RSD-TR-14-82

# SIMPLE ADAPTIVE CONTROL OF A ROBOT ARM

Dave E. Goldberg
A. Galip Ulsoy
Y. Koren

October 1982

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division
COLLEGE OF ENGINEERING
THE UNIVERSITY OF MICHIGAN
ANN ARBOR, MICHIGAN 48109

TABLE OF CONTENTS

## 1. OBJECTIVES

With increased interest in robots and robot-like manipulators, it is natural to inquire whether traditional linear methods are sufficient for effective control of robot arms. Robot arms carry a wide-ranging load at varying degrees of extension. As a result to obtain reasonable response it may be necessary to have a control strategy which adapts to changing conditions.

The purpose of this project is to, first, investigate the need for adaptive control in a single degree of freedom robot arm and, second, develop a computer package for evaluating adaptive control schemes.

In this report, we examine the motivation for adaptive control with some bracketing linear system calculations. An adaptive control scheme is defined and implemented and results of numerical simulation are discussed. Areas for further research are also addressed.

## 2. MOTIVATION FOR ADAPTIVE CONTROL

To determine the need for adaptive control, we perform some bracketing calculations based on a linear system model. Figure 1 shows a simple control loop with an integral controller and robot arm modeled as a first order lag. With this model, the major parameter is the arm time constant which is proportional to its moment of inertia. Assuming a nominal range for the time constant of 10 (varying from 0.1 to 1.0) and a selection of $K_I$ such that the performance is optimal with respect to an ISE criterion, we may look at performance of the system for five bracketing cases.

The parameters used for these cases are shown in the table below.

| Case No. | K | $\tau$ | Response Type |
|---|---|---|---|
| 1 | 0.5 | 1.0 | Optimal |
| 2 | 0.5 | 0.5 | Critical |
| 2 | 0.5 | 0 | First Order |
| 4 | 5.0 | 0.1 | Optimal |
| 5 | 5.0 | 1.0 | Under Damped |

To obtain optimal performance under an ISE criterion, we use the relationship $K = (1/2\tau)$ where $K$ is the overall open-loop gain. Numerical results are obtained from the analytical solutions of a second order system responding to a step input. These have been placed in the program ANAMOTOR presented in appendix B.

Physically the five cases are important as they demonstrate the need (or lack of need) for adaptive control. They indicate the limits of performance if we could maintain optimal control throughout any parameter variation. They also indicate the degradation of performance we might expect as a result of any lags in the adaptation process.

In figure 2, we compare cases 1, 2, and 3. The optimal case (#1) is underdamped with little overshoot as we expect with a damping factor of 0.707. Case 2 is critically damped. In fact, the waveform changes little as $\tau$ goes to zero (case 3) because the system equation approaches first order. These results are also interesting from a practical point of view. If the response of case 3 (first order system) is adequate, linear, integral control maybe sufficient and fancier schemes may not be needed.

However, as $\tau$ decreases, (decreasing moment of inertia) we know that we should be able to speed up response with proper controller tuning. We can see this in figure 3. Case 4 shows the optimal tuning for a minimum $\tau$ of 0.1. Note that the waveform is identical to the optimal tuning (case 1) with a compression of the time scale by a factor of 10. This makes sense if we recognize that the natural frequency for the system is given by $W_n^2 = \frac{K}{\tau}$; however, for the optimal tuning $K = 1/2\tau$ therefore $W_n = \frac{1}{\tau\sqrt{2}}$.

Looking at the other extreme, the response is underdamped with 49% overshoot. This is not surprising, as the tuning is only optimal for the minimum moment of inertia. This result is disturbing because an adaptive controller must track system parameters closely and quickly to maintain proper control with acceptable overshoot.

Is adaptation required? Clearly, the underdamped-first order response of cases 1, 2 and 3 with $K$ tuned for $\tau_{max}$ is consistent and relatively insensitive to $\tau$ changes. However, as $\tau$ decreases, we may desire the quicker response obtainable by tuning the controller with some adaptation mechanism. In the next three sections, we motivate one particular adaptive control scheme to see if some improvement is possible. We look at system equations, solution technique, and some results of numerical simulation to help evaluate the methodology.

## 3. SYSTEM EQUATIONS -ADAPTIVE CONTROL WITH I-ALGORITHM IDENTIFICATION

The physical system is essentially a motor, represented by a nonstationary, first order system, controlled by an integral controller which in turn is controlled by a model reference adaptive controller. The control strategy is to use the model-identified time constant to select an integral control coefficient that would be optimal if the identification procedure were perfect and the system were stationary. The result is a realizable algorithm which should give practical near-optimal results over a range of parameters.

We envision the physical system as four subsystems:

1) Motor-arm
2) Integral controller
3) Identification System
4) Adaptive Controller

An overall block diagram of this system is shown in figures 4 and 5.

The motor-arm, reference model and integral controller subsystems are represented by first order differential equations. Strictly speaking, the use of Laplace transform notation as in figure 4 is incorrect because the coefficients $K_I, \tau, \hat{\tau}$ are not constant; however, the intended meaning is clear for these subsystems and the equations are not examined further.

Instead, we identify the equations used for the identification algorithm and adaptive control scheme. The identification algorithm adopted is the so-called I algorithm [1,3]. For the first order system and reference model given by the model equations

$$\frac{dx}{dt} = ax + bu \qquad :system$$

$$\frac{d\hat{x}}{dt} = \hat{a}(t)\hat{x} + bu \quad :model$$

The I algorithm for this system, where $e = x - \hat{x}$, may be written as follows:

$$\frac{d\hat{a}}{dt} = K_A \ \hat{x}e$$

$$\frac{d\hat{b}}{dt} = K_B \ ue$$

This algorithm is easily implemented upon realizing that $a = (1/\hat{\tau})$.

