

NASA  
Technical Memorandum 103101

AVSCOM  
Technical Report 90-C-007

# An Expert System to Perform On-Line Controller Tuning

Jonathan S. Litt  
*Propulsion Directorate*  
*U.S. Army Aviation Research and Technology Activity—AVSCOM*  
*Lewis Research Center*  
*Cleveland, Ohio*

— (NASA-TM-103101) AN EXPERT SYSTEM TO  
— PERFORM ON-LINE CONTROLLER TUNING (NASA)  
15 p CSCL 09B

N90-23991

Unclass  
G3/63 0281592

Prepared for the  
1990 American Control Conference  
sponsored by the American Automatic Control Council  
San Diego, California, May 23-25, 1990

**NASA**





# AN EXPERT SYSTEM TO PERFORM ON-LINE CONTROLLER TUNING

Jonathan Litt

Propulsion Directorate  
U.S. Army Aviation Research and Technology Activity-AVSCOM  
Lewis Research Center  
Cleveland, OH 44135

## ABSTRACT

This paper describes an expert system which tunes a Proportional-Integral-Derivative (PID) controller on-line for a single-input-single-output multiple-lag process with dead time.

The expert system examines features of the previous transient responses and their corresponding sets of controller parameters. It determines a new set of controller gains to obtain a more desirable time response. This technique can be used to determine and implement a different set of PID gains for each operating regime and, once in steady state, the system can be used to find optimal parameters for load disturbance rejection.

The expert system can be applied to any system of the specified form (aerospace, industrial, etc.) and can be expanded to include additional process models.

## INTRODUCTION

Proportional-Integral-Derivative, Three-Mode, or Three-Term Controllers are an extremely popular form of controller. PID controllers or controllers made up of a subset of the terms (P, PI, PD) are used in most industrial (classical) applications [1] and, in some form, in many multivariable (modern) applications (for instance, see [2] or [3]).

The PID controller has no standard form and it can be represented in several ways [4]. For this work, a form common in industrial applications,

$$G_c(s) = K_c \left( \frac{1}{T_I s} + 1 + T_D s \right) \quad (1)$$

was used. In (1),  $K_c$  represents the proportional term,  $T_I$  is the reset time (reciprocal of the integral), and  $T_D$  is the rate time or derivative term.

In many circumstances, the ability to tune a controller to meet closed loop time domain specifications is considered more of an art than a science. In situations where the controlled system is hard to model accurately or is unmodeled as in many industrial settings, the tuning might be done by someone with an intuitive feel for the process and not necessarily a theoretical understanding of it. It is common for each plant operator to have his own preference for the best type of transient response and

operators will often change each other's controller settings from shift to shift. In such situations, one speaks of the optimal response as the transient which looks the best to the operator. The operators consider such features as rise time and overshoot to be the determining factors in an acceptable response. This is strictly a time domain approach. In industrial applications, the dissatisfaction with controller performance is so high among plant operators that over 50% of process control loops are run in manual (open loop) mode [5]. In modern control applications, many of which are optimal designs using frequency domain techniques, tuning is still a concern even though it is often described as an iterative design approach (for example, see [6] or [7]). There are still cases, however, where the classical terminology is applied to modern problems (see for instance [8] or [9]).

The work described in this paper was inspired by the tuning maps of Doris Wills [10]. Wills' original work addressed the tuning of a PID controller for a process described by

