Fuzzy Logic Mode Selection for a Recuperative Turboshaft Engine

by

Kenneth H. Chiang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science

and

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1992

Author .....
Department of Electrical Engineering and Computer Science
May 8, 1992

Certified by .....
Sanjoy K. Mitter
Thesis Supervisor (Massachusetts Institute of Technology)
Thesis Supervisor

Certified by .....
Mark E. Dausch
Company Supervisor (General Electric Company)
Thesis Supervisor

Accepted by .....
Campbell L. Searle
Chairman, Departmental Committee on Graduate Students

Fuzzy Logic Mode Selection for a Recuperative Turboshaft Engine

by

Kenneth H. Chiang

Submitted to the Department of Electrical Engineering and Computer Science
on May 8, 1992, in partial fulfillment of the
requirements for the degrees of
Bachelor of Science
and
Master of Science

Abstract

The LV100 recuperative turboshaft engine, General Electric's entry into the US Army's Advanced Integrated Propulsion System competition, currently employs a hierarchical scheme using a number of low level controllers and a higher level mode selector. This selector uses a ladder of minimum and maximum functions to determine which one of the low level controllers should be in command of the engine at a particular time.

In the place of the min/max ladder, a fuzzy logic based mode selector combines the outputs of two or more of the low level controllers, allowing more than one controller to govern the engine at a given time. By taking advantage of the interpolative action of fuzzy logic, smoother switching between controllers is achieved, resulting in a response and a fuel consumption comparable to that of the conventional control scheme.

Thesis Supervisor: Sanjoy K. Mitter
Title: Thesis Supervisor (Massachusetts Institute of Technology)

Thesis Supervisor: Mark E. Dausch
Title: Company Supervisor (General Electric Company)

Acknowledgments

This thesis was brought to you by:

- Charlie Shortlidge, who patiently answered all my neophyte gas turbine questions;
- Mark Dausch, Piero Bonissone and Jim Comly, who gave me invaluable technical and practical help in attacking this and other related problems;
- Sanjoy Mitter, who took the time from his tight schedule to be my advisor;
- Saad Ayub, who has a number of great stories about Pakistan;
- Xuejun Cai, who should be able to finish *his* thesis sometime before I get mine signed off;
- John Cha, who hacks the stuff I never even want to touch with a 390 foot pole;
- Chen Dong, who taught me the MIT thesis salute;
- Scott Hwang, who managed to graduate before I did;
- Dave Yao, who stayed late nights on Athena with me;
- my parents, whose generous financial contribution made this whole endeavor possible;
- and my sisters, who provided much needed comic relief.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Designed as the power plant for the chassis of the next generation armored vehicle, the LV100 gas turbine is a joint entry by General Electric and Textron Lycoming into the competition for the US Army Advanced Integrated Propulsion System.

The two primary criteria upon which the power plants are judged are good acceleration and low fuel consumption. The gas turbine and its rival, the diesel engine, exhibit comparable acceleration characteristics, but at idle conditions and under partial loads, gas turbines historically have had greater rates of fuel consumption. The LV100 is intended to better this record without sacrificing performance[Baker].

The LV100 is controlled by a set of ten low level controllers, designed to govern the engine when specific conditions, or modes, are sensed. Each time step, a dominant controller is selected from that group by a ladder of minimums and maximums and its output passed on to the actuators.

Because only one controller is dominant at a given time, abrupt changes in control action may occur as regulation of the engine passes from one controller to another. To remedy this, a mode selector based upon fuzzy logic has been designed and implemented. Replacing the min/max ladder, the selector identifies modes and utilizes the interpolating action of fuzzy logic to blend the outputs of two or more controllers.

This is not the first time fuzzy logic has been used in a hierarchical fashion. Control of a small helicopter has been demonstrated; three low level controllers are used for roll, pitch, and yaw stability, while a fuzzy logic supervisory control processes pilot commands and perturbs the inputs to the stability controllers[Sugeno].

The second chapter briefly describes the engine and the principles behind its operation, as well as its software simulation. Control strategy is then detailed, along with its realization in the form of conventional low level controllers and the min/max ladder.

The third chapter gives a brief overview of fuzzy logic and its application to control theory. The design, implementation, and tuning of the mode selector is then discussed.

The fourth chapter compares the conventional proportional-integral controller with its fuzzy logic equivalent. The performance of the fuzzy logic controller is examined and contrasted with that of the nonlinear sliding mode controller. Then, the replacement of the conventional low level controllers with their fuzzy logic counterparts is discussed.

In the fifth chapter, the results of a number of tests run on the three types of control—the conventional low level controllers with min/max mode selection, the conventional controllers with fuzzy logic mode selection, and the fuzzy logic controllers with fuzzy logic mode selection— are analyzed.

The sixth and final chapter presents the conclusions drawn from this line of investigation.

Chapter 2

The LV100

In this chapter, the LV100 hardware and its software simulation are discussed. Next, a number of goals to be met in controlling the engine are presented, and finally each part of the current scheme of control is described.

2.1 Hardware

The LV100 is a recuperative turboshaft, with a high pressure spool supplying an airflow to a free power turbine, in order to generate the high torque necessary for moving a sixty ton vehicle from rest. To address the fuel consumption problem, both a heat exchanger (recuperator) and a variable area turbine nozzle (VATN)[1] are included to increase the efficiency of the thermodynamic cycle.

The thermodynamic cycle of the gas turbine bears some basic resemblance to that of the Otto cycle used in powering the typical Detroit automobile. In the automobile, reactions take place in a cylinder with a movable piston that alternately supplies power to and receives power from the engine crankshaft. When the cycle begins, the piston is at the top of its range of travel; as the crankshaft spins, drawing the piston downward, air is taken into the cylinder through an intake valve. Next, the valve is closed, a charge of fuel is injected, and the resulting mixture is compressed by an upstroke of the piston. The addition of a spark ignites the mixture and releases heat energy. The resulting increase in pressure forces the piston downward again in an expansion stroke that imparts rotational energy back into the crankshaft. Finally, an exhaust valve opens, the piston travels upwards once more, and the hot gases are expelled from the cylinder.

The operation of the gas turbine is based upon the Brayton cycle, in which intake, compression, combustion, expansion, and exhaust all take place simultaneously. As can be seen in the upper half of Figure 2-1, after entering the engine through an intake, air passes through a multistage axial-centrifugal compressor, which increases the pressure of incoming air from the ambient to a maximum value. Next, the compressed flow is warmed in the heat exchanger by the exhaust air. Fuel is injected into the combustor and ignited, providing the flow with additional heat energy. A high pressure turbine extracts pressure and heat energy from the flow and converts it to rotational energy, which is transmitted down a common shaft to the compressor. Together, the assembly of compressor, common shaft, and high pressure turbine is termed the high pressure spool, or core.

Upon exiting the high pressure spool, the flow passes through the variable area turbine nozzle, a set of vanes that control air flow through entire engine by the degree to which they are angled with respect to the flow. The free power turbine further extracts pressure and heat energy from the flow and converts it to rotational energy that is transmitted down a shaft to the six gear automatic transmission. The flow is then forced through the recuperator again, and finally released back into the atmosphere through the exhaust. A pressure-volume curve, shown in the lower half of Figure 2-1, relates the thermodynamic cycle to the location where that part of the cycle occurs in the engine.

[1] A number of commonly used abbreviations and their meanings are listed in Appendix A.

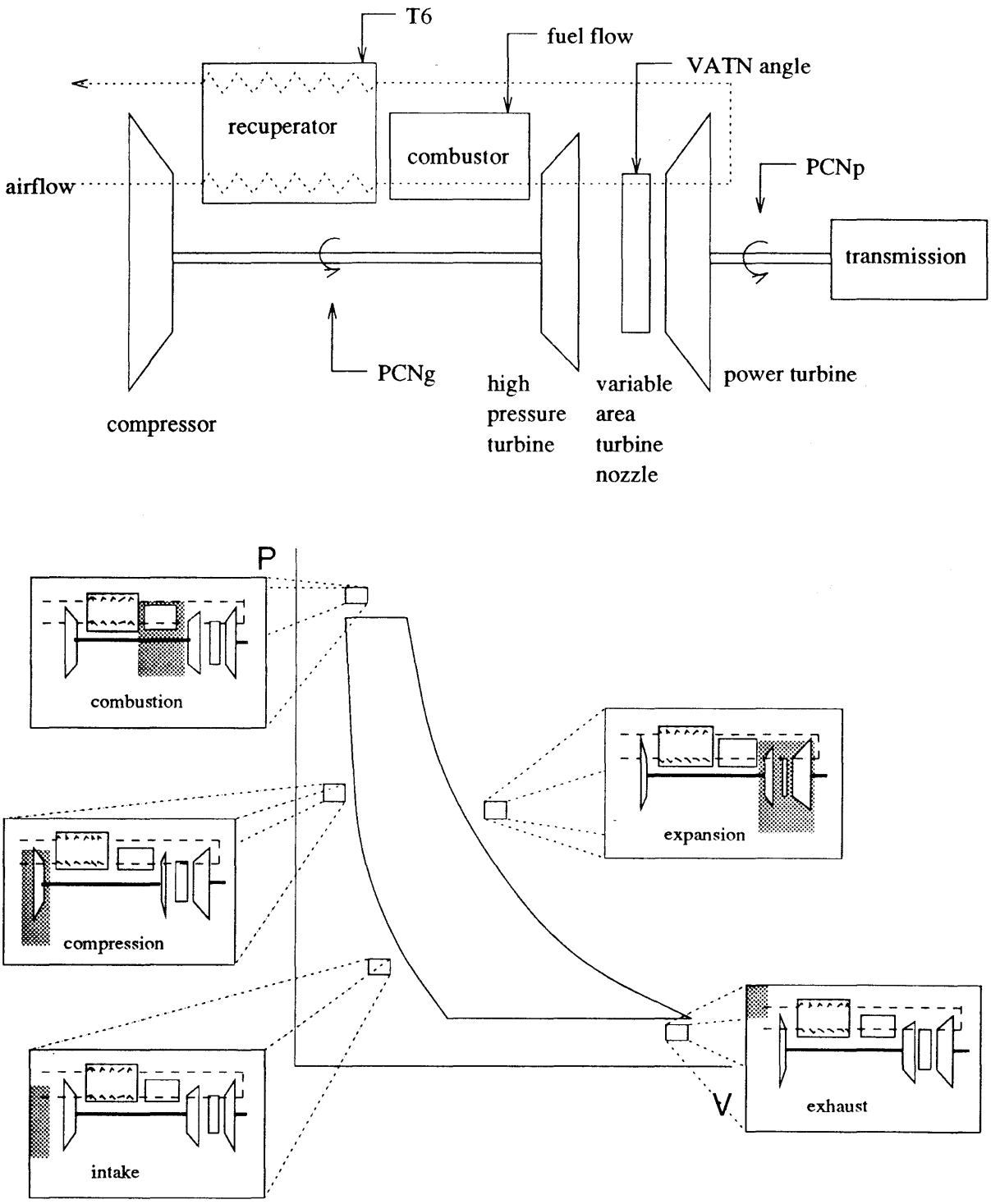

Figure 2-1: The LV100 component diagram and its pressure-volume curve.