The adaptive controller uses this information to calculate a quasioptimal value of control coefficient $K_I$. For a stationary system without disturbance $T_d$ an optimal setting of $K_I$ for given $K_M$ and known $\tau$ requires a damping coefficient $\zeta = \frac{\sqrt{2}}{2}$ using an ISE minimization criterion. This leads to the following relationship:

$$K_I = \frac{1}{2K_M\hat{\tau}}$$

This relationship dictates that the currently identified time constant ($\hat{\tau}$) directs the choice of controller coefficient $K_I$.

## 4. SOLUTION TECHNIQUE

Fourth order Runge-Kutta is chosen to solve the resulting system of equations. For a single model equation of the form, $\frac{dy}{dt} = f(y,t)$, 4th order Runge-Kutta may be written as follows [2]:

$$\begin{aligned}
k1 &= h * f(tn, yn) \\
k2 &= h * f(tn + \tfrac{1}{2}h, yn + \tfrac{1}{2}k1) \\
k3 &= h * f(tn + \tfrac{1}{2}h, yn + \tfrac{1}{2}k2) \\
k4 &= h * f(tn + h, yn + k3) \\
yn+1 = yn &= \frac{1}{6} * (k1 + 2k2 + 2k3 + k4)
\end{aligned}$$

For the system of equations presented in the previous section, we apply this technique by partitioning the equation set into two subsets, differential and algebraic relationship. For each substep of the R-K process, we first solve for the differential equations and then enforce the algebraic relationships. As a result, the procedure is both explicit and easy to implement.

## 5. RESULTS OF I-ALGORITHM TRIALS

To evaluate the proposed adaptive control scheme four simulations have been performed. To simulate the sudden extension and retraction of a robot arm, we model a step increase and a step decrease in the time constant $\tau$. In each case, we assume that initially the adapted $\tau$ and corresponding $K_I$ are set for the actual $\tau_o$. For each case, we examine the motor speed vs. time trace with the adaptation enabled and disabled.

For all cases, we specify a desired square wave input with zero mean, amplitude of one and a period of 4 seconds. In all cases, motor gain is taken as $K_M = 1$. Adaptation coefficients are $K_A = K_B = 1000$. The remaining parameters are specified in the following table:

| CASE TYPE | CASE NUMBER | SPECIFIED $\tau$ | INITIAL $K_I$ | ADAPTATION |
|---|---|---|---|---|
| Step Down | A1 | 1.0 -> 0.1 | 0.5 | Off |
| Step Down | A2 | 1.0 -> 0.1 | 0.5 | On |
| Step Up | A3 | 0.1 -> 1.0 | 5.0 | Off |
| Step Up | A4 | 0.1 -> 1.0 | 5.0 | On |

The results from the step down cases are shown in figures 6 and 7. The motor speed vs. time traces shown in figure 6 are not unexpected. The unadapted response is quite sluggish as we might anticipate from the linear bracketing computations of a previous section. In the adapted case, the response quickly approaches the optimal as the estimated time constant approaches the actual. The estimation of the time constant is shown in figure 7.

While the step down results are encouraging, the step-up results are not. The step up results, are shown in figures 8 and 9. The motor speed traces are not acceptable because of the high degree of overshoot. The unadapted case retains this overshoot throughout the run, while the adapted case approaches the optimal results as the time constant adjusts. We note that the so-called optimal results for this case are none too good as the rise time (4-5 second) is of the same order as the input signal period. Nonetheless, the adapted results are unacceptable because of the initial overshoot. The problem is further illustrated in figure 9 showing the estimated time constant trace slowly rising to approach the actual value $\tau = 1$.

## 6. CONCLUSION

In this report, we have examined the motivation for adaptive control of a robot arm. We have also examined results from numerical experiments with a particular adaptive control scheme.

The need for adaptive control may or may not exist depending upon performance requirements. Linear bracketing computations have shown that linear integral control with gain sized for optimal performance at maximum time constant provides response to a step function which is relatively insensitive to variations in time constant. This may be adequate for certain applications. Where it is not, some form of adaptation may be necessary.

A particular adaptive control scheme has been simulated using the I-algorithm and a gain adjustment heuristic. The results from these simulations are mixed. A simulated arm retraction shows a marked improvement over a case with no adaptation. The response approached the desired output in one period. A simulated arm extension causes unacceptable performance due to severe overshooting. The identification algorithm does not respond quickly enough. As a result, the controller thinks it is shorter than it is and thus, sets the gain too high, thereby, causing the overshoot difficulty.

While this experimentation is far from exhaustive it, does point in several important directions:

(1) Linear methods may be sufficient in may situations. Other standard methods, (e.g., PI, PID, and PDF, controllers) may be useful in obtaining quick response which is relatively insensitive to parameter variations.

(2) Where adaptive control is used either the algorithm must identify parameters quickly or it should be used to identify quantities which are expected to change slowly.

With these directions in mind, we recommend the following for further action:

(1) Other linear methods should be examined. If we can obtain acceptable response from simple methods why use fancy adaptation techniques?

(2) The I algorithm should be investigated more systematically. I was only able to do a few runs after getting a stable algorithm (version 4 ). It may turn out that high $K_A$ & $K_B$ values converge more quickly. (Care must be taken with high $K_A, K_B$ to set $\Delta$ t for numerical stability).

(3) Other adaptive control algorithms may be more fruitful. It is not clear why $\frac{d\hat{a}}{dt} = K_A \hat{x} e$ should be effective. Why should we use the product of $\hat{x}e$? For a linear system, the error at different $e$ levels assuming different starting points would be the same for the same error in parameter (a or b). Perhaps, the relay algorithm ( see [3]) would be more appropriate.

$$\frac{d\hat{a}}{dt} = K_A e \quad sgn(\hat{x})$$

(4) The I-algorithm results were fine as long as $\tau$ was being identified from above. This suggests an ad-hoc fix; reset $\hat{\tau}$ to $\tau_{max}$ at any load or extension change. This would cause sluggish but well damped response at first followed by increasingly desirable performance.