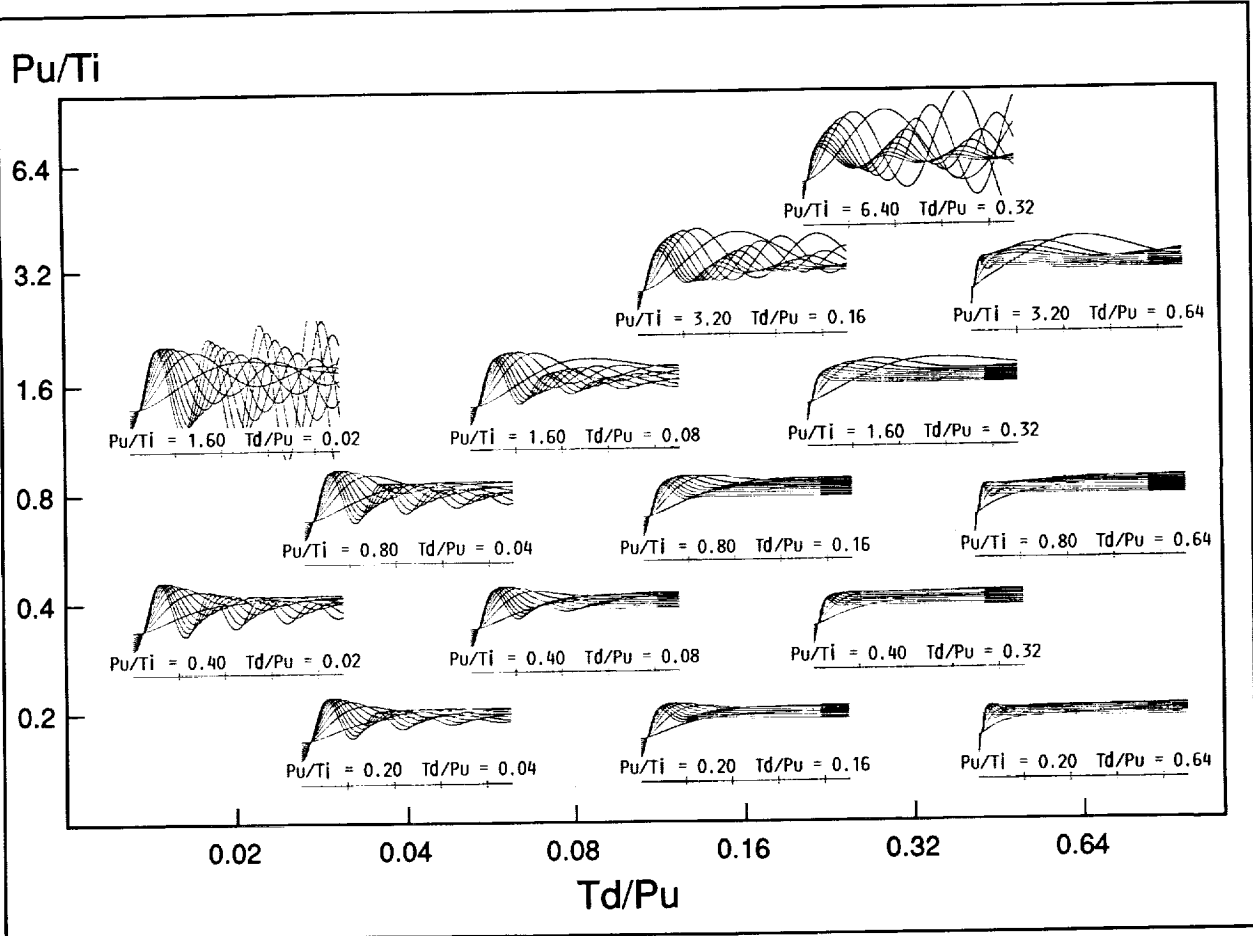
$$H(s) = \frac{e^{-Ts}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (2)$$

with  $\tau_1 = \tau_2 = 10T$ . A process with dead time using the appropriate proportional-only control will respond to a step input with a sustained oscillation [11]. The period of this oscillation is known as the Ultimate Period,  $P_U$ , and Wills used it as a normalizing factor in her tuning maps. A desired transient response was chosen for each map based on such features as quarter-decay response [12], critical damping, or minimum Integral Time Absolute Error (ITAE) [13]. For given values of  $T_I$  and  $T_D$ , the transient of the desired form (quarter-decay response, etc.) along with the value of  $K_C$  which achieved it were plotted in the appropriate region of the  $T_D/P_U - P_U/T_I$  plane. The normalization of the axes by  $P_U$  made them dimensionless and therefore general to a class of systems of the type in (2). Given a plant which may be approximated by (2), and its experimentally-determined ultimate period, the PID parameters for the desired response can be selected directly off the appropriate map.

## THE EXPERT SYSTEM

In this work the tuning map idea was extended from specified response curves plotted in the  $T_D/P_U - P_U/T_I$  plane to response surfaces plotted in the  $T_D/P_U - P_U/T_I - K_C$  space (figure 1). Each set of transients in the figure consists of ten responses with  $T_I$  and  $T_D$  fixed while  $K_C$  varies from 1.0 to 19.0 in increments of 2.0. The transient sets are displayed obliquely with  $K_C$  increasing from top to bottom. The horizontal axis for each set is marked in intervals of  $P_U$  seconds. The sets of responses are positioned appropriately in the  $T_D/P_U - P_U/T_I$  plane, presented on a log-log scale.

Tuning rules were extracted directly from this three-dimensional map. These rules, which are used with a forward-chaining inference engine, tune the PID controller



logaxes.drw

Figure 1. Three-dimensional tuning map.

parameters based on the important features of the response. The expert system looks at the following features: 1) percent overshoot, 2) ratio of the overshoot of the second peak to that of the first, 3) period of oscillation, 4) rise time, and 5) damping. The features' names are self-explanatory except for damping which has a nonstandard definition of the form

$$\text{damping} = \frac{P_2 - V_1}{P_1 - V_1}$$

where  $P_1$  is the overshoot of the first peak of the step response,  $P_2$  is the overshoot of the second peak, and  $V_1$  is the undershoot of the first valley. The damping is larger if the response is highly oscillatory and smaller if the response settles quickly. This is the opposite of the usual definition but similar to the definitions used in [14] and [15]. The process model used here is of the form in (2) with  $\tau_1 = \tau_2 = 10$  and  $T = 1$ .

