Proceedings of the 1999 IEEE
International Symposium on Computer Aided Control System Design
Kohala Coast-Island of Hawai'i, Hawai'i, USA • August 22-27, 1999

Automatic PID Tuning: An Application of Unfalsified Control *

Myungsoo Jun and Michael G. Safonov†
Dept. of Electrical Engineering — Systems
University of Southern California
Los Angeles, CA 90089-2563, USA

Key Words: Adaptive control, self-tuning control, unfalsified control, PID control, controller identification.

Abstract

In this paper, we give detailed procedures for using unfalsified control theory for real-time PID controller parameter tuning and adaptation. Related to the candidate-elimination algorithms of machine learning, our PID tuning technique does not need a plant model and makes PID gain selection possible by just using observed data. Simulation results are included.

1. INTRODUCTION

Traditional controller synthesis theories contain many assumptions about the plant. However, some assumptions which are made to simplify modeling procedure restrict applicability and some are so unrealistic that they make designs based on those assumptions unreliable. Modeling techniques such as [3, 8] were proposed to improve traditional modeling methods. However, these new modeling techniques still have some assumptions about the plant.

With *unfalsified control theory* [6, 7], you can design a controller that is consistent with a performance objective and measured past data without plant models or assumptions on the plant. The theory works by eliminating or pruning hypotheses that are not consistent with evolving experimental data. The hypotheses in question assert that members of a class of candidate controllers can meet prescribed closed-loop performance goals. The theory is related to certain artificial intelligence concepts, such as *list-then-eliminate* and *candidate-elimination* algorithms of machine learning theory [4]. In this paper, we use the unfalsified control theory to adaptively tune PID controller gains.

There are several methods for tuning PID gains. One of the most widely used methods for tuning PID gains manually is the Ziegler-Nichols method [9]. Åström *et al.* [1] proposed a procedure for automatic tuning of regulators of the PID type to specifications on phase and amplitude margin. Nishikawa *et al.* [5] proposed a method for determination of the control parameters based on the minimization of quadratic performance objectives in the time domain. However, it may be difficult to apply these methods for the automatic tuning of PID parameters to more complex systems because the rules are based, either implicitly or explicitly, on identifying approximate plant models. In this paper we describe a method based on unfalsified control theory for tuning PID parameters adaptively based on input-output data only, that is, without a plant model.

2. UNFALSIFIED CONTROL THEORY

Further details about unfalsified control theory are described in [6] and [7]. The formal definition of *unfalsification* and *falsification* is as follows:

Definition 1 [6] *A controller is said to be* falsified *by measurement information if this information is sufficient to deduce that the performance specification* $(r, y, u) \in T_{spec}$ $\forall r \in \mathcal{R}$ *would be violated if that controller were in the feedback loop. Otherwise, the controller is said to be* unfalsified.

With the above definition of *unfalsification* and *falsification* we can state the following theorem in order to solve unfalsified control problem. Let the symbol K denote the set of triples (r, y, u) that satisfy the equations that define the behavior of controller. Denote by P_{data} the set of triples (r, y, u) consistent with past measurements of (u, y) – cf. [6].

Theorem 1 [6] *A control law K is unfalsified by measurement information set P_{data} if, and only if, for each triple $(r_0, y_0, u_0) \in P_{data} \cap K$, there exists at least one pair (u_1, y_1) such that*

$$(r_0, y_1, u_1) \in P_{data} \cap K \cap T_{spec} \tag{1}$$

Fictitious reference signals occupy an important position in unfalsified control theory. Given measurements of plant input-output signals u, y, there may correspond for each candidate controller, say K_i, one or more fictitious reference signals $\tilde{r}_i(t)$. The $\tilde{r}_i$'s are hypothetical signals that would have exactly reproduced the measured

*Research supported in part by AFOSR Grant F49620-98-1-0026.
†Corresponding authors; email mjun@bode.usc.edu, msafonov@usc.edu; fax +1-213-740-4449.