(5) Experiment and theorems in [3] suggest that parameter convergence requires input of sufficiently high frequency content. This implies that we may need a calibration signal superimposed over the demand input to assist in identification. A zero mean, low amplitude signal might help in keeping the system calibrated to the current parameter state.

While not entirely conclusive, this report sheds some light on a subject which will not doubt receive increasing attention over the next few years.

## 7. REFERENCES


1. Ulsoy, G., *Course notes for ME561*, University of Michigan, 1982.

2. Dahlquist, G. and A. Bjorck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N. J., 1974.

3. Landau, Y.S., *Adaptive Control Systems*, Marcel Dekker, 1979.

# 8. APPENDIX A - Description of Control Programs

## Implementation of Adaptive Control Testbed Programs

The system has been divided into two separate programs.

1) Model and report writer (ROBOT.V4)

2) Stripchart simulator (STRIP.PLOT)

ROBOT.V4 implements the system of equations, and coordinates problem input and output. Input comes from a single relatively, fixed format data file (ROBOT.DATA.TEXT). Output is directed to the console (CONSOLE:) and to a dump file for subsequent plotting (ROBOT.PLOT.TEXT).

STRIP.PLOT implements a rudimentary line printer graphics capability. The line printer is treated as a strip chart recorder with up to 6 signals plotted upon a single graph. The use specifies a gain and base for each signal, thus, controlling scaling and plot positioning. Setup input for this procedure is interactive with the user specifying the plot from the console (CONSOLE:). Mass data input comes from the dump file created by ROBOT.V4 although the program may be used to plot data from any program if the data file is in the proper format.

Both programs have been written and developed in UCSD Pascal and reside on 8 inch floppy disks appropriate for execution on the ME PDP 11/23 in room 2300 West Engineering. A copy of the Pascal source code is available from A. G. Ulsoy. In the remainder of this section, we discuss the program logic and input data requirements.

## Hierarchy Charts

Hierarchy diagrams are presented for each program to clarify the program flow and logic. Figure 10 is a diagram of the model program, ROBOT.V4 showing relationships among the major blocks of code.

In figure 11, a similar hierarchy diagram is presented for the strip chart simulator, STRIP.PLOT. Less detail is presented as this program is more straightforward in design and implementation detail.

## Input Data Description

An annotated input file for program ROBOT.V4 is available with the program listings. The data entry is self-explanatory. The user has complete control over the initial state in addition to the various system parameters.

## Dump Data Description

The dump data file is output from ROBOT.V4 and used as input to STRIP.PLOT. The data is output at each time step and thus provides a complete record of the simulated state time history. At each time step, a single record with 12 real values is output. Each real value is output with field width = 11. The values correspond precisely to the output report generated by ROBOT.V4. This report should be used as a guide in selecting the proper values in STRIP.PLOT.

## BRIEF PROCEDURE DESCRIPTIONS

| | |
|---|---|
| robot | - system model with input and output systems |
| input_parameters | - reads in input data from input file (file id=din) |
| initial_conditions | - calculates and initializes model initial conditions |
| report | - generates state variable report for each time step with page header |
| plot_report | - dumps state variable vector to a disk file (file id=plotfile) at each time step |
| time_calculations | - coordinates repetitive computations of each time step |
| runge_kutta | - coordinates system solution via R-K |
| delta_evaluation | - calculates derivative values |
| intermediate_values | - calculates state variables at intervals |
| average | - averages and finishes iteration of R-K |
| algebraic_evaluation | - calculates algebraic relations |

## OTHER UTILITY PROCEDURES

| | |
|---|---|
| space | - prints specified number of spaces on specified device |
| skip | - skip specified number of lines on device |
| uscore | - underscores a line with specified character a specified number of places on a device |
| linear-interp | - linear interpolation on time-ordered vector of time and dependent variable values |