Looking at the map in figure 1, one can infer general rules about how the changes in the controller parameters affect the features of the response. The observations are as follows:

1. Increasing  $K_C$  decreases period and vice versa.
2. Increasing  $K_C$  increases overshoot and vice versa.
3. Increasing  $K_C$  decreases rise time and vice versa.
4. Increasing  $K_C$  increases damping and vice versa.
5. Decreasing  $T_I$  increases overshoot ratio and vice versa.
6. Decreasing  $T_I$  increases damping and vice versa.
7. Decreasing  $T_I$  decreases stability and vice versa.
8. Increasing  $T_I$  decreases overshoot and vice versa.
9. Increasing  $T_D$  increases stability and vice versa.
10. Increasing  $T_D$  decreases rise time and vice versa.

The inputs to the expert system are 1) a desired numeric value for each of the features (for instance, desired overshoot is 10%, desired damping is 0.5, etc.), 2) a numeric weight for each feature indicating relative importance to the user, 3) maximum acceptable overshoot, and 4) initial PID values. Each transient is given a numeric score which is the sum of the normalized error in each feature multiplied by the feature's respective weight as introduced in [16]. The normalized error,  $ne$ , is computed as

$$ne = \frac{| \text{actual feature value} - \text{desired feature value} |}{\text{desired feature value}} \quad (3)$$

Thus, the error in each feature is scaled so an error of  $\pm 100\%$  has a value of 1.0 independent of units.

Based on the errors in the features, the expert system determines which way the tuning parameters must be altered. Since there will almost always be a conflict with respect to the direction of parameter adjustment when tuning to attain several desired features at once, the PID terms are modified in the direction which will have the most impact. To determine which direction this is, the score for each feature is multiplied by +1 or -1 to indicate an increase or decrease in the value of the particular tuning term. The signed scores associated with each controller term are then added and the sign of the sum determines the direction of change for that term. In the situations where more than one tuning parameter affects a feature, the feature's entire weighted error is assigned to the term which, in order of importance, will

1. Reduce overshoot if overshoot is close to or above the maximum allowed.
2. Have the smallest impact on other features.
3. Improve stability.

Once the direction of adjustment for each term is inferred, the change is implemented.

A fibonacci sequence is used to determine the amount of change. A fibonacci sequence is an ordered set of numbers defined recursively as  $f(n)=f(n-1)+f(n-2)$  for  $n \geq 3$ . An example is  $\{f(n)\}=\{1, 1, 2, 3, 5, 8, 13, \dots\}$ . The amount of change in each controller parameter is the reciprocal of  $f(n)$  times the value of the term times the sign of the sum of weighted errors associated with that controller parameter. The general equation for updating a term is

$$\text{term}(n) = \text{term}(n-1) \left( 1 + \frac{1}{f(n)} \text{sgn} \left( \sum_{i=1}^R s_i w_i n e_i \right) \right) \quad (4)$$

$$\text{where } \text{sgn}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$$

$R$  is the number of features assigned to the particular term,  $s_i$  is the sign of the weighted error for the  $i$ th feature determined by the expert system,  $w_i$  is the weight assigned by the user to the  $i$ th feature, and  $n e_i$  is the normalized error of the feature defined in (3). Each of the three terms uses its own fibonacci sequence so, since all three tuning parameters might not be updated on each iteration, the value of  $n$  in (4) can be different for each term. This technique of adjusting the controller gains will never allow the values to become negative and will cause the values to converge after a finite number of trials.

The feature values and weighted errors, total score, and PID parameters for each response are stored in a frame [17]. A frame is a data structure which consists of slots, facets, and values. Each frame represents one transient and the slots it contains each represent an important piece of information about the transient such as a feature. Each slot has associated facets which contain the name of a specific piece of information describing the slot and a value corresponding to it. Figure 2 shows a frame with all of the slots, facets, and values filled.

The transient which receives the lowest score is considered the best by the expert system. This information is not used until the controller terms have converged, however. Since the expert system is emulating a human tuner, it bases its adjustments on differences between the most recent achieved response and the desired response. The justification for this strategy is that the rules contain the information needed to achieve a minimum score and they will find it after enough iterations. Thus, it is possible that tuning may actually cause a higher score from one transient to the next by making too large a change and overshooting the optimal set of parameters in the  $(K_C, T_I, T_D)$  space, for instance. Since the step size is successively reduced, the rules should cause the score to approach a minimum in the long run. If the change in the value of each PID gain is small compared to the respective gain's total value and the score is relatively constant, the expert system will invoke rules which consider the total

TRANSIENT#0001		
percent overshoot	value	16.38
	weighted error	5.43
damping	value	0.086
	weighted error	16.56
rise time	value	1.87
	weighted error	0.63
period	value	8.31
	weighted error	2.92
overshoot ratio	value	-0.118
	weighted error	29.43
total score	value	54.97
$K_C$	value	20.0
$T_I$	value	50.0
$T_D$	value	4.0

Figure 2. Example of a frame used to store transient data.

score of the transient. Once the values of the parameters and the score have converged, if the overshoot error of the most recent response is significantly above the desired value and it is weighed heavily in the scoring, a rule is fired which decreases  $K_C$ , increases  $T_I$ , and resets the fibonacci sequences to the beginning. If the overshoot error is not large enough to trigger this rule, the parameters are set to those of the lowest-scoring transient and the tuning process terminates.