0-7803-5500-8/99 $10.00 © 1999 IEEE

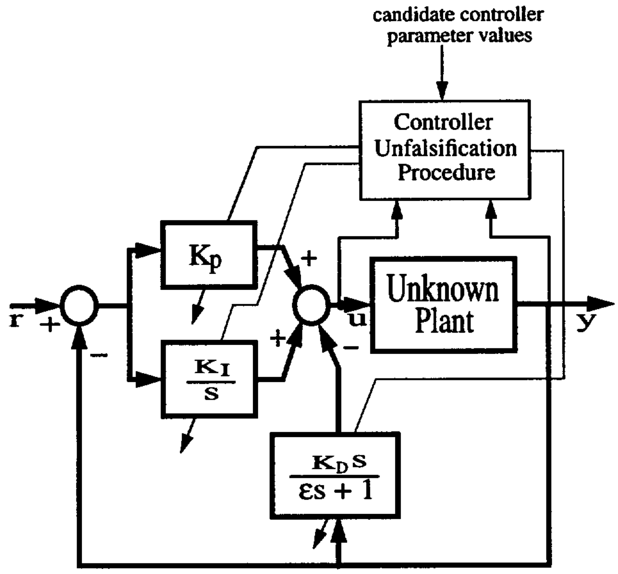

Figure 1: PID controller configuration with approximated derivative term

data (u, y) if the candidate controller K_i had been in the feedback loop during the *entire* time period over which the measured data (u, y) was collected. Because the data (u, y) may have been collected with a controller other than K_i in the feedback loop, the fictitious reference signal $\tilde{r}_i$ is in general not the same as the actual reference signal $r(t)$. A candidate controller K_i is called *causally-left-invertible* if a unique values for its fictitious reference signal $\tilde{r}_i(t)$ is determined by past values of the open-loop data $u(t)$ and $y(t)$. Further details about fictitious reference signals can be found in [7].

3. PID CONTROLLER

PID control is used commonly in industrial and aerospace applications. It is its simplicity and performance characteristics that make PID popular. The ideal PID controller can be expressed as $u = (k_P + k_I/s)(r-y) - sk_D y$, where k_P, k_I and k_D are non-negative real numbers called the proportional gain, integral gain, and derivative gain, respectively. The integral part makes steady-state tracking of step commands robust and the proportional and derivative part affect stability and transient behavior.

The ideal PID controller has an improper transfer function. It is hard to exactly implement the derivative part. Thus, an approximated derivative $\frac{sk_D}{\varepsilon s+1}$ is used in realization where ε is a small number. A PID controller

$$u = (k_P + \frac{k_I}{s})(r - y) - \frac{sk_D}{\varepsilon s + 1} y \qquad (2)$$

with approximated derivative term is shown in Fig. 1.

Standard PID controllers have gains $k_P > 0$, $k_I \geq 0$, $k_D \geq 0$. They are always *causally left invertible*, which

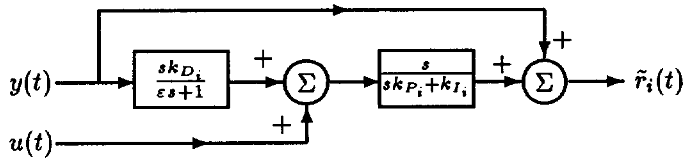

Figure 2: Generating the i-th fictitious reference signal $\tilde{r}_i(t)$.

means that, given past values of $u(t)$ and $y(t)$, there is a unique fictitious reference signal $\tilde{r}_i(t)$ associated with each controller K_i. The signal $\tilde{r}_i(t)$ can be reliably computed in real-time by filtering the measurement data (u, y) via the following expression (obtained by rearranging Eq. (2) with appropriate substitutions):

$$\tilde{r}_i = y + \frac{s}{sk_{P_i} + k_{I_i}} \left(u + \frac{sk_{D_i}}{\varepsilon s + 1} y \right). \qquad (3)$$

A bank of such filters (one for each $i \in \mathbf{I}$) may be used to generate the fictitious reference signals $\tilde{r}_i(t)$ in real-time — see Fig. 2. One filter is required for each candidate controller K_i (i.e., for each distinct triple of candidate PID gains $(k_{P_i}, k_{I_i}, k_{D_i})$).

