \frac{\dot{a}_0}{\gamma_0} \left[ \dot{a}_0 + \gamma_0\dot{e}_3\ddot{y} \right] 
+ \frac{\dot{a}_1}{\gamma_1} \left[ \dot{a}_1 + \gamma_1\dot{e}_3\ddot{y} \right] 
+ \frac{\dot{c}}{\gamma_c|c|} \left[ \dot{c} + \gamma_c\text{sgn}(c)\dot{e}_3w^* \right] \]

(50)

where the Laplace transform of the filtered control action \( w^* \) is given by

\[ W^*(s) = (s^3 + \dot{a}_1s^2 + \dot{a}_0s)Y_r(s) + \left[ (k_1 + k_2 + k_3 - \dot{a}_1)s^2 
+ (2 + k_1k_2 + k_2k_3 + k_1k_3 - \dot{a}_0)s + (k_1 + k_3 + k_1k_2k_3) \right]E(s) \]

(51)

This corresponds to the certainty-equivalence-based adaptive version of \( W(s) \). To obtain \( \dot{V}_3^* \leq 0 \), obvious update laws are

\[ \dot{\dot{a}}_0 = -\gamma_0\dot{e}_3\ddot{y} \]  

(52)

\[ \dot{\dot{a}}_1 = -\gamma_1\dot{e}_3\ddot{y} \]  

(53)

\[ \dot{\dot{c}} = -\gamma_c\dot{e}_3\text{sgn}(c)w^* \]  

(54)

Because the parameter estimates are not constant, sliding the integrator from the plant expression to the controller is not as easy as in the non-adaptive case. The result of this integration is given by

\[ u = \int \dot{u}dt = \int \dot{c}w^*dt = \dot{c} \int w^*dt \]  

\[ = \int \left[ \dot{c} \int w^*dt \right]dt \]  

(55)
This command can be written as follows

\[ u = u_{ce} + u_{dyn} \tag{56} \]

where \( u_{ce} \), the certainty-equivalence-based adaptive version of (37) is, in the s-plane,

\[ U_{ce}(s) = (\hat{T}_d s + \hat{T}_i + \hat{K}_c) E(s) + \hat{c}(s^2 + \hat{a}_1 s + \hat{a}_0) Y_r(s) \tag{57} \]

and \( u_{dyn} \) compensates for the adaptation dynamics

\[ u_{dyn} = \gamma_0 \hat{c} \int y \dot{y} \varepsilon_3 dt + \gamma_1 \hat{c} \int \ddot{y} y \varepsilon_3 dt + \gamma_c \text{sgn}(c) \int \left[w^* \int w^* dt\right] \varepsilon_3 dt \tag{58} \]

The adaptive tuning parameters are given by

\[ \hat{T}_d = \hat{c}(k_1 + k_2 + k_3 - \hat{a}_1) \tag{59} \]
\[ \hat{K}_c = \hat{c}(2 + k_1 k_2 + k_1 k_3 + k_2 k_3 - \hat{a}_0) \tag{60} \]
\[ \hat{T}_i = \hat{c}(k_1 + k_3 + k_1 k_2 k_3) \tag{61} \]

### 3.3 Nonlinear damping

The enhancement of the convergence properties, obtained by the new adaptive PID, can be further increased. Indeed, the introduction of nonlinear damping terms \cite{1} in the controller expression guarantees a more robust control that allows speeding up the adaptation without losing the loop stability. With this nonlinear damping, the errors remain bounded even when the adaptation is turned off. To introduce the damping terms in the above-obtained controller, a new filtered control is defined

\[ v = w^* - \varepsilon_3 (m_1 \dot{y}^2 + m_2 \ddot{y}^2) \tag{62} \]
The derivative of the actual control is also augmented with a nonlinear damping term associated to the estimation error \( \tilde{c} \)

\[
\dot{u} = \dot{c}v - \text{sgn}(c)m_3\varepsilon_3v^2
\]  

(63)

All the parameters \( m_1, m_2 \) and \( m_3 \) are positive. Using the above definitions and the update laws

\[
\dot{\hat{a}}_0 = -\gamma_0 \varepsilon_3 \dot{y}
\]

(64)

\[
\dot{\hat{a}}_1 = -\gamma_1 \varepsilon_3 \ddot{y}
\]

(65)

\[
\dot{\hat{c}} = -\gamma_c \varepsilon_3 \text{sgn}(c)v
\]

(66)

leads to the following derivative of the clf

\[
\dot{V}_3^* = -k_1\varepsilon_1^2 - k_2\varepsilon_2^2 - k_3\varepsilon_3^2 - m_1(\varepsilon_3 \dot{y})^2 - m_2(\varepsilon_3 \ddot{y})^2 - \frac{m_3}{|c|}(\varepsilon_3 v)^2
\]

\[
\leq 0
\]

(67)

In the absence of the adaptation, i.e., when \( \gamma_0 = \gamma_1 = \gamma_c = 0 \), (67) verifies the inequality

\[
\dot{V}_3^* \leq -k_0|\varepsilon|^2 + \frac{1}{4m_0}|	ilde{\theta}|^2
\]

(68)

where

\[
\tilde{\theta} = \begin{bmatrix} \hat{a}_0 & \hat{a}_1 & \hat{c} \end{bmatrix}^T
\]

(69)

\[
m_0 = \min\{m_1, m_2, m_3|c|\}
\]

(70)

\[
k_0 = \min\{k_1, k_2, k_3\}
\]