### EXAMPLE

The open-loop system used in the example was of the type in (2) with  $\tau_1=\tau_2=10.0$  seconds and the dead time approximated by a first order lag with a time constant of 1.0 second. Euler integration with a time step of 0.01 seconds was used to evolve the system through time. These two approximations (a lag to represent dead time and numerical integration) introduce small differences between the simulation and the ideal system in (2) which help verify the robustness of the rules. In the example, a quarter-decay response was desired. Thus the desired feature values and their weights were



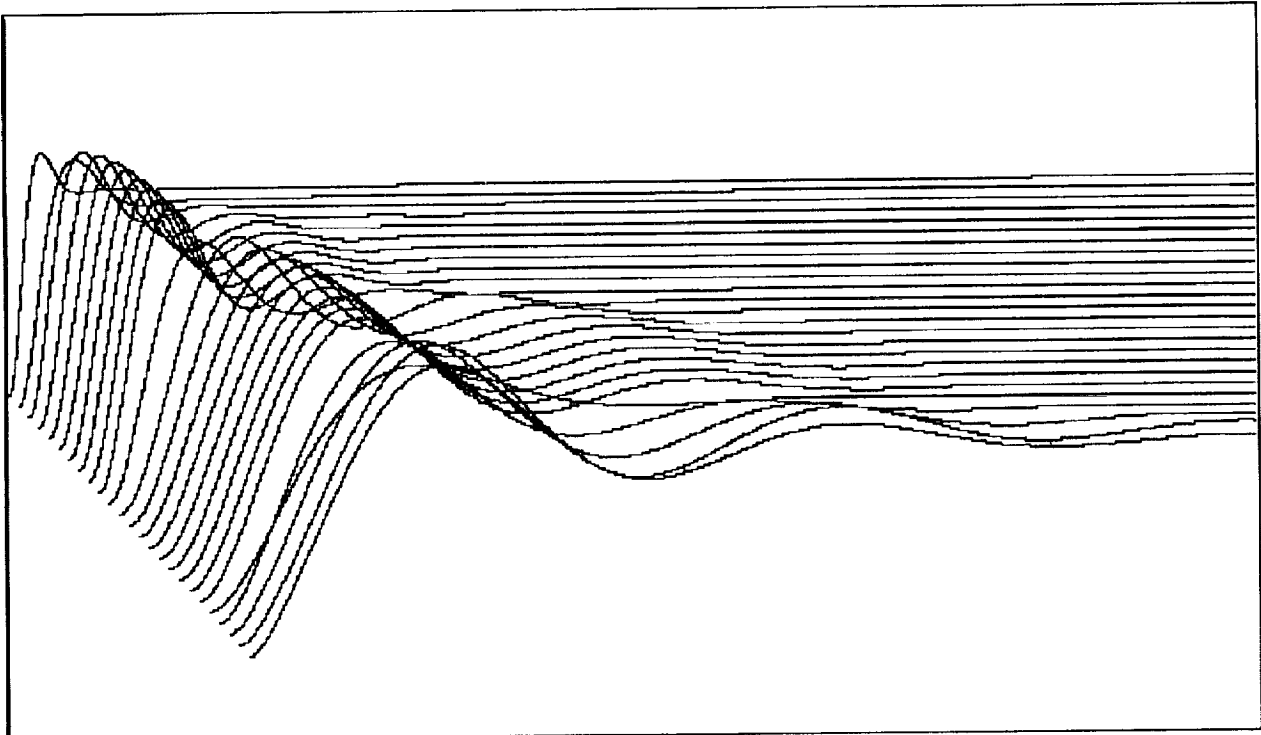


Figure 3a. First 25 step responses from example.