We denote by $[\tilde{A}_i, \tilde{B}_i, \tilde{C}_i, \tilde{D}_i]$ a state-space realization of filter associated with the i-th candidate controller transfer function K_i, $(i \in \mathbf{I})$. That is, $[\tilde{A}_i, \tilde{B}_i, \tilde{C}_i, \tilde{D}_i]$ is a state-space realization of the system in Fig. 2 with the values of $(k_{P_i}, k_{I_i}, k_{D_i})$ associated with the i-th controller K_i inserted. The state vector is represented by $\tilde{x}_i(t)$.

4. CONTROLLER PARAMETER ADAPTATION

The main difference between unfalsified control and other adaptive methods is that one can adjust controller parameters in unfalsified control based on measured data alone without any assumptions about the plant. Our algorithm for tuning PID gains uses only measured past data in adapting its gains. While, in principle, the unfalsified control theory allows for the set $\mathbf{K}$ to be an arbitrary subset of R^n where n is the number of controller parameters to be adjusted, we discretize candidate controller set $\mathbf{K}$ so that it has only a finite number of elements in order to simplify computations.

At each time τ, the performance specification set T_{spec} consists of the set of triples (r, y, u) satisfying an integral performance inequality of the form

$$J(t) \triangleq -\rho + \int_0^t T_{spec}(r(t), y(t), u(t))dt \leq 0, \ \forall t \in [0, \tau] \qquad (4)$$

where $\rho \geq 0$ and $T_{spec}(\cdot,\cdot,\cdot)$ are chosen by the designer.

By Theorem 1, the i-th candidate PID controller K_i is unfalsified at time τ by plant data $u(t), y(t)$, $(t \in [0, \tau])$ if, and only if,

$$\tilde{J}(i, t) \leq 0, \ \forall t \in [0, \tau]$$

where

$$\tilde{J}(i,t) \triangleq -\rho + \int_0^t T_{spec}(\tilde{r}_i(t), y(t), u(t))dt, \quad \forall t \in [0,\tau], \tag{5}$$

$u(t), y(t)$, $(t \in [0,\tau])$ is measured past plant data, and $\tilde{r}_i(t)$ denotes the fictitious reference signal for the i-th controller K_i — see Fig. 2.

Discretizing time, we may recursively compute each of the fictitious reference signals $\tilde{r}_i(k\Delta t)$ and its corresponding cost $\tilde{J}(i, k\Delta t)$ at each time $\tau = k\Delta t$. We use MATLAB function `c2d.m` to discretize the fictitious reference signal generating systems depicted in Fig. 2. Eq. (5) is discretized as

$$\begin{aligned}
&\tilde{J}(i,k\Delta t)\\
&= \tilde{J}(i,(k-1)\Delta t) + \\
&\quad \int_{(k-1)\Delta t}^{k\Delta t} T_{spec}(\tilde{r}_i(t), y(t), u(t))dt \qquad (6)\\
&\approx \tilde{J}(i,(k-1)\Delta t) + \\
&\quad \frac{1}{2}\Delta t \cdot \{T_{spec}(\tilde{r}_i(k\Delta t), y(k\Delta t), u(k\Delta t)) + \\
&\quad T_{spec}(\tilde{r}_i((k-1)\Delta t), y((k-1)\Delta t), u((k-1)\Delta t))\} \qquad (7)
\end{aligned}$$

when $\rho = 0$.

The adaptation algorithm is as follows and detailed calculation procedure is described in Section 5.

Algorithm 1 (Controller Unfalsification Procedure)
INITIAL SETTING:

- *a finite set* $\mathbf{K}$ *of* m *controller candidates* K_i, $i \in \mathbf{I} \triangleq \{1, \cdots, m\}$.
- *performance functional* $T_{spec}(\cdot,\cdot,\cdot)$
- *sampling time* Δt.
- *the values of* ε, ρ *and* σ *(see Eq.(9).*
- *initial time* $k = 0$.
- *initial consistency criterion* $\tilde{J}(i,0) = 0$, $i = 1, \cdots, m$.
- *initial controller* K_m.