(71)
The inequality (68) shows clearly that $\dot{\mathcal{V}}_3^*$ is negative for

$$|\varepsilon| \geq \frac{|\tilde{\theta}|}{2} \sqrt{\frac{1}{k_0 m_0}}$$  \hspace{1cm} (72)

which means that $\mathcal{V}_3^*$ is bounded, and so is $\varepsilon$, for any parameter estimation errors. The bound on the error vector is given by

$$|\varepsilon| \leq \frac{|\tilde{\theta}|}{2} \sqrt{\frac{1}{k_0 m_0}}$$  \hspace{1cm} (73)

The resulting control law is now expressed as

$$u = u_{ce} + u_{dyn} + u_{nld}$$  \hspace{1cm} (74)

where the certainty-equivalence part $u_{ce}$ is identical to the previous case (57), while the part that compensates for the adaptation dynamics is, this time, expressed as

$$u_{dyn} = \gamma_0 \dot{\hat{c}} \int y \dot{y} \varepsilon_3 dt + \gamma_1 \dot{\hat{c}} \int \dot{y} \ddot{y} \varepsilon_3 dt + \gamma_c \text{sgn}(c) \int \left[ \varepsilon_3 v \int v dt \right] dt$$  \hspace{1cm} (75)

The contribution of the nonlinear damping terms is given by

$$u_{nld} = -\int \left[ \dot{\hat{c}} \left( m_1 \dot{y}^2 + m_2 \ddot{y}^2 \right) + m_3 \text{sgn}(c) v^2 \right] \varepsilon_3 dt$$  \hspace{1cm} (76)

4 Simulation example

A comparison between a certainty-equivalence PID that uses the least-square algorithm to update the parameters and the adaptive backstepping PID will be made.
The plant is unstable and is defined by the following parameters

\[ a_0 = -2 \quad (77) \]
\[ a_1 = -1 \quad (78) \]
\[ c = 0.5 \quad (79) \]

For both controllers, the initial parameter values are set to one-quarter of their true values, the reference model is

\[ G_r(s) = \frac{1}{(1 + s)^3} \quad (80) \]

and the plant output is filtered with

\[ G_f(s) = \frac{1}{(1 + 0.07s)^2} \quad (81) \]

Both PID controllers have the same tuning

\[ k_1 = 2.8428 \quad (82) \]
\[ k_2 = 1.4286 \quad (83) \]
\[ k_3 = 0.0144 \quad (84) \]
which correspond to placing all three regulation poles at $-1/0.7$. Other tuning parameters for the backstepping PID are

$$\gamma_0 = k_1$$

(85)

$$\gamma_1 = 3 \times k_2$$

(86)

$$\gamma_c = 4 \times k_3$$

(87)

$$m_1 = k_1$$

(88)

$$m_2 = k_2$$

(89)

$$m_3 = k_3$$

(90)

The initial covariance matrix of the least-square algorithm (for the certainty-equivalence PID) is $P(0) = 5 \times I$, where $I$ is the identity matrix.

Figure 2 shows the step setpoints and the input disturbance applied to both control systems. The plant outputs and manipulated variables are depicted in Figure 3. Figure 4 shows the parameters estimates. The loops are closed at $t = 0$. Better transients, with a smoother manipulated variable and better parameter convergence, are achieved with the backstepping PID.

If the backstepping system was simulated for a longer time after the second step setpoint with $d_i$ as the only excitation, it would become unstable at time $t \approx 528$ because of the parameter drift. This could be avoided simply by freezing the adap-
tation mechanism \(i.e.,\) by selecting \(\gamma_0 = \gamma_1 = \gamma_c = 0\) when the performance seems satisfactory. Safely setting the adaptation off is rendered possible by the use of the nonlinear damping terms. The parameter projection [5] and the switching \(\sigma\)-modification [6] are two other possible techniques to avoid the parameter drift. The switching \(\sigma\)-modification would also allow the control of slowly time-varying plants [7]. The parameter drift is more pronounced with the least-squares algorithm (Figure 4).

The plant being unstable, the adaptation gains of the certainty equivalence PID must be relatively small to avoid the instability. Therefore, the parameters converge slowly (Figure 4) and the response to the first reference trajectory change (Figure 3) is quite unsatisfactory. Since the backstepping PID is inherently more robust, especially with the introduction of the nonlinear damping terms, the adaptation gains can be made larger, thus leading to a faster parameter convergence and much better transients. Indeed, even at the first reference trajectory change, the plant output follows quite closely the reference trajectory. With \(P(0) = 5 \times I\), the least-squares system start oscillating at time \(t \approx 33\), if the setpoint remains constant. A slightly larger initial value \(P(0)\) would not improve very significantly the transients, at the cost of a faster parameter drift (larger increases in \(P(0)\) would mean instability).
5 Conclusion

Certainty equivalence-based PID are still the most used adaptive structures. However, during the control design, the adaptation dynamics are not taken into account. To overcome this drawback, the adaptive backstepping technique in which, for practical considerations, an integral action is added, is used to design more robust PID controllers. To gain even more robustness, nonlinear damping terms are added to guarantee bounded errors, even when the adaptation is turned off.

References


List of captions to illustrations

- Figure 1: Two-degree of freedom PID
- Figure 2: Setpoint and input disturbance
- Figure 3: Plant output and input
- Figure 4: Parameter estimates
Figure 1: Two-degree of freedom PID
Figure 2: Setpoint and input disturbance
Figure 3: Plant output and input
Figure 4: Parameter estimates