## SYSTEM PROCEDURES

read, readln, write, writeln, reset, rewrite, close, page

## 9. APPENDIX B - Description of ANAMOTOR Programs

ANAMOTOR

Analytical Response of Motor

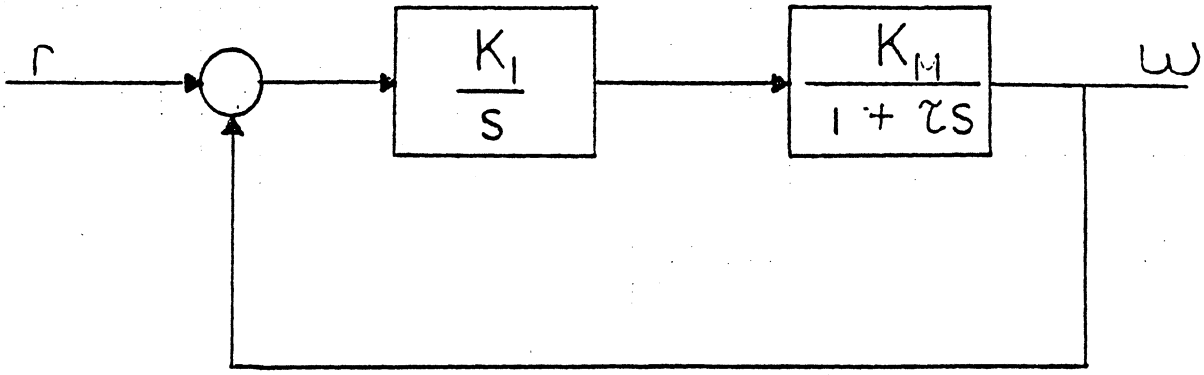
& Integral Controller to Step Response

*Hierarchy Diagram*

Motor
- reset files
- entry
- initialization
- initreport
- solvex

Program Listings Available from A. G. Ulsoy

Figure 1. Simple Motor Control Loop




WHERE : $K_M$ - motor gain

$\tau$ - motor time constant

$K_I$ - integral controller gain

$r$ - reference input

$\omega$ - speed output

**Figure 2. Comparison of Cases With** $K$ **Sized For** $\tau_{max}$ - Linear Bracketing Computations

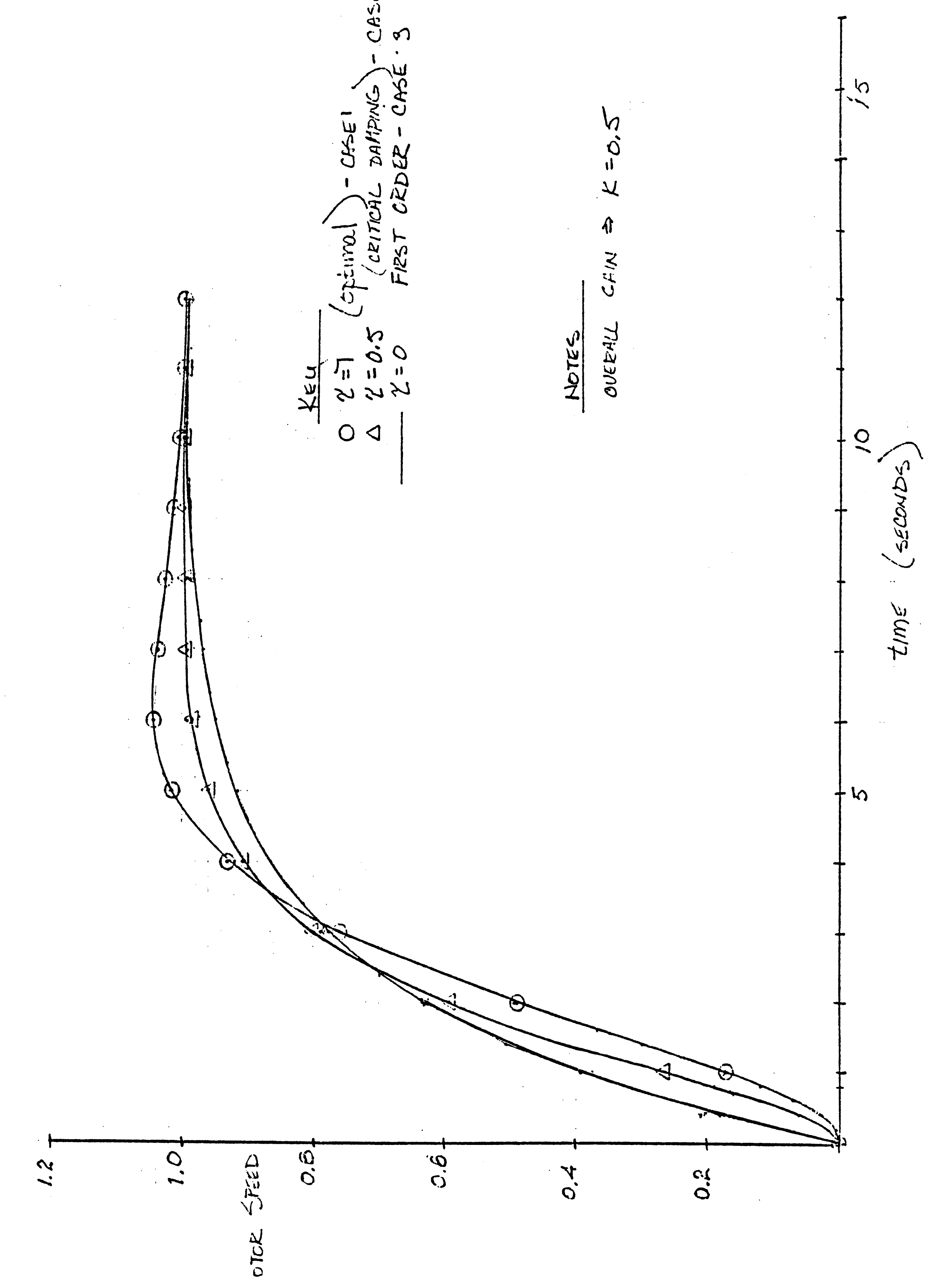


**Figure 3. Comparison of Cases With** $K$ Sized For $\tau_{min}$ Linear Bracketing Computations

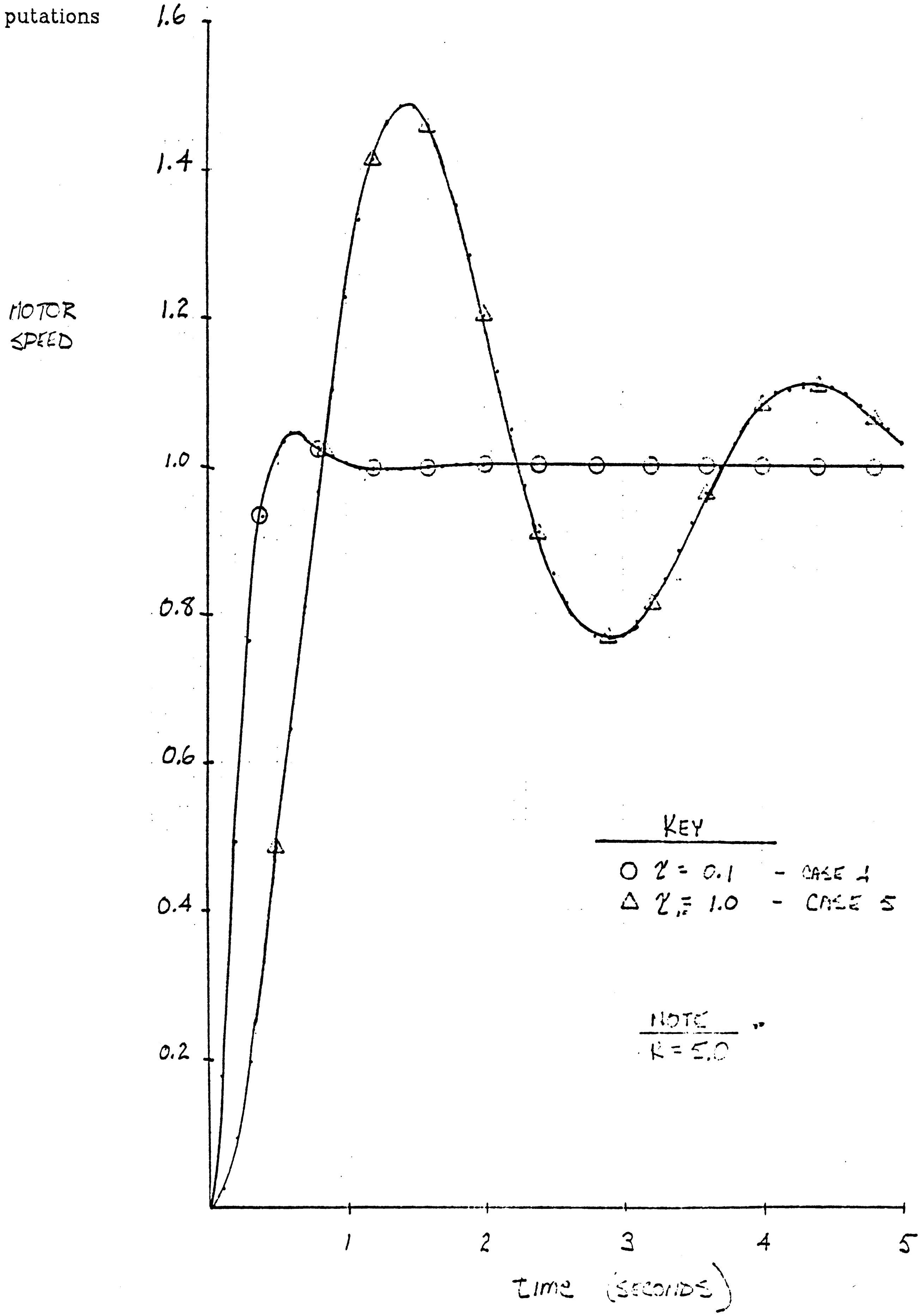