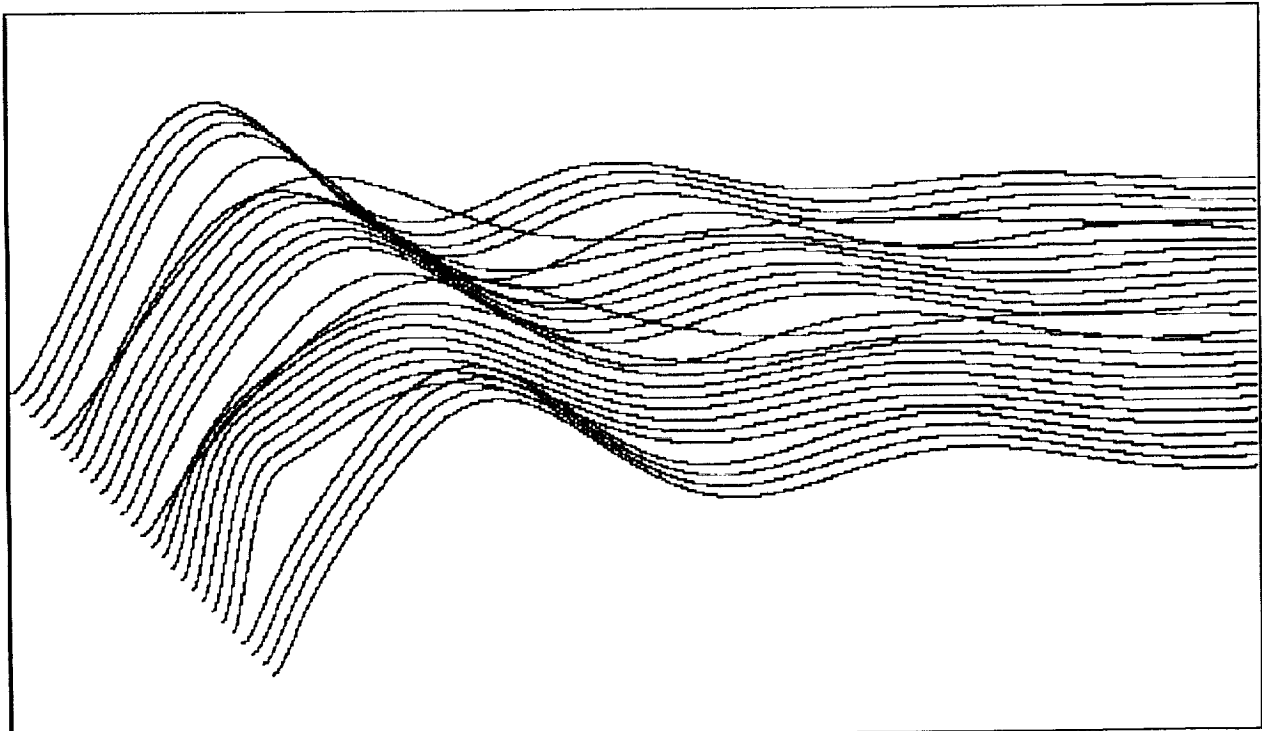


Figure 3b. 26th through 52nd step response from example.

specified as in Table I with a maximum overshoot value of 45% allowed.

<u>FEATURE</u>	<u>DESIRED VALUE</u>	<u>WEIGHT</u>
percent overshoot	20	30.0
overshoot ratio	0.25	20.0
period	20.0 seconds	5.0
rise time	5.0 seconds	1.0
damping	0.5	20.0

Table I

It can be seen from the weights associated with rise time and period that these features were not considered significant in the quality of the response. The initial controller values were given as  $K_C=20.0$ ,  $T_I=50.0$ ,  $T_D=4.0$ . Table II lists the transients chronologically along with their respective PID parameters, feature values, and score.

Figures 3a and 3b show all of the step responses listed in Table II with transient number increasing as the starting points move from upper left to lower right.

The first ten transients were the result of the standard tuning rules applied to the errors in the features. The parameters converged with the score hovering around 41. At that point, the large error in the overshoot prompted the expert system to take more drastic action by significantly decreasing  $K_C$  and increasing  $T_I$ .

The score of the eleventh response was much lower than those of the previous transients, about 33. Percent overshoot and damping were fairly close to desired but there was a large error in overshoot ratio. This was the major contributor to the score. Unfortunately, damping and overshoot ratio, which were both too small, were working against percent overshoot, which was too big. In an attempt to fix the three errors simultaneously, the expert system slowly decreased both  $K_C$  and  $T_I$ . This resulted in the score rising, however, because percent overshoot increased even more. After nine more transients, when the parameters converged, the score had risen to almost 45. Since percent overshoot was quite high again,  $K_C$  was decreased and  $T_I$  was increased again.

Even though the twenty-first response was very different from the previous few, by coincidence the score was still about 45. This time, however, the three main features were all too small. Based on this, the expert system was able to tune so that the scores for the twenty-second through the twenty-ninth response went down slowly to a low of about 39. The damping and overshoot were almost exactly the values desired but percent overshoot was very high so  $K_C$  and  $T_I$  were respectively decreased and increased again.