2.2 Software Simulation

In the component level model of the LV100, obtained from General Electric Aircraft Engines in Lynn, Massachusetts, each of the engine's parts is modeled by a set of equations determining the enthalpy, pressure, and temperature at a given point in that component. First, a self-consistent operating point is calculated with a modified Newton-Raphson method. Then, changes from that steady state point are computed by a slightly less complicated version of that same numeric method.

However, this does not allow the model to run in real time, because a large number of iterations sometimes must be performed to get a solution within the specified tolerances. Thus, once the operating point is determined, the derivatives required by the Newton-Raphson method are obtained by consulting a previously computed table, and only one iteration by that method is permitted. This does allow some error to creep into the figures generated by the model, but these inaccuracies are negligible at the scale on which the data is examined.

A much more complete description of the modeling of the LV100 can be found in [Bevilacqua].

2.3 Control Strategies

There are three major goals in the control of the turboshaft portion of the LV100:

- to respond to requests from the driver.
- to operate the engine as efficiently as possible for optimum fuel consumption.
- to protect the engine against overtemperature or undue mechanical stress.

The operator indicates the desired power by a throttle setting, also referred to as the power lever angle, or PLA. A PLA of 62.0 corresponds to the minimum power possible, whereas a PLA of 95.0 is maximum power.

Three variables are available for use by the control system in realizing that power request: compressor vane angle, fuel flow, and VATN angle. Because satisfactory performance is obtained by scheduling the first variable as a function of corrected core speed, only fuel flow and VATN angle are actually independent.

Before discussing how changes in these two manipulated variables affect the engine, three crucial points must be addressed. First, the recuperator acts as a "phantom fuel flow" by scavenging from the exhaust air heat energy that would otherwise be wasted. This is not a problem upon acceleration, but in deceleration the fuel flow would naturally be decreased. However, the recuperator does not reflect a decrease in heat energy as rapidly, so a naive control law would attempt a greater negative change in the fuel flow. If the fuel flow falls below a minimum value for a given core speed, there will be insufficient amounts of fuel to support combustor operation, and the engine will *flame out.*

Second, the high pressure spool has a minimum speed to sustain the required minimum airflow, as well as a maximum speed to prevent mechanical stress. In addition, if its acceleration is too great, the core speed will increase while the flow will lag behind. Thus, the rear stages of the compressor will windmill, not having much air to act upon. The back pressure from the flow downstream of the compressor will attempt to force the flow back through that component, and the compressor will *stall.*

Third, the power turbine itself has a minimum speed to power assorted accessories at idle, most important of which is the cooling system. A maximum speed is also imposed, again to prevent mechanical stress.

The obvious correlation exists between fuel flow and power— if an increase in power is desired, the fuel flow must be increased. Unfortunately, if the increase is too great, one or more of the following may occur:

- the compressor may stall;
- various engine components downstream of the combustor may suffer stress from the high temperature or high change in temperature;
- the high pressure spool may overspeed;
- the power turbine may overspeed.

If a decrease in fuel flow is too great, one or more of the following may occur:

- the engine may flame out;
- the minimum airflow may not be supported;
- no power may be generated for the accessories.

Control of the fuel flow may be thought of as a coarse adjustment to engine operation, whereas fine-tuning of the thermodynamic cycle is achieved by controlling the VATN angle. When the VATN is fully open, the airflow is not obstructed. Closing it decreases the flow, increasing temperatures and pressures throughout the engine, as well as overall cycle efficiency. Hence high temperatures and pressures are encouraged, but excessively high values for these quantities may have the same detrimental qualities noted above. In addition, closing the VATN speeds up the power turbine, but this has a less pronounced effect than that of increasing fuel flow.

The control of the LV100 power plant is distributed. Thus, the turboshaft control does not coordinate any of its actions with the transmission control. At 90% of maximum power turbine speed, the shifting action from a lower to a higher gear is initiated, and any gas turbine control must be able to cope with the drastic changes in power turbine speed.

2.4 The Current Control Scheme

As can be seen in Figure 2-2, control of the LV100 is currently accomplished with a hierarchical approach. A number of low level controllers examine various sensor readings from points within the engine and produce fuel flow and VATN angle derivatives, also known as *wfdots* and *vtdots*, respectively. These rates then drop through a chain of minimum and maximum functions that selects one of the fuel flow rates and one of the VATN rates. The selected rates pass through integrators equipped with limiters, and finally on to the actuators. [2]

2.4.1 Low Level Controllers

The low level controllers can be divided into four main categories:

nominal:	Ng-T6 governor
engine protection:	Ng bottomer and Ng topper
	Np-T6 bottomer and Np-T6 topper
	T6 limiter
acceleration:	Ndot governor
VATN saturation:	Ng governor
	Np bottomer and Np topper

The *Ng-T6 governor* is designed to control the engine under normal quasi-steady state conditions, responding to the driver's commands while maintaining T6, the recuperator inlet temperature, at an optimal value. PLA is used as the reference for core speed; a precomputed recuperator temperature, scheduled as a function of corrected core and power turbine speeds, is used as the T6 reference. The governor is MIMO because synchronous action is required between fuel flow and VATN angle; upon acceleration, opening up the VATN too quickly without a corresponding increase in fuel flow will cause a rapid drop in power turbine speed, while increasing the fuel flow without opening the VATN may cause the engine to approach dangerously close to its thermodynamic limits.

In order to prevent the power turbine from both underspeeding and overspeeding, the *Np-T6 bottomer* and *Np-T6 topper* use fuel flow and VATN derivatives to keep the power turbine speed, or Np, in its normal range of operation while maintaining T6 at an optimum temperature. In the bottomer, whenever the power turbine speed approaches and drops below the minimum reference, the fuel flow is increased and the VATN is opened more slowly to minimize undershoot. In the topper, the opposite method of attack is employed—whenever the power turbine speed approaches and goes above the maximum reference, fuel flow is decreased

[2] The contents of this section have been summarized from [AIPS] and [Medeiros].

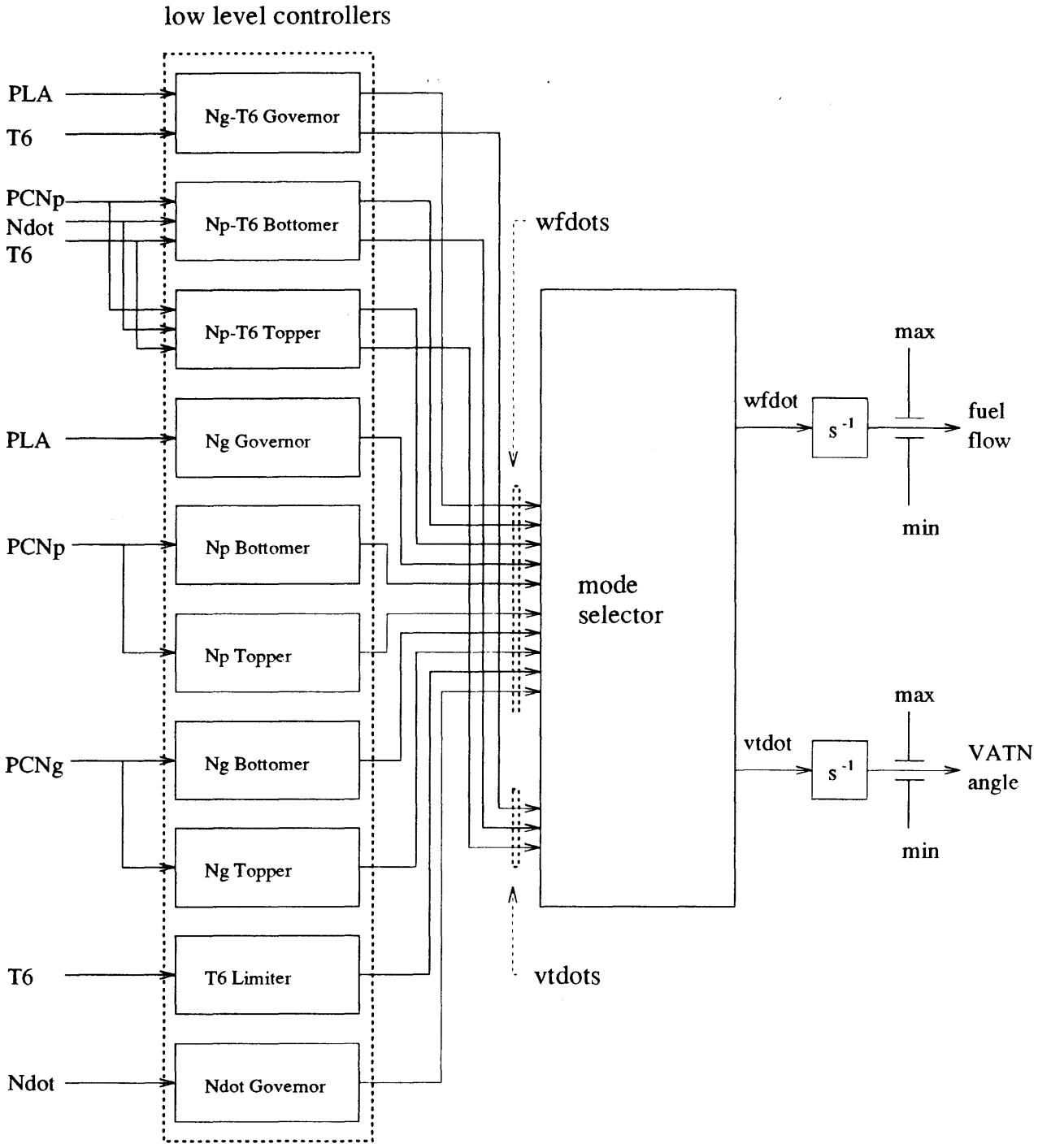

Figure 2-2: Conventional low level controllers and min/max mode selection.

and the VATN is closed more slowly to minimize overshoot. In both controllers, core acceleration is used as an additional feedback variable.