Figure 4. System Block Diagram Adaptive Control Testbed System

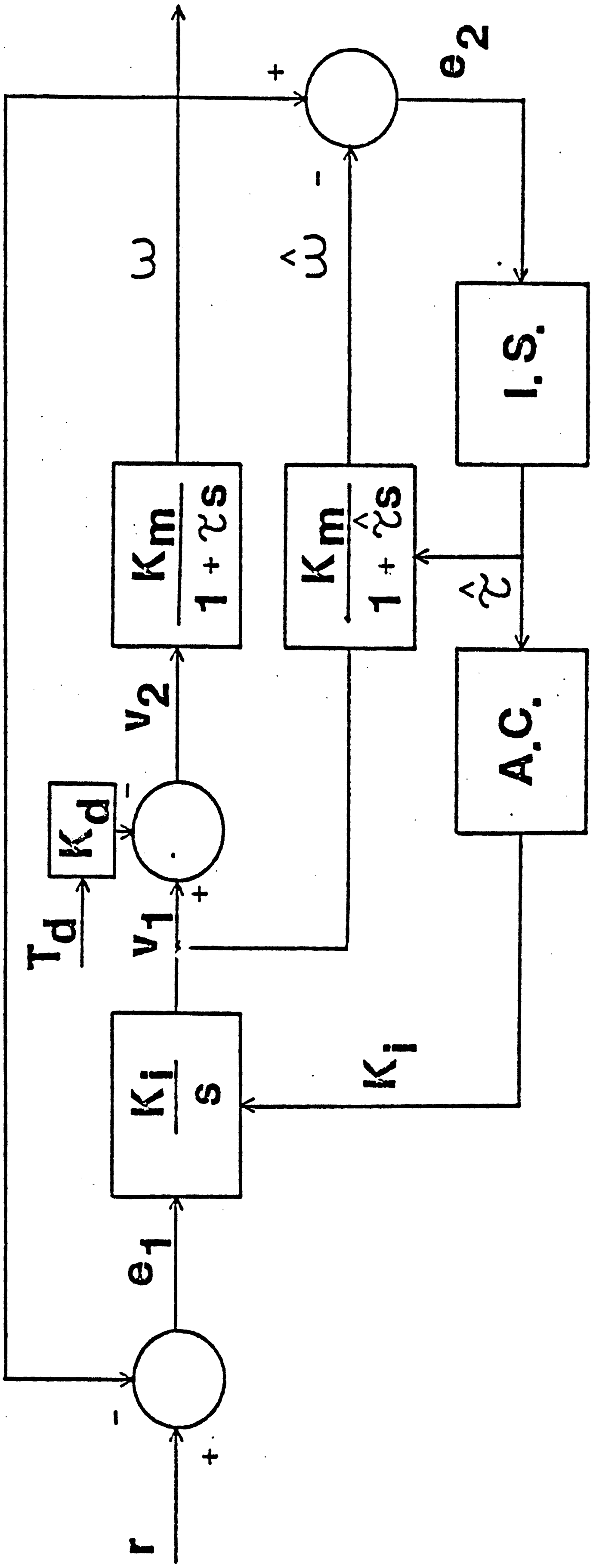


**Figure 5. Description of Variables for Block Diagram**

| | |
|---|---|
| r | - input signal (desired speed) |
| v1 | - controller output |
| $T_d$ | - disturbance torque |
| v2 | - disturbed controller output |
| $\omega$ (omega) | - motor-arm speed |
| $\hat{\omega}$ (omegahat) | - reference model speed |
| e2 | - error (model to actual speed) |
| e1 | - error (actual to desired speed) |
| $K_I$ | - integral controller gain |
| $K_m$ | - motor-arm gain |
| $\tau$ (tau) | - motor-arm time constant |
| $\hat{\tau}$ (tauhat) | - reference model time constant |
| $K_d$ | - disturbance torque gain |
| I.S. | - identification system |
| A.C. | - adaptive controller |