PROCEDURE (at each time $\tau = k\Delta t$*):*

1. *Measure* $u(k\Delta t)$ *and* $y(k\Delta t)$.
2. *For each* $i \in \mathbf{I}$,
   a) *calculate* $\tilde{x}_i((k+1)\Delta t)$ *and* $\tilde{r}_i(k\Delta t)$ *using a MATLAB* `c2d.m` *discretized approximation of Fig. 2,*
   b) *calculate* $\tilde{J}(i, k\Delta t)$ *using Eq. (7), and*
   c) *if* $\tilde{J}(i, k\Delta t) > 0$, *then delete the controller index element* i *from* $\mathbf{I}$ *(since* K_i *has been falsified by the measured data up to time* $k\Delta t$*); else continue.*
3. *If the set* $\mathbf{I}$ *is empty, then terminate algorithm; else, set the current controller to* $K_{\hat{i}(k)}$, $\hat{i}(k) = \arg\min\{\tilde{J}(i, k\Delta t),\ i \in \mathbf{I}\}$ *and increment time* $(k \leftarrow k+1)$, *go to step 1 and repeat.*

If the set $\mathbf{I}$ becomes the empty set, the algorithm terminates because all controllers in $\mathbf{K}$ are falsified. In this case we have to either relax the performance specification or augment the set $\mathbf{K}$ with additional controller candidates.

In general, many candidate controllers will be falsified and discarded even before they are ever inserted into the feedback loop. Consequently, the algorithm often converges quite quickly. The current controller $K_{\hat{i}(k\Delta t)}$ remains in the loop so long as it remains unfalsified by the past data. If at some time $k\Delta t$ the current controller becomes falsified by new data $(u(k\Delta t), y(k\Delta t))$, then the algorithm switches to a new controller $K_{\hat{i}}$ which we chose to be the one that has the largest index $\hat{i}(k\Delta t)$ among the as yet unfalsified controller candidates K_i in $\mathbf{K}$.

5. SIMULATION

In this section we describe a simulation of PID controller parameter adaptation using unfalsified control theory. The simulation shown was conducted with no disturbance, no noise and zero-initial conditions, though this is not essential.

The performance specification set T_{spec} is taken to be the set of $(r, y, u) \in L_{2e} \times L_{2e} \times L_{2e}$, which for all $\tau \geq 0$ satisfy the inequality

$$\|w_1 * (r-y)\|_\tau^2 + \|w_2 * u\|_\tau^2 - \sigma^2\tau \leq \|r\|_\tau^2 + \rho \tag{8}$$

where $\|f\|_\tau^2 = \int_0^\tau |f(t)|^2 dt$, and $*$ denotes the convolution operator. Design parameters are σ (a constant representing the r.m.s. effects of noise on the cost), and the signals w_1 and w_2 are weighting filters. Therefore, $T_{spec}(r(t), y(t), u(t))$ is

$$\begin{aligned}
&T_{spec}(r(t), y(t), u(t)) = \\
&\quad |w_1 * (r(t) - y(t))|^2 + |w_2 * u(t)|^2 - \sigma^2 - |r(t)|^2 \qquad (9)
\end{aligned}$$

The simulation is conducted as follows: At each sampling time $\tau = k\Delta t$, the data $u(k\Delta t)$ and $y(k\Delta t)$ are measured. Then, the controller unfalsification procedure (Algorithm 1) is invoked to determine which, if any, previously unfalsified controllers are now falsified based on the consistency test

$$\tilde{J}(i, k\Delta t) \leq 0. \tag{10}$$

The consistency criterion $\tilde{J}(i, k\Delta t)$ is computed based on the discrete-time approximation (7) and MATLAB `c2d.m`