The SISO *Ng bottomer* and *Ng topper* manipulate fuel flow to prevent the high pressure spool from underspeeding and overspeeding. The structure of both controllers is similar. A single lead lag network acts on core speed. The lag acts as a low pass filter to reduce the effects of noise and decrease mode contention. The lead portion, when combined with the common fuel flow integrator, provides proportional-integral action. Each controller implements the obvious law; for the bottomer, whenever the core speed drops below its minimum value, the fuel flow is increased. The opposite action occurs in the topper.

Because temperature limitations are paramount, the SISO *T6 limiter* is designed to hold the engine temperature below the minimum of three temperature limits— the combustor outlet limit, the high pressure turbine outlet limit, and the recuperator inlet limit itself. After extensive study, it was found that compressor stall margin and the various temperatures throughout the engine could be determined from the temperature at the recuperator inlet. Thus, stall margin and temperature limits could be converted into T6 values as a function of both corrected core and power turbine speed, reducing the number of sensors required. The structure of the limiter is similar to that of the Ng bottomer and Ng topper, and it also handles the obvious control law: whenever T6 approaches its maximum, the fuel flow is decreased.

During transients, the SISO *Ndot governor* holds the acceleration of the high pressure spool to a reference value, to avoid compressor stall and mechanical constraints. This value is the minimum of three references: the maximum acceleration permissible based on compressor stall margin, the maximum based on high pressure turbine inlet temperature, and the maximum based on high pressure turbine outlet temperature. In addition to a lead lag network that compensates for the fuel pump, the limiter uses proportional-integral control to manipulate fuel flow, reducing it to decrease acceleration.

Saturation occurs when the VATN is fully open or closed. Three different controllers, each with a structure similar to that of the T6 temperature limiter, are specifically designed to control only fuel flow whenever this condition arises. The *Ng governor* uses PLA as a reference for core speed, like its MIMO counterpart, while the *Np bottomer* and *Np topper* attempt to hold power turbine speed within its range of operation, like their MIMO counterparts.

2.4.2 The Min/Max Ladder

Each of the controllers produces its actuator derivatives, which then pass through a ladder of minimum and maximum functions to select one of the fuel flow derivatives and one of the VATN derivatives. The structure of the ladder is shown in Figure 2-3, with the s^+ boxes indicating the maximum operator and the s^- boxes indicating the minimum operator. Although it appears complicated, the diagram is readily understood after some study.

As an example, consider the group of MIMO controllers near the top of the figure. The Np-T6 bottomer provides a minimum rate and the Np-T6 topper a maximum rate. If the rate of the Ng-T6 governor is less than that of the bottomer, the rate of the bottomer is chosen; if greater than that of the topper, the rate of the topper is chosen. This action is shown in the lower half of Figure 2-3.

The same reasoning is behind the set of SISO controllers, seen in the figure below the MIMO group. In a similar fashion, the Ng bottomer and Ng topper bound the resultant rate, while both the Ndot governor and T6 limiter attempt to hold that rate below their outputs.

Normally, all ten controllers are active. However, the MIMO controllers must be disabled whenever the VATN becomes saturated, because no coordination between fuel flow and VATN angle is possible when the VATN has moved to its minimum or maximum position.

Selection is thus accomplished in an intuitive manner. The most important controllers appear at the end of the chain, so that their rates may have the greatest effect. Whenever it is desired that a controller not be selected, its gain is adjusted so that the controller output appears excessive— so much greater or smaller than the rest that it would be pruned by the ladder.

The ladder actually operates on only the fuel flow derivative. The VATN angle derivative is determined based upon the fuel flow selected. If one of the MIMO controllers is chosen, its VATN derivative is passed to the VATN integrator. Otherwise, the VATN derivative of the Ng-T6 governor is bounded by those of the Np-T6 bottomer and Np-T6 topper.

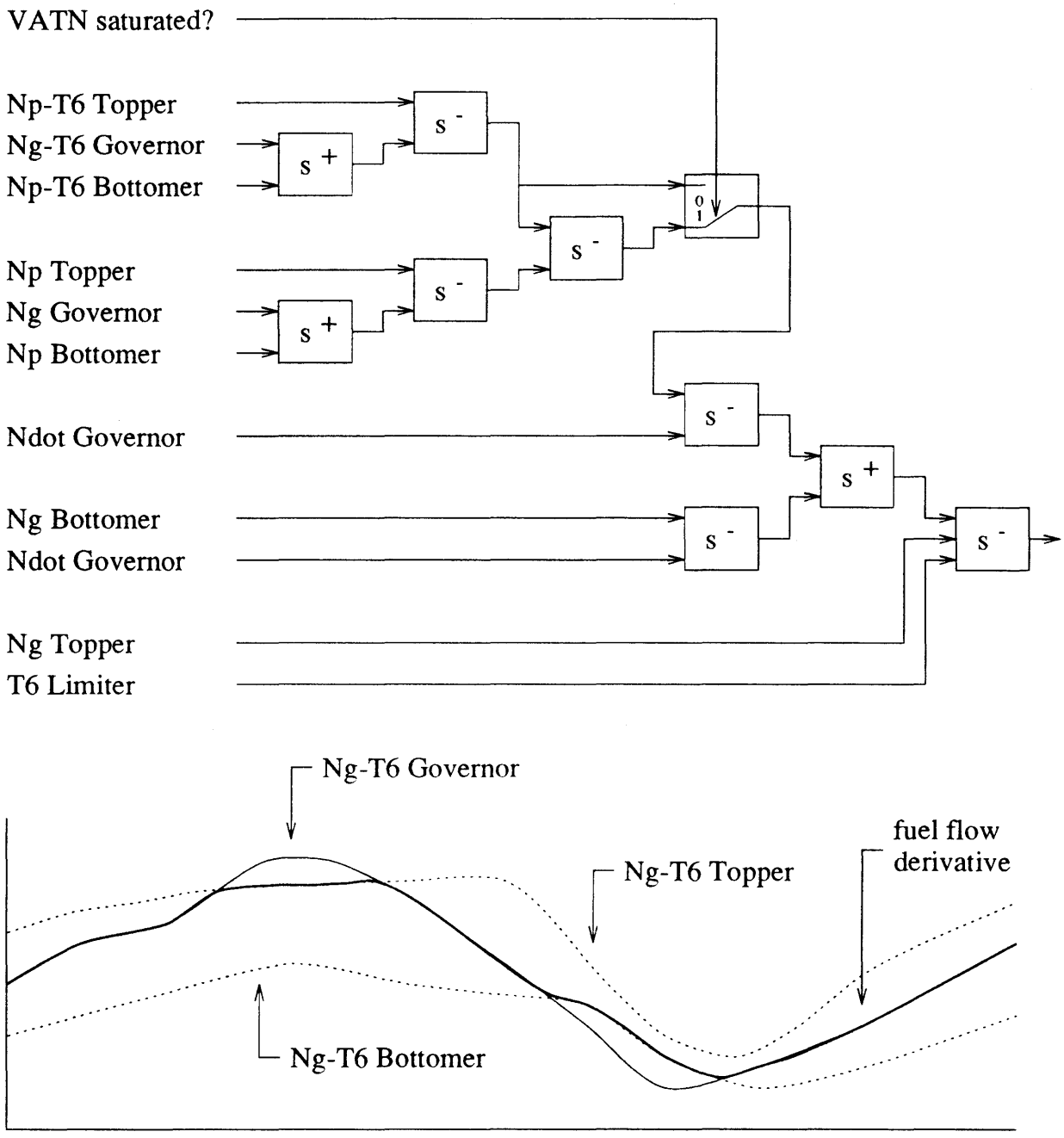

Figure 2-3: The min/max ladder and an example of the ladder in action.

2.4.3 Command Limiters

Once a dominant controller has been singled out, both derivatives are integrated and limited by schedules. The minimum fuel flow schedule protects against flame out on deceleration, and maximum schedule shadows the Ndot governor on accelerations. Both are functions of corrected core speed. The minimum VATN schedule provides a minimum airflow through the engine based on corrected core speed, while the maximum VATN schedule reflects physical limits on the actuator.

Chapter 3

The Fuzzy Logic Mode Selector

The min/max selection scheme was replaced by one implemented with fuzzy logic. Instead of choosing a single dominant mode based upon controller rates, the selection is accomplished by examining engine sensor values. There are three major advantages to this method:

- there is smoother switching between modes;
- the structure of the selector is more readily apparent;
- it is easier to design the low level controllers.

In this chapter, the basic concepts behind fuzzy logic and its application to control theory is discussed. The rationale underlying the design of the fuzzy logic mode selector is subsequently addressed. Tuning of the selector is examined in the final section.

3.1 Controls and Fuzzy Logic

Consider the variable x, defined over its universe of discourse X, the range of its possible values. Let boolean set B be defined on universe $\mathbf{X}$. For certain values of x, x would be in the set; for all other values, it would not. In other words, B could be thought of as having a membership function $\mu_B(x)$, mapping x into the set $\{0, 1\}$.

Fuzzy logic extends this concept of set membership, allowing x to be a member of fuzzy set F to a fractional degree. So, F would have a membership function $\mu_F(x)$, mapping x into the interval [0, 1].

Until recently, fuzzy logic has been used to perform reasoning with uncertainty in expert systems. If a controller is thought of as a real time expert system, then the application of fuzzy logic is obvious. A much more detailed explanation of fuzzy logic in relation to control theory can be found in [Lee]. [1]

As illustrated in the upper half of Figure 3-1, a typical fuzzy logic controller is composed of four major parts— a fuzzifier, an inference engine, an associated knowledge base, and a defuzzifier. The fuzzifier takes the sensor readings and converts them into fuzzy sets. Formally, if the controller has n inputs and each input i_j for $j = 1 \ldots n$ is a value for a state variable x_j defined over its universe of discourse $\mathbf{X_j}$, the fuzzifier takes input vector $\vec{I}$ and maps each of the n elements i_j into a fuzzy set I_j defined upon $\mathbf{X_j}$. In other words,

$$\begin{aligned} \vec{I} &= [i_1\ i_2\ \cdots\ i_n] \\ &= [\mu_{I_1}(x_1)\ \mu_{I_2}(x_2)\ \cdots\ \mu_{I_n}(x_n)] \end{aligned}$$

Usually, sensor readings are single-valued. So if sensor reading j has a value of x_0, then $\mu_{I_j}(x_j)$ is unity for $x_j = x_0$ and zero elsewhere. The property of single-valuedness is sometimes termed *crisp*.[2]

[1] This section itself was borrowed heavily from [Bonissone].