Figure 6. Comparison of Motor Speed Traces- Step Down Cases

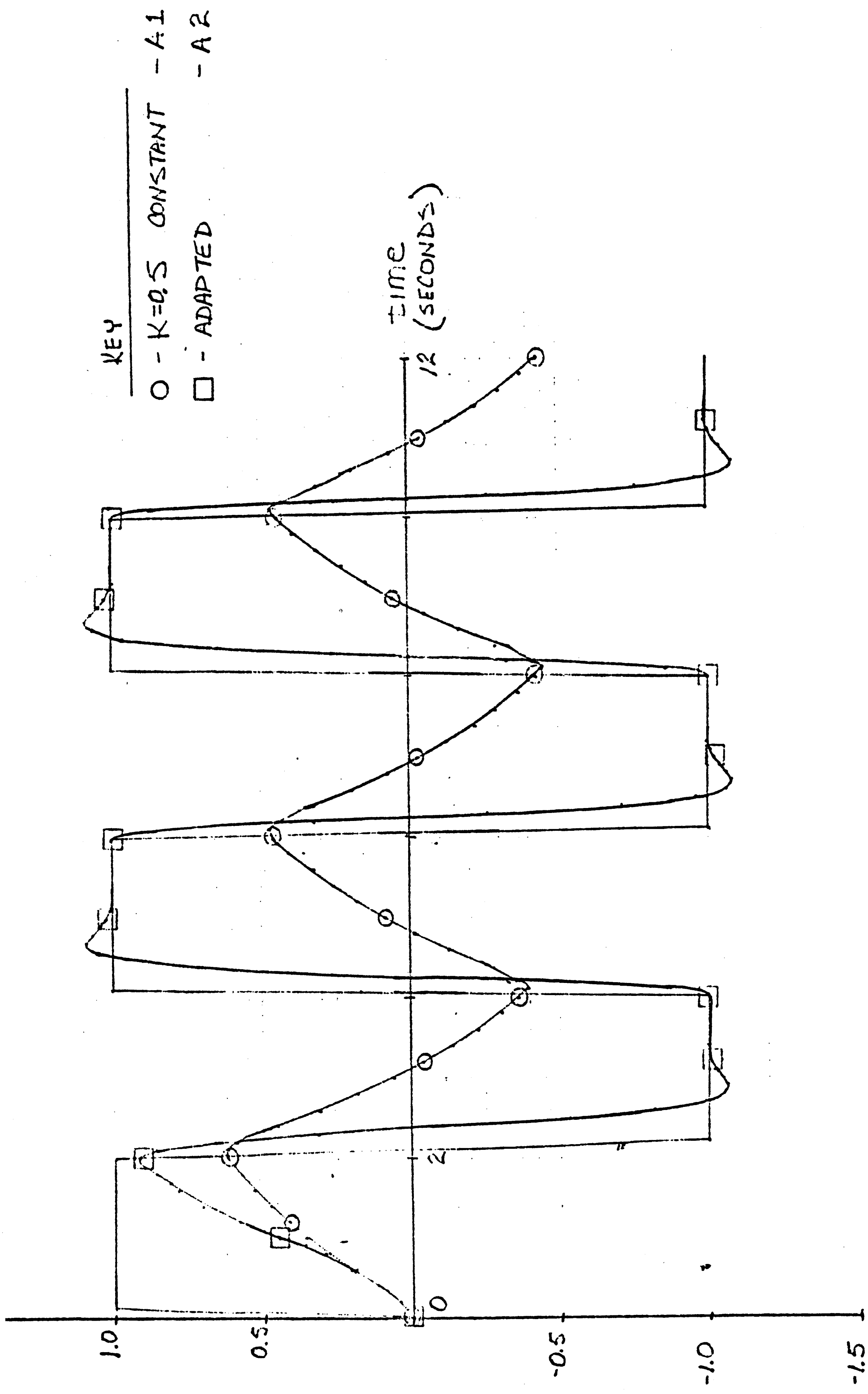


Figure 7. Estimated Time Constant Trace - Step Down Case

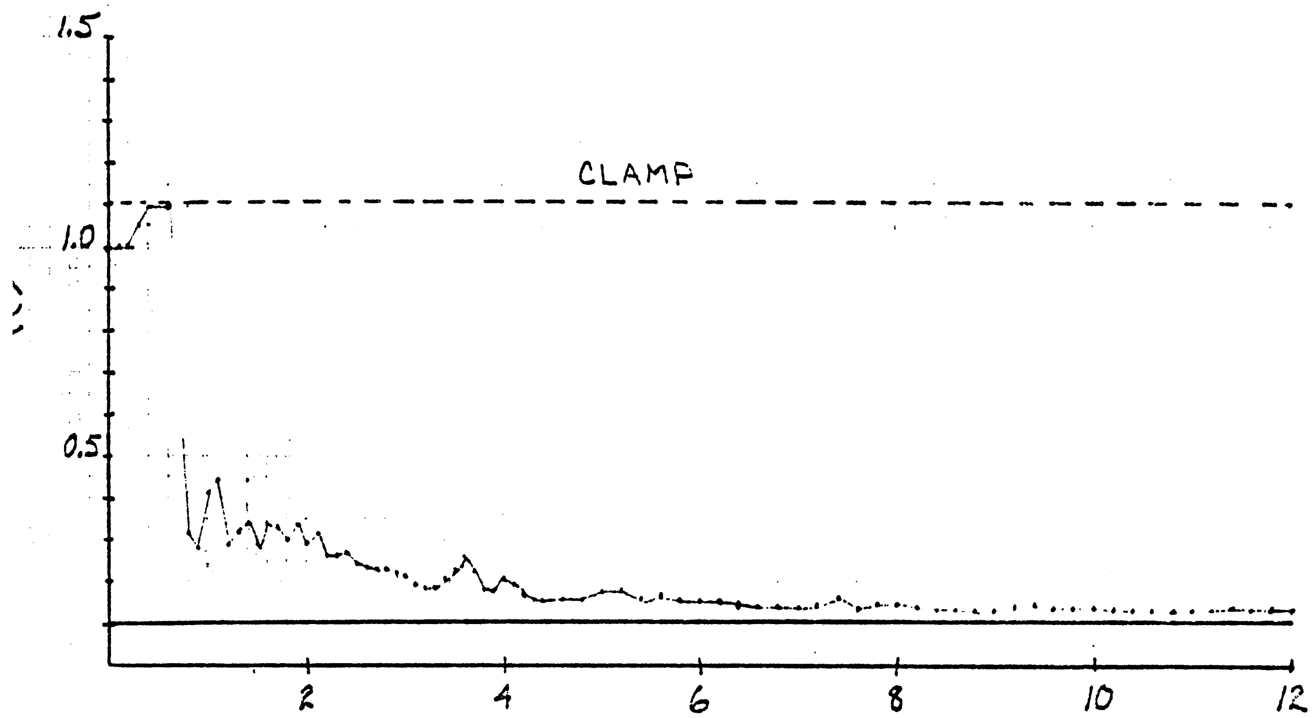


**Figure 8. Comparison of Motor Speed Traces- Step Up Cases**

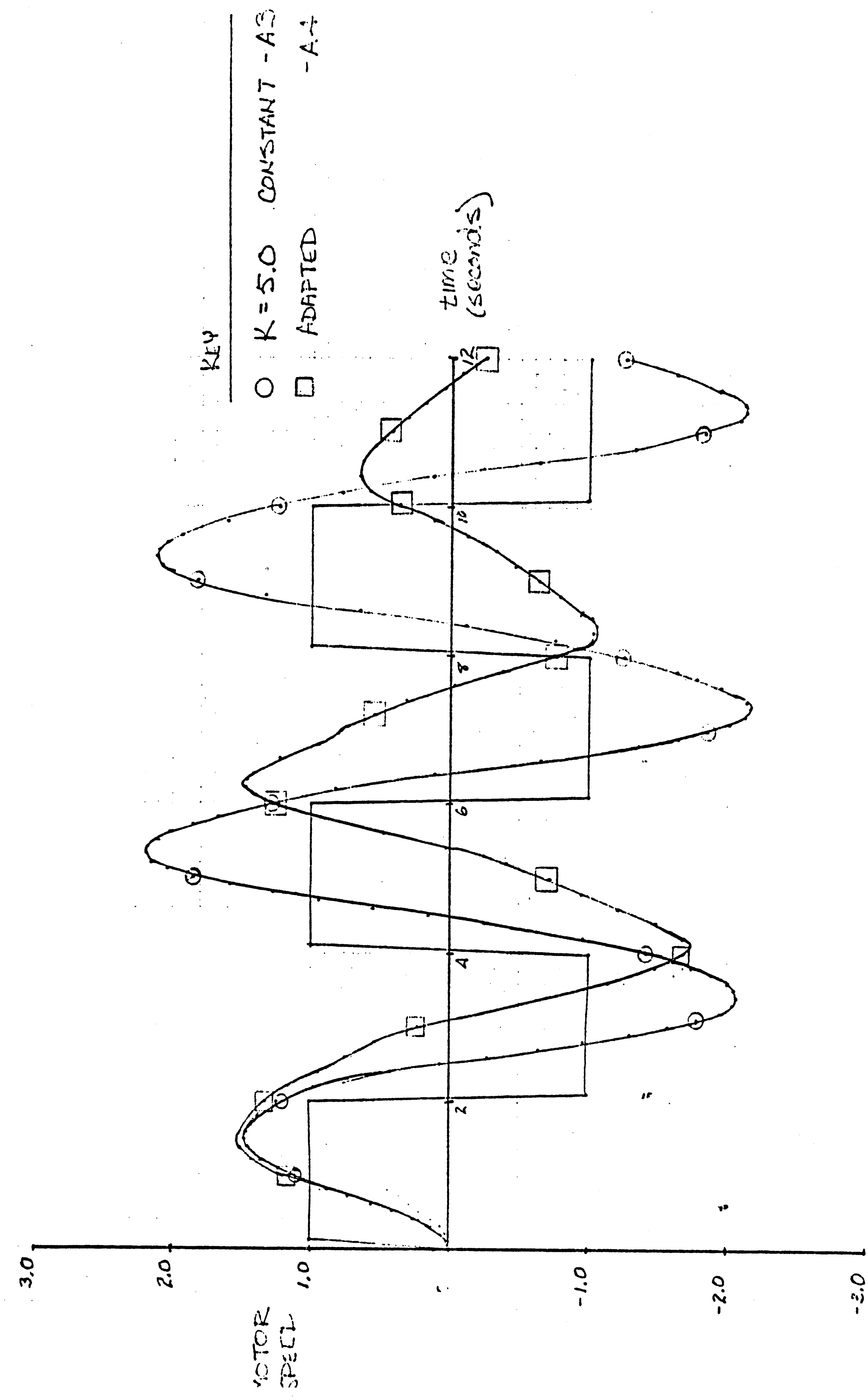