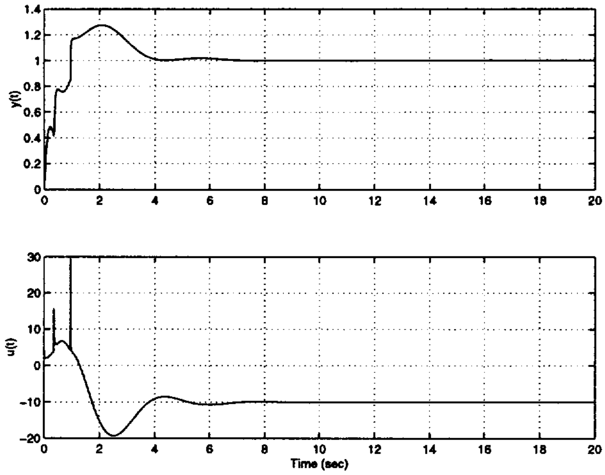

Figure 3: Plots of signals $y(t)$ and $u(t)$ when the states of the controller are not reset at switching time. Poor transients with spikes can occur if we fail to properly reset controller states at switching time.

discrete-time approximations of the $\tilde{r}_i(t)$ system of Fig. 2 and of the filters w_1 and w_2.

When the current controller is among those falsified by the most recent data, the algorithm switches to a new controller. At each such switching time, the control algorithm resets the states of the integrator term $(k_{P_i} + \frac{k_{I_i}}{s})$ and the approximate differentiator term $\frac{k_{D_i} s}{\varepsilon s+1}$, thereby preventing any discontinuity in either of their respective output signals, say $u_{PI}(t)$ and $u_D(t)$. This assures that the control signal $u(t) = u_{PI}(t) - u_D(t)$ is smooth, avoiding abrupt changes or high peaks that might otherwise result from switches in (k_P, k_I) or k_D, respectively. If we do not reset the states of the controller at switching time but maintain them as were before switching, we can see undesirable high peaks in the signal $u(t)$ and higher overshoot in the signal $y(t)$ from the Fig. 3. Therefore, it is important to reset controller states at switching time in such a way as to prevent this.

The following were used in the simulation:

- unknown plant $P(s) = \frac{2s^2+2s+10}{(s-1)(s^2+2s+100)}$
- $W_1(s) = \frac{s+10}{2(s+0.1)}$, $W_2(s) = \frac{0.01}{1.2(s+1)^3}$
- step reference signal $r(t) = 1, \ \forall t \geq 0$
- all initial conditions at time 0 are zero
- sampling time Δt is 0.05 second
- the value of $\varepsilon = 0.01$
- no noise ($\sigma = 0$) and zero initial conditions ($\rho = 0$).
- $\mathbf{K}_D = \{0.6, 0.5\}$, $\mathbf{K}_P = \{5, 10, 25, 80, 110\}$, $\mathbf{K}_I = \{2, 50, 100\}$. Thus, the number of candidate controllers in $\mathbf{K}$ is 30.

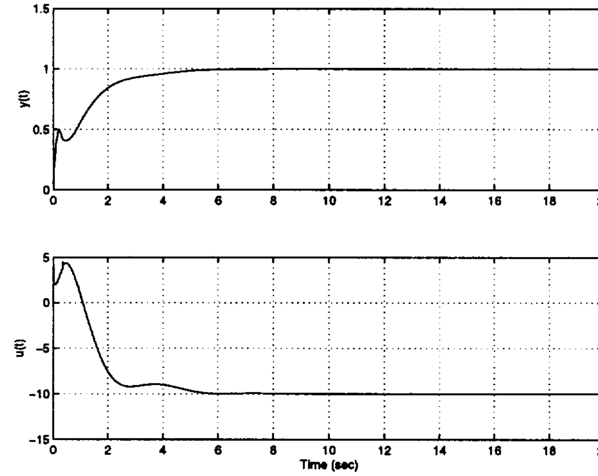

Figure 4: Simulation results showing good transient response with correctly reset controller states