The thirtieth step response had a very high score, over 68. The overshoot error was small but there was no second peak so damping and overshoot ratio were both zero, and period was given its maximum value, 100.00, which contributed 20 points to the

Transient Number	$K_c$	$T_I$	$T_D$	Percent Overshoot	Damping	Rise Time	Period	Overshoot Ratio	Score
1.	20.00	50.00	4.00	16.38	0.09	1.87	8.31	-0.12	54.97
2.	13.33	33.33	2.67	17.89	0.00	3.07	100.00	0.00	63.56
3.	16.00	26.67	2.13	27.15	0.22	2.82	16.00	0.00	43.17
4.	14.00	23.33	1.87	30.28	0.27	3.17	17.43	0.05	41.88
5.	12.92	21.54	1.72	32.33	0.30	3.38	18.27	0.07	41.43
6.	12.31	20.51	1.64	33.63	0.32	3.50	18.79	0.09	41.28
7.	11.95	19.91	1.59	34.45	0.33	3.57	19.11	0.09	41.24
8.	11.73	19.55	1.56	34.95	0.34	3.62	19.31	0.10	41.22
9.	11.60	19.33	1.55	35.26	0.34	3.65	19.43	0.10	41.22
10.	11.52	19.19	1.54	35.45	0.35	3.66	19.50	0.11	41.23
11.	7.64	25.48	1.53	25.11	0.31	4.93	24.29	0.04	33.19
12.	5.10	16.99	1.02	31.91	0.36	6.19	29.86	0.12	36.99
13.	4.08	13.59	1.22	31.57	0.32	7.01	34.48	0.10	40.54
14.	3.57	11.89	1.38	32.28	0.31	7.52	37.69	0.10	42.84
15.	3.29	10.98	1.48	33.02	0.32	7.83	39.67	0.10	43.93
16.	3.14	10.45	1.55	33.57	0.33	8.03	40.86	0.11	44.43
17.	3.04	10.15	1.60	33.95	0.33	8.14	41.56	0.11	44.66
18.	2.99	9.96	1.63	34.19	0.34	8.21	41.99	0.11	44.78
19.	2.96	9.85	1.65	34.35	0.34	8.25	42.24	0.12	44.84
20.	2.93	9.78	1.66	34.45	0.34	8.28	42.40	0.12	44.87
21.	1.95	12.99	1.66	17.37	0.17	12.07	55.37	0.03	44.98
22.	2.60	8.66	2.22	33.78	0.35	8.91	46.23	0.12	43.87
23.	2.08	6.93	2.66	38.98	0.45	9.70	49.41	0.20	42.32
24.	1.82	6.06	3.00	42.38	0.51	10.15	50.83	0.26	44.09
25.	1.96	6.53	3.23	38.06	0.46	10.08	51.18	0.21	40.36
26.	1.86	6.22	3.38	39.23	0.48	10.25	51.68	0.23	39.65
27.	1.81	6.03	3.48	39.95	0.50	10.35	51.96	0.25	39.18
28.	1.78	5.92	3.54	40.40	0.51	10.42	52.13	0.26	40.54
29.	1.80	5.99	3.58	39.72	0.50	10.41	52.22	0.25	38.85
30.	1.19	7.93	3.61	24.37	0.00	15.34	100.00	0.00	68.62
31.	1.59	5.29	4.81	39.44	0.53	11.09	54.79	0.29	43.25
32.	1.90	6.34	5.77	28.22	0.41	11.02	58.02	0.17	33.25
33.	1.67	5.55	6.49	31.35	0.47	11.54	58.59	0.22	31.32
34.	1.54	5.12	6.99	33.18	0.51	11.82	58.94	0.26	31.74
35.	1.61	5.37	7.32	30.32	0.48	11.84	60.04	0.23	29.50
36.	1.56	5.21	7.54	30.96	0.49	11.95	60.17	0.24	29.00
37.	1.53	5.12	7.68	31.35	0.50	12.02	60.24	0.25	28.68
38.	1.52	5.06	7.76	31.59	0.50	12.07	60.29	0.25	29.26
39.	1.02	6.79	7.82	21.78	0.00	18.25	100.00	0.00	65.33
40.	1.36	4.53	10.42	30.10	0.53	12.93	63.56	0.28	31.40
41.	1.63	5.43	12.51	20.52	0.00	13.48	100.00	0.00	62.48
42.	1.83	4.75	14.07	20.45	0.00	11.91	100.00	0.00	62.06
43.	1.97	4.39	15.15	19.98	0.00	11.01	100.00	0.00	61.24
44.	2.07	4.18	15.87	19.53	0.00	10.48	100.00	0.00	61.79
45.	2.13	4.06	16.34	19.21	0.00	10.15	100.00	0.00	62.21
46.	2.17	3.98	16.64	19.00	0.00	9.96	100.00	0.00	62.50
47.	2.19	3.94	16.82	18.86	0.00	9.84	100.00	0.00	62.68
48.	2.21	3.91	16.94	18.77	0.00	9.77	100.00	0.00	62.80
49.	1.53	5.12	7.68	31.35	0.50	12.02	60.24	0.25	28.68
50.	1.53	5.12	7.68	31.35	0.50	12.02	60.24	0.25	28.68
51.	1.53	5.12	7.68	31.35	0.50	12.02	60.24	0.25	28.68
52.	1.53	5.12	7.68	31.35	0.50	12.02	60.24	0.25	28.68

Table II

score. The expert system immediately went to work on making the response more oscillatory. Over the next eight transients, the damping and overshoot ratio values were returned to their desired levels while overshoot was kept to about 31%. The score achieved was just below 29.