[2] Fuzzification can serve another purpose— that of encoding uncertainty in the state variable. For instance, a fuzzy set may be interpreted as a sensor reading in which one is most certain of the value of the reading at the point of the set with the greatest degree of membership.

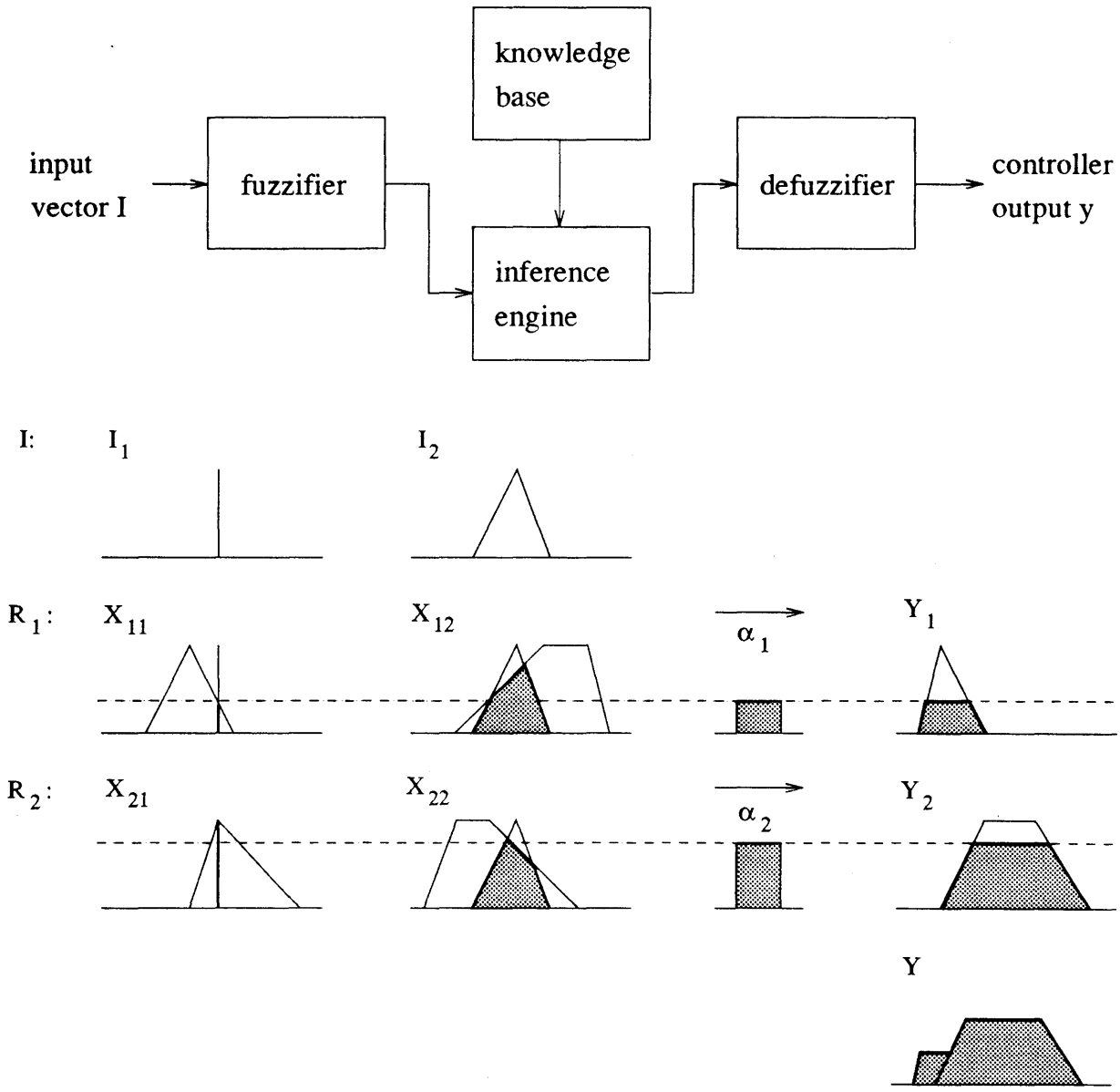

Figure 3-1: The breakdown of a typical fuzzy logic controller; inferencing on an input vector with a simple two rule knowledge base.

The input vector then passes onto the inference engine, which operates as a forward chaining system. The engine consults fuzzy production rules in the knowledge base to determine the degree to which each rule applies to the fuzzified input, and weights the output of that rule accordingly. Next, every rule output is collected together to give a fuzzy set representing the controller output.

More formally, consider the ith fuzzy production rule $R_i : \vec{X}_i \rightarrow Y_i$. The premises of the rule, $\vec{X}_i$, are the conjunction of n previously defined reference fuzzy sets X_{ij} for the state variable x_j, where $j = 1 \ldots n$. The degree to which a given X_{ij} matches an input fuzzy set I_j is the possibility measure $\prod(X_{ij}|I_j)$. If the pointwise conjunction operator is $\bigwedge$, then

$$\prod(X_{ij}|I_j) = \bigwedge_{x_j}(\mu_{X_{ij}}, \mu_{I_j})$$

Graphically, this can be interpreted as superimposing the membership function for the input fuzzy set on the membership function for the reference fuzzy set and then taking the minimum value of both functions at every point. The result of this operation is another function. In the case of a crisp value, say x_0, $\prod(X_{ij}|I_j)$ is equivalent to $\mu_{X_{ij}}$ at the point x_0.

In the lower half of Figure 3-1, a two element input vector $\vec{I}$ has already been fuzzified. Notice that I_1 is a crisp value, whereas I_2 has been converted into a fuzzy set. Inferencing is then done on a simple two rule knowledge base. The membership function for fuzzy set I_j is superimposed on the function for fuzzy set X_{ij}; the possibility measure of X_{ij} given I_j is denoted by the shaded area and darker lines.

To determine the degree of applicability α_i of the rule R_i, the possibility measures of each matching for the n elements are intersected in a technique called *min-max inferencing*. The maximum point of each possibility measure is determined, and then the minimum of these maximum points is taken. This minimum scalar value is α_i; if it is nonzero, the rule is said to have *fired*. Notationally,

$$\alpha_i = \min_j(\max_{x_j} \prod(X_{ij}|I_j))$$

Graphically, the possibility measures are superimposed and the minimum taken at every point. In Figure 3-1, for each rule in the sample knowledge base, this operation is represented by a dashed horizontal line. The height of this line is the alpha value for that rule.

After the alpha value for each rule R_i has been calculated, the inferencing engine uses that value to clip the membership function of the reference fuzzy set for the controller output associated with that rule, μ_{Y_i}. Once this has been done, the union of the clipped membership functions is performed. If a membership function μ_{A_i} is equal to α_i for all y_i and the pointwise disjunction operator is $\bigvee$, then

$$\mu_Y = \bigvee_{y_i \forall R_i} (\bigwedge_{y_i}(\mu_{A_i}, \mu_{Y_i}))$$

Graphically, α_i and Y_i are superimposed and the minimum taken at every point. This is done for all rules and the results are superimposed again, but this time the pointwise maximum is taken. In Figure 3-1, the horizontal dashed line clips the output membership functions Y_i; this clipped function is represented by the shaded area of the Y_i. The final fuzzy set for the controller output is indicated in the lower right portion of the figure.

Once the inference engine has done its work, the fuzzy set for controller output needs to be converted back into a crisp value to be passed on to the actuators. Two of the most popular are the centroid method and the height method. The former method simply takes the center of mass of the fuzzy set Y and returns that value as the controller output y. The latter method uses α_i to scale the center of mass for Y_i. These scaled centers of mass are then summed and divided by the sum of the α values, resulting in controller output y.

To make encoding of the rules simpler, a linguistic label is given to each reference fuzzy set X_{ij}, and the membership functions of the set are stored in the knowledge base. Rules are then written referring to the reference sets by their labels.

3.2 Mode Selector Implementation

Figure 3-2 shows the conventional low level controllers and their relation to the fuzzy logic mode selector. Three key engine parameters were chosen upon which the possible modes of engine operation were defined.

In addition, to simplify matters, five of the low level controllers were disabled. These are shown in the figure as empty boxes.