Figure 9. Estimated Time Constant Trace - Step Up Case

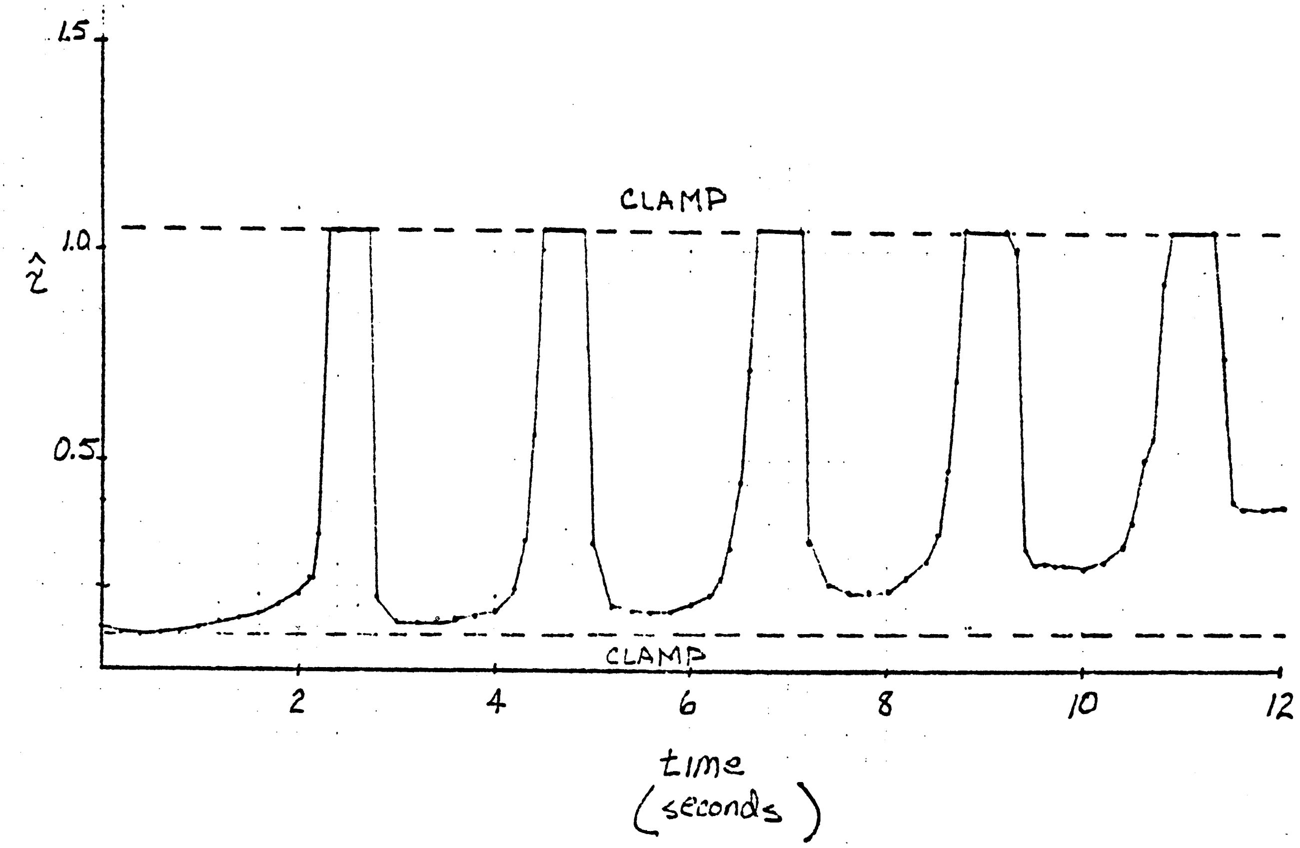


**Figure 10. Hierarchy Diagram for robot.v4**

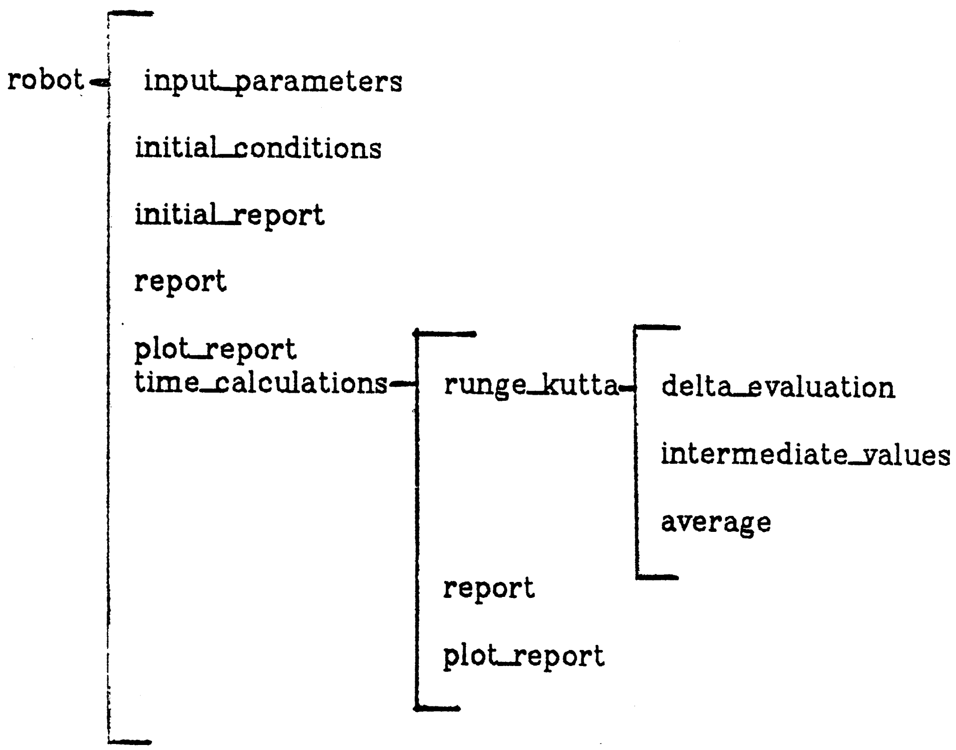


Figure 10. Hierarchy Diagram for robot.v4 (continued)

- intermediate_values
  - computations
  - algebraic_evaluation
- average
  - computations
  - algebraic_evaluation
- report
  - header & set up
  - output_line
- plot_report
  - output_line

Figure 11. Hierarchy Diagram for strip_chart

- strip_chart
  - read_data
  - header
  - plot_line
    - scale plot
    - grid_plot
    - read_line
    - vars_plot
    - print out the physical line