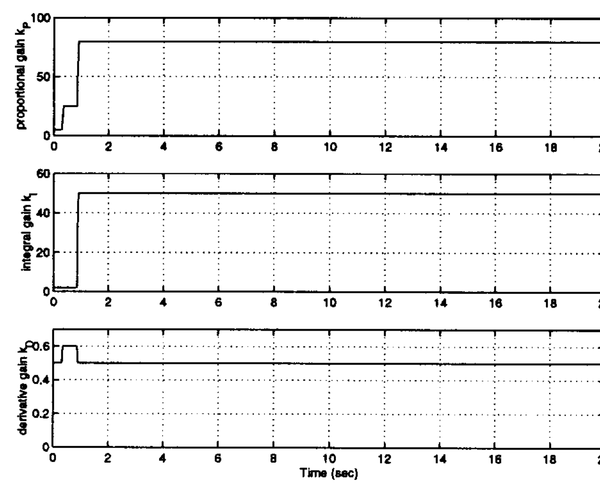

Figure 5: Simulation results showing the changes in controller gains

The simulation was carried out using Simulink. The results are as shown in Fig. 4 and Fig. 5, and the Simulink model used in simulation is in Fig. 6. The figure shows two times at which gain switching occurs. The values of k_P and K_I switch at the first switch time, and at the second switch all three gains k_P, k_I and k_D change. At each switching time, the current controller is falsified and a new, as yet unfalsified controller is switched into the close loop. The final values of the controller parameters are $k_P = 80$, $k_I = 50$ and $k_D = 0.5$. The final number of unfalsified elements of the set $\mathbf{K}$ is 12.

6. DISCUSSION

While the simulation shown in Fig. 4 was conducted assuming no noise ($\sigma = 0$) and zero initial conditions ($\rho = 0$), the algorithm is actually fairly robust to noise and initial state perturbations. However, if the noise or