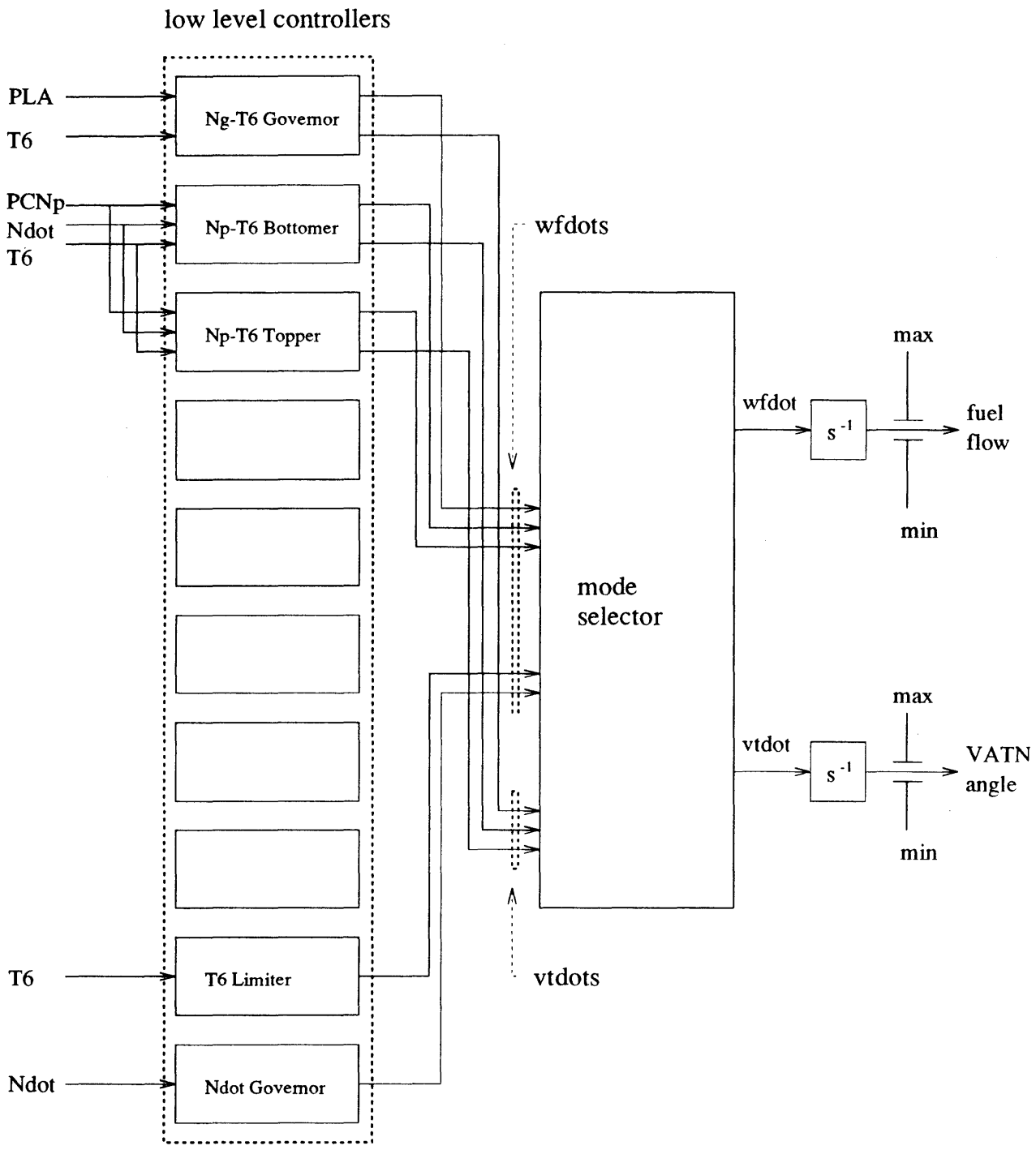

Figure 3-2: Conventional low level controllers and the fuzzy logic mode selector.

Reference fuzzy sets were defined for the three engine signals. The recuperator inlet temperature (T6) was divided into two regions that were assigned the linguistic labels *normal* and *high*; the power turbine speed (PCNp) into three regions: *low*, *normal*, and *high*; and the core acceleration (Ndot) into four regions: *positive high*, *positive low*, *zero*, and *negative*. Thus, twenty-four modes of engine operation were identified.

In addition, more than one controller was permitted to be active, but individual controllers were given weights of zero if their control laws were not required. In this vein, the Ng-T6 governor has a nonzero weighting in every mode, because some response to the driver's commands is necessary. When the power turbine speed is too low, the Np-T6 bottomer has a nonzero weighting; a similar case holds for the Np-T6 topper when the speed is too high. The T6 limiter is dominant when the recuperator temperature is too high, and similarly for the Ndot governor when the core acceleration is too great.

The low level controllers themselves were not modified in any way. However, their outputs were limited in magnitude, because the controllers still output excessive rates. This made the removal of the Ng bottomer and Ng topper possible, since core speed did not undershoot or overshoot unacceptably.

The three SISO controllers used to back up the MIMO controllers— the Ng governor, the Np bottomer, and the Np topper— were also removed. Their function was performed by their MIMO counterparts, and their absence reduced the number of modes to consider, as well as the number of parameters to adjust in the selector.

The actual operation of the mode selector is somewhat similar to that of the fuzzy logic controller described in the previous section. The three signals are fuzzified and passed to the inference engine, which calculates the possibility measures and the alpha values for all twenty-four rules in the knowledge base. However, there are no fuzzy sets for the selector output. Instead, each rule has a set of controller weights associated with it; these weights are listed in Table 3.1. If a given rule fires, a weighted average of the low level controller fuel flow derivatives and another weighted average of the low level controller VATN angle derivatives are calculated. No VATN rates are available from the SISO controllers, so the VATN rates from the Ng-T6 governor are used in their place. If two or more rules fire, indicating that the engine is presently operating in two or more modes, another weighted average is performed on the derivatives, using the alpha values of the rules as weights. The final rates are then passed out of the mode selector.

This approach is mathematically equivalent to the height method if, instead, fuzzy sets having centers of mass equal to the values listed in Table 3.1 had been defined for the controller weights. However, the implementation is computationally less expensive than storing a number of linguistic sets, and tuning the rules becomes slightly less difficult.

T6	PCNp	Ndot	Ng-T6 governor	Np-T6 topper	Np-T6 bottomer	T6 limiter	Ndot governor
normal	normal	pos high	0.40				0.60
		pos low	1.00				
		zero	1.00				
		neg	1.00				
	low	pos high	0.26		0.14		0.60
		pos low	0.40		0.60		
		zero	0.30		0.70		
		neg	0.20		0.80		
	high	pos high	0.13	0.27			0.60
		pos low	0.33	0.67			
		zero	0.33	0.67			
		neg	min	min			
high	normal	pos high	0.06			0.34	0.60
		pos low	0.27			0.73	
		zero	0.65			0.35	
		neg	min			min	
	low	pos high	0.06		0.11	0.40	0.43
		pos low	0.10		0.20	0.70	
		zero	0.16		0.67	0.17	
		neg	0.16		0.67	0.17	
	high	pos high	0.09	0.19		0.12	0.60
		pos low	0.23	0.47		0.30	
		zero	0.26	0.48		0.26	
		neg	min	min		min	

Table 3.1: Controller weights for the twenty-four modes of the mode selector.

3.3 Tuning

A mode selector is employed to deal with the long-term behavior of a system. Dynamics are the province of the low level controller, and if unsatisfactory, the low level controllers must be adjusted. However, once this is accomplished and problems still persist, then the fault lies within the mode selector.

In this vein, some tuning of the selector was required to achieve the desired response. Repeated simulation runs were conducted with a long burst of power, where the engine was accelerated from minimum power to maximum power, kept at that maximum for a relatively long period of time, and then decelerated back to minimum power. This allowed a large number of the modes to be exercised at once, and if performance was unacceptable, controller weights were readjusted and the cycle repeated.

In Figure 3-3, some representative plots of PLA, power turbine speed, maximum T6, T6 and core acceleration are shown against time. The PCNp plot has been divided into three regions, each corresponding to one of the variable's three reference fuzzy sets. Similarly, T6 has been divided into its two regions, and Ndot into its four. Notice that, because the maximum T6 varies with time, the regions over which T6 is considered normal and high also vary.

When the engine is first started, the transmission is not engaged. Recuperator temperature is not near its upper limit; power turbine speed is at its lower limit, but slowly increasing; and core acceleration is almost zero. Thus, the engine is operating in mode normal/low/zero, written in the format of T6/PCNp/Ndot.

A few seconds into the test, however, the transmission comes on line, and power turbine speed drops below its minimum reference with the large increase in load. The Np-T6 bottomer attempts to bring the speed back up, increasing the fuel flow and slowly opening the VATN. The engine passes from normal/low/zero to normal/low/pos low as the bottomer makes its presence felt. Temperatures rise rapidly, with the T6 limiter attempting to prevent T6 from reaching its limit, as the engine goes through high/low/pos low, normal/low/zero, and normal/low/neg.

In half a minute, the turbine is back to a steady state point in mode normal/low/zero, but the driver calls for a large power increase by increasing the PLA. The Ng-T6 governor responds with a large fuel flow that the T6 limiter and Ndot governor attempt to reduce in mode high/normal/pos high. Passing through mode normal/normal/pos low, the power turbine speed nears its maximum reference, and the Np-T6 topper joins in the opposition in modes normal/high/pos low and high/normal/pos low. A much longer time is necessary before the engine drops back into a steady state mode of normal/high/zero.

Upon the driver's reduction of PLA, the Ng-T6 governor requests a large decrease in fuel. The minimum fuel flow schedule opposes the fuel flow decrease in order to avoid flame out. Core speed drops to its minimum, slowly followed by power turbine speed, as modes normal/high/neg and normal/normal/neg are exercised. As the latter reaches its minimum, the Np-T6 bottomer tries to oppose that action, causing temperatures to rise again, as modes normal/low/zero, normal/low/pos low, high/low/zero, and high/low/pos low are encountered. Finally, a steady state is achieved again in mode normal/low/zero.

It was discovered that a 2:1 ratio of bottoming or topping controller to Ng-T6 governor was necessary to keep the power turbine speed above or below its minimum or maximum reference.

Adjustment of the Ndot governor controller was accomplished by observing core acceleration as the engine was accelerated, while tuning of the T6 limiter was more problematic, as the engine always overtemperatured on transients. Finally, the T6 limiter was replaced by a fuzzy logic proportional-integral controller, remedying the difficulty.[3]

For modes normal/high/neg, high/normal/neg, and high/high/neg, it was decided that the minimum controller fuel flow derivative and the maximum VATN angle derivative should be taken, instead of performing a weighted average. In all cases, this was to ensure good engine performance on large decelerations.

[3] Fuzzy logic low level controllers will be discussed in Chapter 4.