Since the overshoot error was still too large,  $K_C$  and  $T_I$  were adjusted again and the thirty-ninth response had no second peak and a score of about 65. The tuning process produced a score of about 31 for the fortieth transient but the excessive overshoot caused the parameters to be adjusted such that again there was no second peak. The result was scores in the neighborhood of 62 for the forty-first through forty-eighth transients even though their overshoot was very close to the desired value. Once the parameters converged, the tuning stopped because the error in overshoot was small. Since the thirty-seventh response had produced the lowest score, less than 29, those PID parameters became the permanent ones and were used for the remaining transients.

## CONCLUSIONS

The expert PID tuner is able to emulate a human PID tuner using strictly a classical, time domain approach. It works well for the examples tried including the example presented here although it may take many transients to converge. The rules are purposely vague enough that they do not fully model every nuance of the tuning map. This allows the rules to work reasonably well with systems similar to but not exactly of the form used to develop the map. In this way the expert system has wider applicability and is thus more useful.

In this heuristic approach to tuning, there is no guarantee that the parameters will converge such that the score is a global minimum. The tuning can be only as good as the rules and knowledge about the system being controlled. The expert system would benefit from the addition of rules for taking the parameters out of the area of a local minimum if it suspected that one was found. This does not imply that a more complete mapping of the  $(K_C, T_I, T_D)$  space is required, rather it points out that the current stopping criterion is too superficial.

## FUTURE WORK

Fuzzy algorithms [18] are not used here but controller tuning is an excellent application for them. The convergence of the tuning parameters could be speeded up significantly if rules were included which determine how much to adjust the value of a term based on the magnitude of the error in the feature.

An on-line transient stability check such as in [19] or [20] is also an extremely important addition. Were the expert tuner to be implemented on a real process, even though the tuning is done so as to improve stability whenever possible, an on-line stability monitor is an invaluable safety feature.

One reason that plant operators find dissatisfaction with closed-loop control is that on their highly nonlinear processes, a set of PID gains, optimal in one operating range, produces an unacceptable response in another. This expert system can be easily extended to find different sets of PID gains for different operating regimes. The only change required is that all transient data must be grouped according to operating region so that there will be many small groups of frames rather than one large one. As long as the rules, which are general to begin with, are fairly accurate over the whole operating range, the only time the grouping would even come into play is when the lowest-scoring transient for a particular region is required. This is essentially the idea behind gain scheduling in many modern applications.

Once a process is in steady state, closed loop control is used to provide disturbance rejection. Tuning maps have been developed for this application [21]. The procedure used to determine the rules for this expert system could be easily utilized to find rules for tuning PID controllers to handle loads.

The normalization of the error, although intended as an equalizing device, can cause problems in certain cases. If a 1% overshoot is requested and 2% is attained, the score for the error is the same as in the case where a 20% overshoot is requested and a 40% overshoot results. It is unlikely that a 100% error in the first case would be considered significant. This sensitivity issue should be addressed.

The reciprocated fibonacci sequence which was used to adjust the controller parameters was chosen arbitrarily because it is a convenient, convergent sequence. It is not the only sequence that could have been used. The implication of using a sequence of this kind, however, is that it causes the PID parameters to converge after a certain number of iterations rather than at a point of minimum score for the transient. Often the two correspond but sometimes they do not, witness the example. The results might be improved if the step size were decreased based on the change in the score, as in a gradient algorithm.

## REFERENCES

- [1] J. Martin, A. B. Corripio, C. L. Smith, "How to Select Controller Modes and Tuning Parameters from Simple Process Models," ISA Transactions, Vol. 15, No. 4, (1976), pp. 314-319.
- [2] A. H. Jones, B. Porter, "Design of Adaptive Digital Set-point Tracking PI Controllers Incorporating Recursive Step-Response Matrix Identifiers for Gas Turbines," AIAA Paper 86-2156, Guidance, Navigation and Control Conference, Williamsburg, VA, August 18-20, 1986.
- [3] L. C. Himmel, G. J. Geier, K. R. Fletcher, "Examination of Adaptive Control-Based Approaches to Friction Compensation in Spacecraft Gimbal Control Systems,"

Proceedings of the 24th Conference on Decision and Control, Ft. Lauderdale, FL, December 11-13, 1985.

[4] J. P. Gerry, "A Comparison of PID Control Algorithms," *Control Engineering*, March 1987, pp. 102-105.