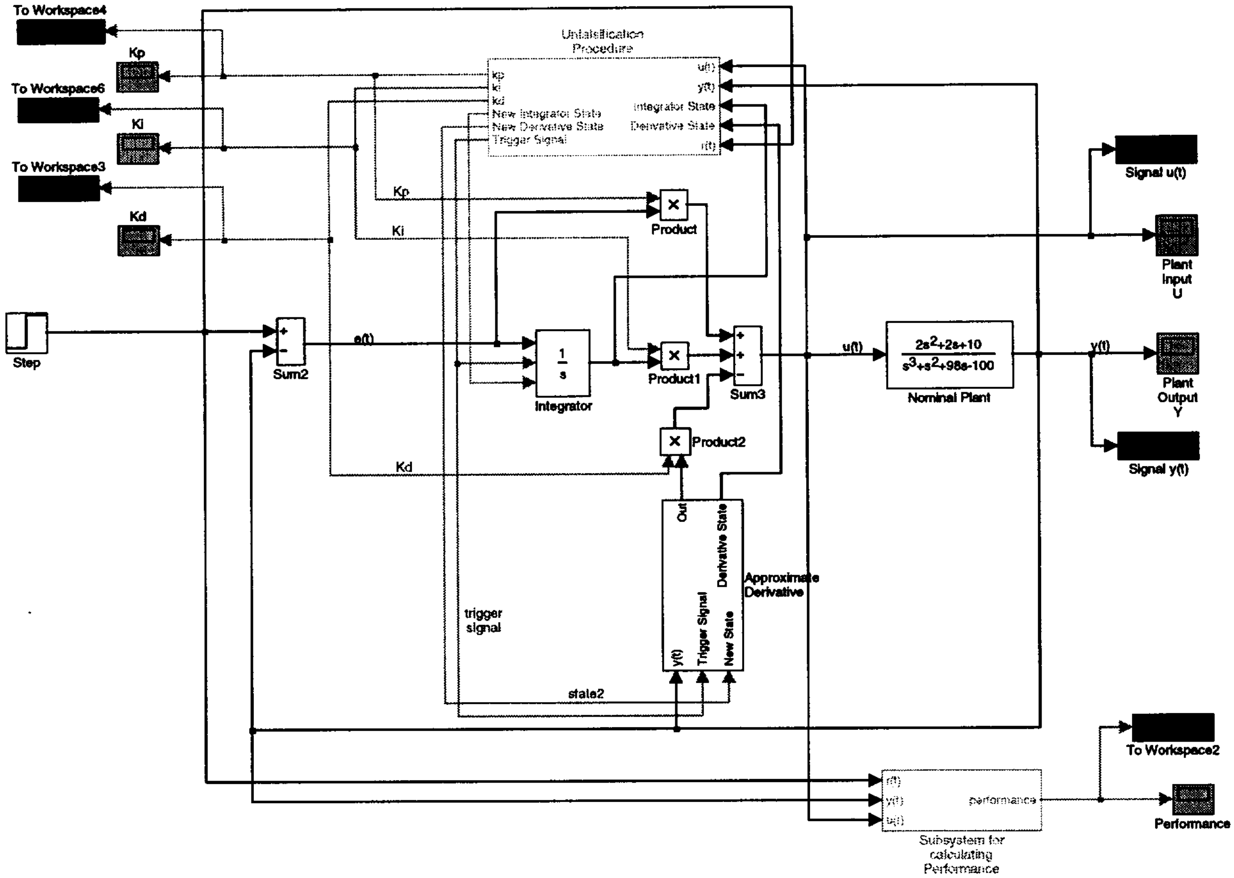

Figure 6: Simulink model used in simulation

initial conditions are very large, then it may sometimes be necessary to use non-zero values for ρ and/or σ in the performance specification (8).

If a plant is slowly time-varying or subject to occasional abrupt changes, the far past data may not contain much information about current plant. In such cases, either an exponential forgetting factor or a finite-memory data-window should be introduced.

While the simulation only shows the result for the case in which a step command $r(t)$ is the input, the algorithm also works when for inputs other than a step signal. It is important only that the input have sufficient strength and spectral breadth to allow candidate controllers K_i to be reliably ordered by the performance specification functional $\tilde{J}(i,t)$.

7. CONCLUSION

In this paper, we described in detail how to adaptively tune the parameters of a PID controller in real-time using unfalsified control theory. An advantage of this approach is that no plant model is required. We need only real-time measurements of input-output data (u, y) from the plant. Thus we may apply this method to distributed parameter systems, to nonlinear time-varying plants as well as to high-order linear time-invariant plants.

Irrespective of how complicated the plant may be, the PID controllers themselves are not very complicated and we can easily compute the unfalsified controller parameters at each time via the recursive procedure described in Section 4. A limitation of our procedure is that the set K of unfalsified controllers may shrink to a null set if there are no PID controllers in K that are capable of meeting the performance specification (4). But when this is not the case, convergence of the algorithm is typically rapid and sure-footed. Our experience with simulations has been that convergence is usually so rapid that satisfactory transient response is obtained on the first try even with no prior knowledge of the plant.

References

[1] Åström, K. J., and T. Hägglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins," *Automatica*, vol.20, pp.645-651, 1984.

[2] Franklin, G. F., D. J. Powell and M. L. Workman,

Digital Control of Dynamic Systems, 3rd ed, Reading, MA: Addison-Wesley, 1997.

[3] Kosut, R. L., M. K. Lau and S. P. Boyd, "Identification of systems with parametric and nonparametric uncertainty," *Proc. 1990 American Control Conf.*, San Diego, CA, May 1990.

[4] Mitchell, T. M., *Machine Learning*, New York, NY: McGraw-Hill, 1997.

[5] Nishikawa, Y., N. Sannomiya, T. Ohta and H. Tanaka, "A methods for auto-tuning of PID control parameters," *Automatica*, vol.20, pp.321-332, 1984.

[6] Safonov, M. G., and T. C. Tsao, "The unfalsified control concept and learning," *IEEE Trans. Automat. Contr.*, vol. 42, no. 6., pp. 843-847, Jun. 1997.

[7] Tsao, T. C., "Set theoretic adaptor systems," Ph.D. dissertation, Univ. of Southern California, May 1994.

[8] Younce, R. C., and C. E. Rohrs, "Identification with nonparametric uncertainty," *Proc. IEEE Conf. on Decision and Control*, Honolulu, HI, Dec. 1990.

[9] Ziegler, J. G., and N. B. Nichols, "Optimum setting for automatic controllers," *Trans. ASME*, vol.64, pp.759-768, 1942.