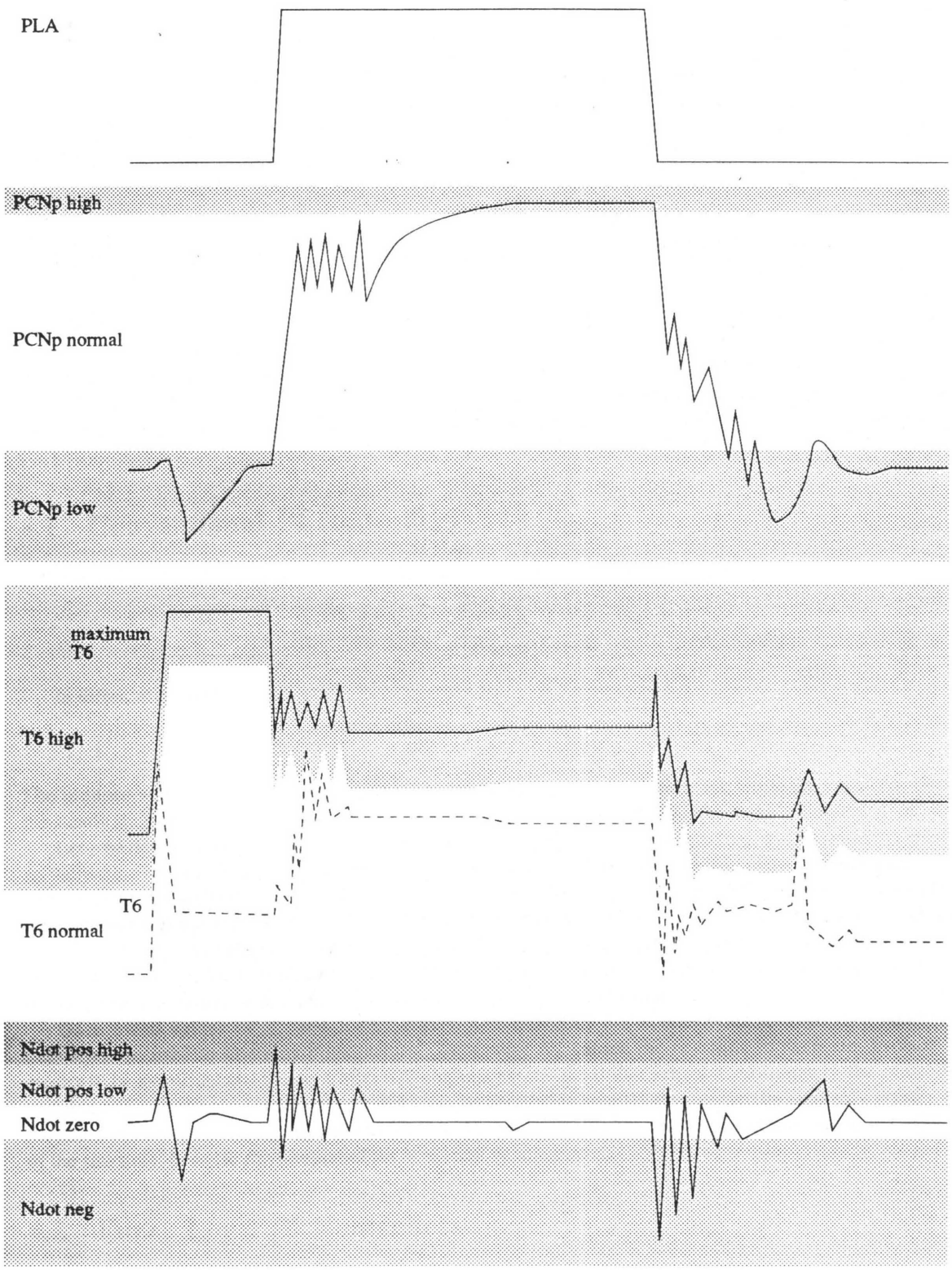

Figure 3-3: The long burst test: PLA, power turbine speed, maximum T6, T6, and Ndot, all plotted against time.

Chapter 4

Fuzzy Logic Low Level Controllers

In this chapter, the operation and tuning of conventional proportional-integral (PI) controllers are discussed. Then sliding mode controllers are presented, and parallels drawn between two dimensional sliding mode controllers and fuzzy logic PIs. Finally, the design and implementation of fuzzy logic PIs to replace the conventional low-level controllers is examined.

4.1 Conventional PI controllers

A conventional proportional-integral controller can be described by the function

$$\begin{aligned} u &= K_p e + K_i \int e dt \\ &= \int (K_p \dot{e} + K_i e) dt \end{aligned}$$

In differential form,

$$du = (K_p \dot{e} + K_i e) dt$$

The proportional term provides control action equal to some multiple of the error, while the integrator forces the steady state error to zero. Otherwise, the controller will always force a change in the manipulated variable.

Let e be defined as the set point subtracted from the actual value of a given signal, and let positive $\dot{e}$ denote an increasing rate of change of e. Assume a control law that requires a high positive du to counteract a high negative e with a high negative $\dot{e}$ and a high negative du to counteract a high positive e with a high positive $\dot{e}$. Also assume that the goal of the control law is to bring the system to the equilibrium point of zero e and zero $\dot{e}$. In a three dimensional space with axes e, $\dot{e}$, and du, the control surface du of a conventional PI would be a plane passing through the origin and oriented at some angle with respect to the e-$\dot{e}$ plane, the angle determined by the particular values of K_p and K_i, as shown in the top third of Figure 4-1.

Once initial values of K_p and K_i have been determined by the Zeigler-Nichols method, a number of heuristics are used to fine tune those values. Increasing K_p causes the rise time to decrease, because the error will be amplified and the controller will output a greater controller action. Unfortunately, the controlled variable will overshoot its steady state value, and the oscillation about that value will be markedly greater, for the same reason. Decreasing K_i will also reduce the overshoot of the controlled variable, but at the expense of the rise time, because the integral of the error will be attenuated.

4.2 Fuzzy Logic PI Controllers

Using fuzzy logic, a step-like control surface with gradations between the steps can be synthesized that approximates the control surface of the conventional PI, as can be seen in the middle third of Figure 4-1. Reference fuzzy sets are defined for e, $\dot{e}$, and du. In a collection of rules termed a *control matrix*, a distribution for the controller output du is defined for each combination of the linguistic sets for e and $\dot{e}$. In the figure, e has been

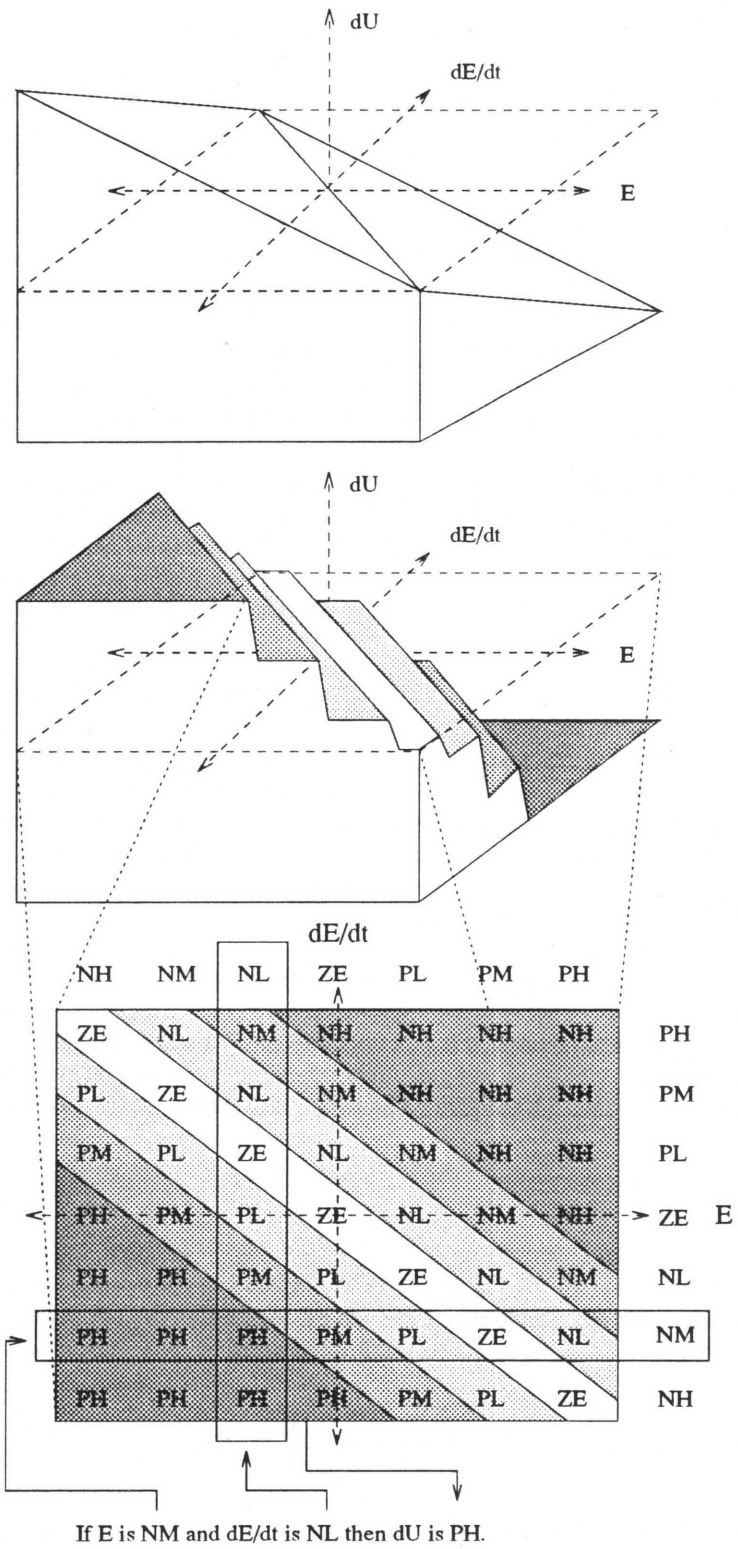

Figure 4-1: The control surface of a conventional PI controller, the surface for a fuzzy logic PI controller, and the control matrix for the fuzzy logic controller derived from that surface.

divided into seven fuzzy sets; PH is positive high, PM positive medium, PL positive low, ZE zero, NL negative low, NM negative medium, and NH negative high. $\dot{e}$ has also been divided into the reference fuzzy sets with the same linguistic labels, but it is important to note that e and $\dot{e}$ are not defined over the same universe of discourse, so the membership functions for the reference sets are not necessarily identical. The rules are rather straightforward. For example, if e has a negative medium value and $\dot{e}$ has a negative low value, then the error is slowly increasing. Thus, the appropriate control action would be as positive high increase in u. If the membership functions for e and $\dot{e}$ are properly defined so that they overlap by a certain percentage,[1] and if either of or both e and $\dot{e}$ happen to fall into the overlapping area, two or more rules will fire. Because of the inferencing mechanism discussed in Section 3.1, the controller output du will be an interpolation of the du values for each firing rule. This results in the gradations in the surface.

This control matrix and the associated reference fuzzy sets then constitute the contents of the knowledge base for the fuzzy logic PI. The fuzzy logic analogue of K_p and K_i can be found in the membership functions of the linguistic sets for e and $\dot{e}$. By increasing the range over which the membership functions for the medium, low, and zero magnitude $\dot{e}$ have nonzero values, K_p is decreased. Likewise, by increasing the range over which the membership functions for the medium, low, and zero magnitude e have nonzero values, K_i is decreased.