[5] N. Andreiev, "A New Dimension: A Self-Tuning Controller That Continually Optimizes PID Constants," *Control Engineering*, August 1981, pp. 84,85.

[6] R. S. McEwen, D. P. Looze, "Quadratic Weight Adjustment for the Enhancement of Feedback Properties," 1982 American Control Conference, Arlington, VA, June 14-16, 1982.

[7] C. A. Harvey, G. Stein, "Quadratic Weights for Asymptotic Regulator Properties," *IEEE Transactions on Automatic Control*, Vol. 23, No. 3, June 1978, pp. 378-387.

[8] T. T. Jussila, H. N. Koivo, "Tuning of Multivariable PI-Controllers for Unknown Delay-Differential Systems," *IEEE Transactions on Automatic Control*, Vol. 32, No. 4, April 1987, pp. 364-368.

[9] H. N. Koivo, S. Pohjolainen, "Tuning of Multivariable PI-Controllers for Unknown Systems with Input Delay," *Automatica*, Vol. 21, No. 1, January 1985, pp. 81-91.

[10] D. M. Wills, "Tuning Maps for Three-Mode Controllers," *Control Engineering*, April 1962, pp. 104-108.

[11] F. G. Shinskey, *Process Control Systems*, McGraw-Hill Book Company, 1988.

[12] J. G. Ziegler, N. B. Nichols, "Optimum Settings for Automatic Controllers," *Transactions of the A.S.M.E.*, Vol. 64, No. 11, (1942), pp. 759-768.

[13] A. M. Lopez, J. A. Miller, C. L. Smith, P. W. Murrill, "Tuning Controllers with Error-Integral Criteria," *Instrumentation Technology*, November 1967, pp. 57-62.

[14] T. W. Kraus, T. J. Myron, "Self-Tuning PID Controller Uses Pattern Recognition Approach," *Control Engineering*, June 1984, pp. 106-111.

[15] R. Devanathan, et. al., "An Expert PID Controller," *Proceedings of the IEEE International Workshop on Artificial Intelligence for Industrial Applications 1988*.

[16] J. Litt, "An Expert System for Adaptive PID Tuning Based on Pattern Recognition Techniques," *Proceedings of the 1986 ISA North Coast Conference, Cleveland, Ohio, May 19-22, 1986*.

[17] P. H. Winston, B. K. P. Horn, *LISP*, Addison-Wesley Publishing Company, 1984.

- [18] L. A. Zadeh, "Fuzzy Algorithms," *Information and Control*, Vol. 12, No. 2, February 1968, pp. 94-102.
- [19] L. L. Scalzott, C. F. Lorenzo, "Practical Stability Criterion and Its Application to Digital Simulation," NASA Technical Note D-4203, December 1967.
- [20] J. M. G. Sá da Costa, N. Munro, "On-Line Transient Stability Analysis in Large Power Systems by Pattern Recognition Techniques," *Proceedings of Control 85*, IEE Conference Publication Number 252, Cambridge, U. K., July 9-11, 1985.
- [21] D. M. Wills, "A Guide to Controller Tuning," *Control Engineering*, August 1962, pp. 93-95.



# Report Documentation Page

1. Report No. NASA TM-103101 AVSCOM TR 90-C-007		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle An Expert System to Perform On-Line Controller Tuning				5. Report Date April 1990	
				6. Performing Organization Code	
7. Author(s) Jonathan S. Litt				8. Performing Organization Report No. E-5422	
9. Performing Organization Name and Address NASA Lewis Research Center Cleveland, Ohio 44135-3191 and Propulsion Directorate U.S. Army Aviation Research and Technology Activity—AVSCOM Cleveland, Ohio 44135-3127				10. Work Unit No. 505-62-0K 1L162211A47A	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001 and U.S. Army Aviation Systems Command St. Louis, Mo. 63120-1798				13. Type of Report and Period Covered Technical Memorandum	
				14. Sponsoring Agency Code	
15. Supplementary Notes Prepared for the 1990 American Control Conference sponsored by the American Automatic Control Council, San Diego, California, May 23-25, 1990.					
16. Abstract This paper describes an expert system which tunes a Proportional-Integral-Derivative (PID) controller on-line for a single-input-single-output multiple-lag process with dead time. The expert system examines features of the previous transient responses and their corresponding sets of controller parameters. It determines a new set of controller gains to obtain a more desirable time response. This technique can be used to determine and implement a different set of PID gains for each operating regime and, once in steady state, the system can be used to find optimal parameters for load disturbance rejection. The expert system can be applied to any system of the specified form (aerospace, industrial, etc.) and can be expanded to include additional process models.					
17. Key Words (Suggested by Author(s)) Controller tuning Expert systems Expert control PID tuning			18. Distribution Statement Unclassified—Unlimited Subject Category 63		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 14	22. Price* A03