Small variations in error or the error derivative have less effect on the fuzzy logic PI. In the conventional PI, perturbations in e and $\dot{e}$ in the direction of or against the gradient of the control surface would cause du to change greatly. Examining the control surface of the fuzzy logic PI reveals that it is mostly parallel to the e-$\dot{e}$ plane. Only if the perturbation forced e and $\dot{e}$ into a transitional region where two or more rules would fire would the perturbations have any effect on the controller output.

4.3 Sliding Mode Controllers

The first half of this section is condensed from [Slotine]. The second half concerning fuzzy logic PIs and their relationship to sliding mode controllers can be found in [Palm].

For second order systems, the problem of forcing state vector $\vec{x} = [x\ \dot{x}]$ to track a desired vector $\vec{x_d} = [x_d\ \dot{x_d}]$ can be reduced to the problem of keeping the function

$$s = \dot{e} + \lambda e$$

as close to zero as possible, where e is the tracking error $x - x_d$, $\dot{e}$ is the tracking error derivative $\dot{x} - \dot{x_d}$, and λ is some problem-specific constant.

In two dimensions, the line defined by the equation $s = 0$ is termed the *switching line*. A sliding mode controller attempts to drive the error vector onto the switching line as rapidly as possible, and then force it to the equilibrium point $[e\ \dot{e}] = [0\ 0]$. This is accomplished by defining the control law u as follows:

$$u(s) = \begin{cases} +K & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -K & \text{if } s < 0 \end{cases}$$

However, the discontinuity at the switching line $s = 0$ makes for extremely high control action if the system does not settle onto the switching line. To remedy this problem, the discontinuity can be smoothed out by a gradation in the region $|s| \leq \Phi$, so the control law becomes

$$u(s) = \begin{cases} +K & \text{if } s > \Phi \\ +K\frac{s}{\Phi} & \text{if } |s| \leq \Phi \\ -K & \text{if } s < -\Phi \end{cases}$$

However, the sliding mode controller is linear in the region close to the switching line. For the fuzzy logic PI, the controller output du can be given exponential gains in the region $|s| \leq \Phi$. For instance, if the membership functions for low magnitude du had their centers of mass a distance d away from the origin, the

[1] A rule of thumb states that the amount of overlap between "properly defined" adjacent membership functions is around twenty-five percent.

membership functions for medium magnitude could have centers of mass $2d$ away from the origin, and those for high magnitude $4d$ away. This would cause the state vector to approach the switching line faster, reducing rise time.

Settling time can also be reduced by placing a deadband around the switching line close to the equilibrium point. This is done by defining the membership functions for positive low and negative low error so that they stop some small distance away from the point at which the error is zero. The same is done for the error derivative. Thus, when the equilibrium point is neared, there is no change in controller output.

4.4 Implementation

The replacement of the conventional low level controllers necessitated some rethinking about control of the VATN. Finally, it was decided to create a VATN governor and employ mode selection on only the fuel flow. As shown in Figure 4-2, six fuzzy logic PIs were created, five to generate fuel flow rates, and one to handle the VATN.

The Ng and VATN governors are used for nominal control. The Ng governor employs PLA as a reference, as does its conventional counterpart, the Ng-T6 governor. Its control matrix, as seen in the upper left of Figure 4-3, is similar to that of the standard fuzzy logic PI, except where the magnitude of the error between PLA and PCNg is too high. For positive errors, when the actual value is above its reference, a large negative action for all values of error derivative is necessary to improve response; for negative errors, a large positive action is required instead. This obliterates the switching line far away from the equilibrium point. If the plant was quick to respond to great changes in actuator command, the system would limit cycle between positive error and negative error. Fortunately, this condition does not hold.

The VATN governor manipulates VATN angle to control T6. A T6 schedule determines the optimal value of recuperator temperature; if the actual temperature is too high, the VATN is opened to reduce that temperature. Thus, the governor's control matrix is identical in form to that of the Ng governor, with the exception that positive control actions are exchanged for negative ones, and vice versa.

The Np bottomer and the Np topper, in the lower left and lower right of Figure 4-3, have the same purposes as their conventional equivalents. A large number of the rules of the standard fuzzy logic PI are not used, so if no rules fire, the controller is not active. In the case of the Np bottomer, if the power turbine speed is above its minimum reference, the controller should not be active. Also, a negative control action is not desirable, since that would decrease fuel flow and reduce power turbine speed. For large negative errors, when the speed is much below its minimum reference, a large control action is preferred, improving bottomer response. A similar case can be made for the Np topper, reversing the explanation where appropriate.

Both the Ndot and T6 limiters attempt to keep an engine parameter below a maximum value, so their control matrices, as shown in the upper right of Figure 4-3, are a close match to that of the Np topper. However, because both Ndot and T6 react quickly to changes in fuel flow, a number of zeros for du are included where there are no rules in the Np topper. Whenever these limiters are active in a given mode, the zeros are used in the weighted average, acting to prevent positive changes in fuel flow from increasing Ndot and T6 too rapidly.

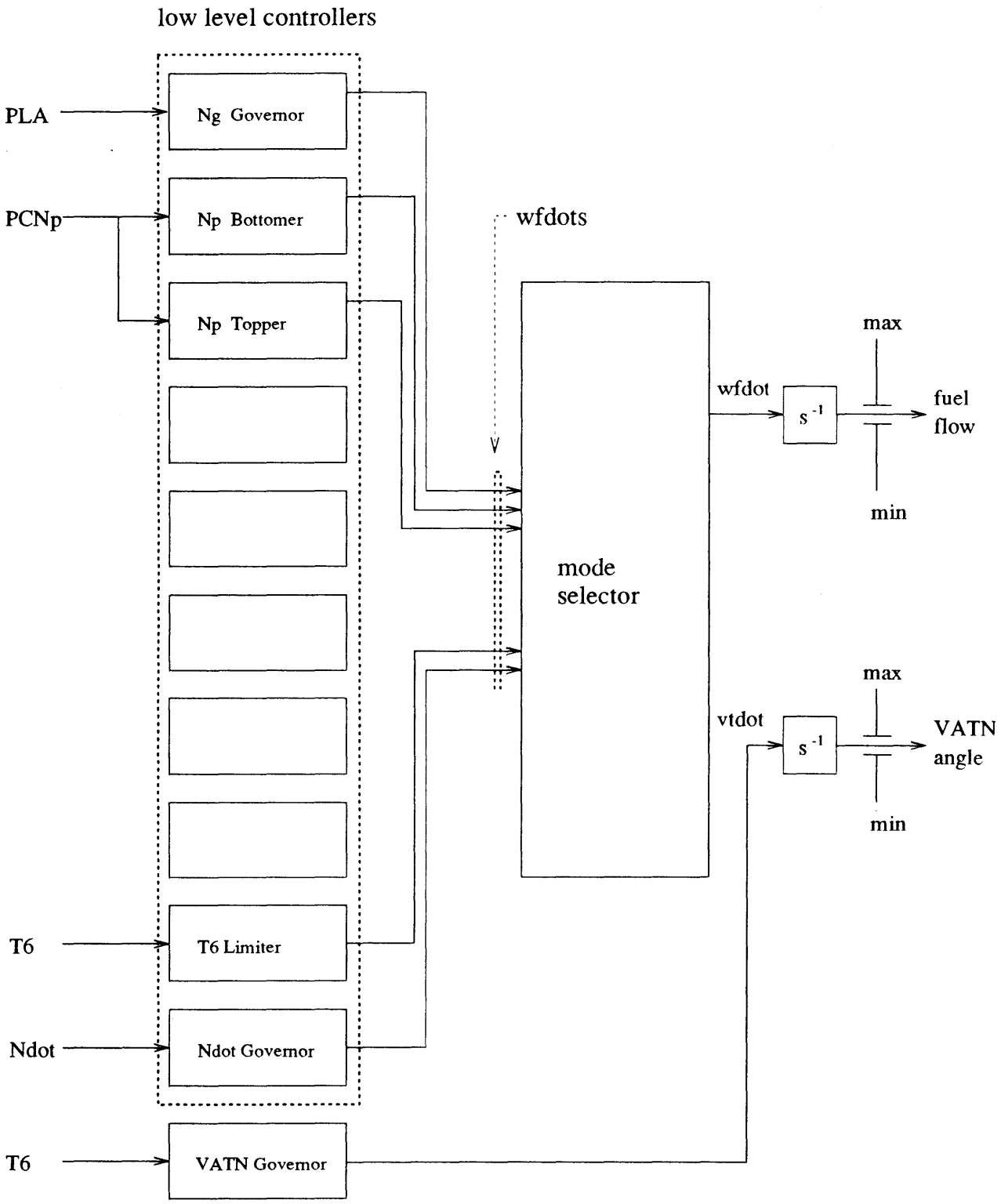

Figure 4-2: The fuzzy mode selector with fuzzy logic low level controllers.

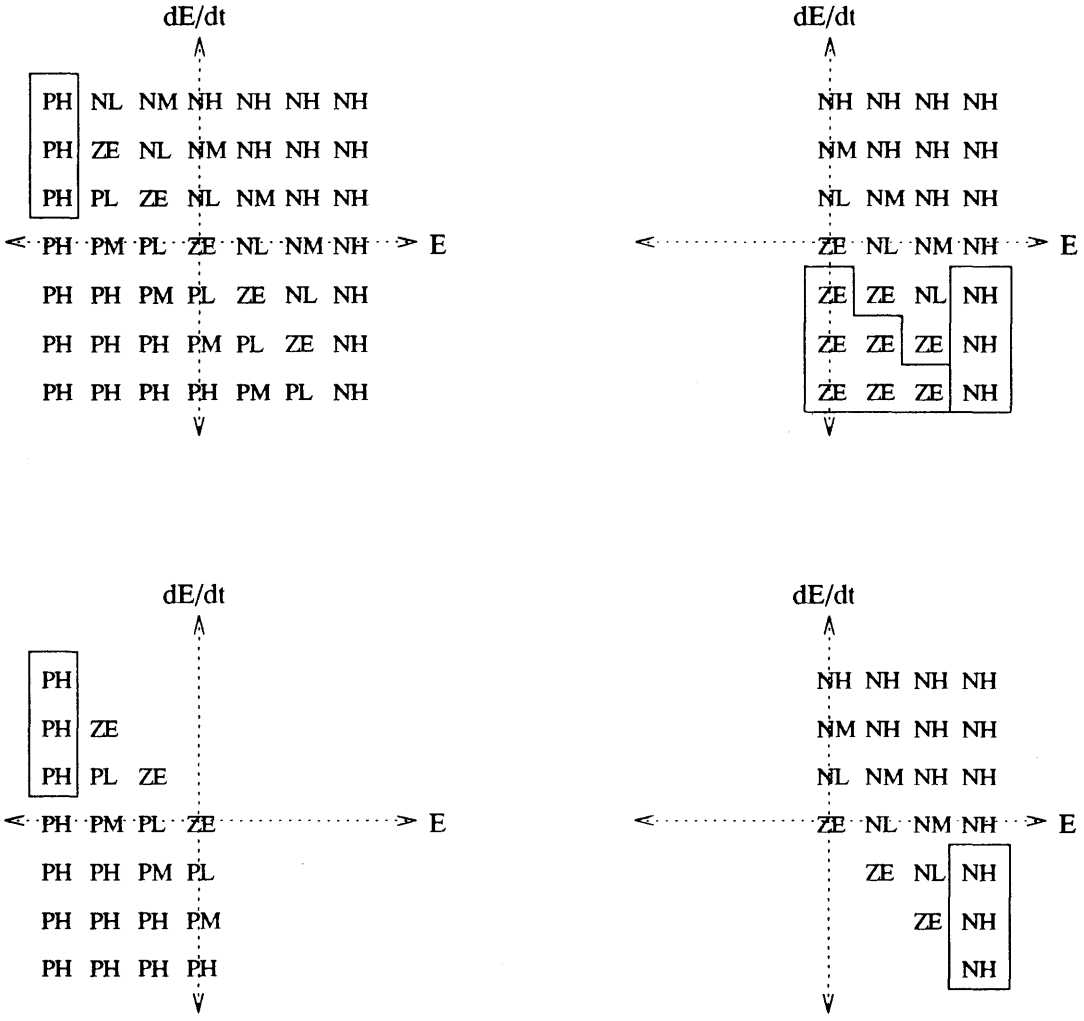

Figure 4-3: Clockwise from the upper left: control matrices for the Ng governor, the T6 and Ndot limiters, the Np topper, and the Np bottomer.

Chapter 5

Data Analysis

For the most part, the functionality of the original control software was maintained, except in three major places. The lead lag compensator for the T6 sensor was adjusted, reducing the accuracy of the compensated value in exchange for decreased lag time.

The other two changes were made to handle the fuzzy logic low level controllers— the minimum fuel schedule was increased and the minimum VATN angle decreased. The first was done to prevent a temperature spike on deceleration. As core speed drops to its minimum value, the Ng governor outputs a negative fuel flow, but when core speed drops below that minimum, the governor's output becomes positive and the engine temperature increases greatly. Increasing the minimum fuel flow schedule was as a way to keep the governor's action negative by forcing the core speed to remain above its minimum as much as possible. Unfortunately, this increased the time required for deceleration. The second modification was made to allow the VATN governor the full range of actuator command.

Two types of tests were employed to gauge engine performance. The first controlled the engine by varying PLA. The long burst test, a long burst of maximum power from idle, followed by a cut back to minimum power, was of the first type. The second used a fuzzy logic PI controller to emulate a human tank driver. Three actual mission profiles, detailing the particular velocities the vehicle should have at given times, were used. [1]

In the long burst test, the conventional control was unable to reach maximum power turbine speed at all during the burst, as shown in Figure B-1. Examining the actuator commands, the VATN was found to be saturated. In addition, T6 did not reach its optimal value as shown in Figure B-4, and consequently the thermodynamic cycle was less efficient. Tests of the fuzzy logic mode selector with and without fuzzy logic low level controllers did not exhibit this behavior. The loss of the Ng topper and Ng bottomer was thought to be responsible for this difference, so both controllers were disabled in the conventional control scheme and the test rerun. No large discrepancies were noted.

The fuzzy logic mode selector, with and without the fuzzy logic PIs, fared better than the conventional control. Engine speeds had comparable rise and fall times, as can be seen in Figure B-2 and Figure B-3. Investigating the temperature curves in Figure B-5 and Figure B-6 shows that, with the fuzzy logic PIs and the decoupling of the VATN, the cycle is more efficient because T6 remains close to its optimal value throughout the burst.

Once all three types of control were working, the entire engine was treated as the plant, with PLA as the input and vehicle velocity as the output. A fuzzy logic PI controller was used to close the loop, attempting to control the vehicle to a certain velocity. Because velocity is also a function of the transmission, there were two additional inputs: a selector, which the driver used to shift into forward, reverse, and neutral; and a set of brakes, which the driver could apply with varying force. Mission profile data, including desired velocities, selector settings, and braking rates, were fed into the driver PI.

The power plant, operating under control of the fuzzy logic mode selector, gave the desired velocities sooner than conventional control, resulting in an increase in bandwidth. When the fuzzy logic mode selector was used with conventional low level controllers, vehicle speeds sometimes exhibited greater underdamped action. Replacing the conventional low level controllers with fuzzy logic ones did alleviate this problem somewhat.

[1] The actual plots of the data can be found in Appendix B.

In addition, the mode selector prevented the VATN from saturating, which led to greater fuel economy, since the engine more often than not operated at the optimal T6 value. Integrating the fuel flow curves revealed a decrease in fuel consumption for the fuzzy logic mode selector. The selector operating with conventional low level controllers achieved a fuel savings of 5.06% and 7.59% over the conventional means of control when testing the first and second mission profiles, while 5.61%, 7.62%, and 2.52% were saved with fuzzy logic PI controllers when running the first, second, and third profiles. The additional savings, realized by the fuzzy logic PIs over that of the conventional low level controllers, was due to the decoupling of the VATN from the fuel flow.

Chapter 6

Conclusions and Future Directions

In this line of investigation, hierarchical control of a recuperative turboshaft has been examined. Using fuzzy logic to implement a mode selector provides a number of advantages:

- Instead of having only a single controller active at a given time, modes can be defined with many controllers active at once. When switching is performed between two or more modes, the inferencing method of fuzzy logic interpolates between the control actions for those modes, resulting in smoother transitions.
- The action of the mode selector is more readily apparent because the dominant mode or modes can be found by examining a set of engine parameters, whereas the min/max ladder requires the low level controllers themselves to output excessive rates so that those controllers will not be selected.
- Fuzzy logic mode selection allows this knowledge to be moved up into the mode selector, so that the low level controllers handle the dynamics, while the higher level mode selector deals with quasi-steady state conditions, thus making the design of low level controllers easier.

The results of employing the mode selector are as follows:

- The response of the plant has been improved with respect to the conventional control scheme.
- Fuel consumption has been lowered with respect to the conventional control scheme.
- A further fuel savings can be realized by replacing the conventional controllers with fuzzy logic PIs and decoupling the actuator actions.

Unfortunately, both the improved response and reduction in fuel consumption were probably obtained at the expense of stall margin. Also, because the dynamics of the plant were not fully explored, a number of low level controllers, omitted to simplify the construction of the mode selector, may actually be necessary. Further research in these areas is required before any additional conclusions can be drawn.

On the whole though, fuzzy logic and fuzzy logic mode selection seems to be experimentally tractable, providing performance comparable to that of the conventional control scheme. However, there are some key points to be addressed regarding the design of such controllers, their theoretical basis, and their stability.

In the design of a conventional PI controller, there are a number of parameters available for adjustment, namely the integrator time constant, and the input and output gain vectors. The designer of a typical fuzzy logic PI controller has to determine, in addition to input and output scaling factors, the desired membership functions and rules; there are more degrees of freedom from which to choose.[1]

Theoretically, it is possible to create an analogue of a fuzzy logic controller using boolean logic. This can be accomplished by writing boolean rules to approximate the control surface of the fuzzy logic controller; these rules would associate a particular set of controller inputs with a specific value for the controller output.

[1] A number of design techniques now exist, although a bibliography of those techniques was not available at the time this thesis was printed.

However, since the rules are boolean, there must be an exact matching between the actual input and those values specified by the rules for the rules to fire. This necessitates a large number of boolean rules to approximate the surface. Ranges of boolean values can be used to greatly reduce that set of rules. However, the controller will still exhibits great sensitivity around the edges of those ranges because of discontinuities. Interpolating between the ranges and smoothing the controller output gives better performance. But this entire mechanism already exists— it is in essence a fuzzy logic controller.

The stability of the fuzzy logic controller has been noted in the literature to be similar to that of a sliding mode controller, yet the issue of stability for the combined fuzzy logic mode selector and low level controllers has not been addressed. Regardless of the type of mode selector, be it conventional or fuzzy logic based, currently there is no way to determine the stability of the resulting system, except for verification by simulation. Although analytical linear models of the plant could be constructed, they would only be valid for the operating points at which those models were constructed. Analysis of the closed loop system with its mode selector is only possible when the low level controllers and the mode selector are also linearized, giving a characteristic polynomial with input-dependent coefficients. In short, the task is daunting; verification by simulation represents a cost-effective way with which to attack the problem.

Appendix A

Nomenclature

Some common abbreviations are listed below.

Ng	high pressure spool speed, core speed, gas generator speed
Np	free power turbine speed, power turbine speed
PCNg	percent core speed
PCNp	percent power turbine speed
PLA	power lever angle, throttle setting
T6	recuperator inlet temperature
VATN	variable area turbine nozzle
vtdot	VATN derivative
wfdot	fuel flow derivative

Appendix B

Data

Two component level models were actually used. One, designated as the E2A cycle, was significantly easier to control than the other, the E2 cycle, because the thermodynamic constraints of the E2A were not as tight as the E2.

Most development was done on the E2A cycle, until it was discovered that the Newton-Raphson method would not converge fast enough whenever a deceleration of the engine was requested. Moreover, the E2 cycle did not have this problem in real time mode, so it was decided to transition work to that model. All the data given in this appendix was generated from simulations of the E2 cycle.

This appendix gives the curves for the long burst test, as well as the three mission profiles. Figures B-1 - B-6 give the results of the long burst test. The first three are of PLA, core speed, and power turbine speed, all plotted against time. The last three are of recuperator inlet temperature, optimal recuperator temperature, and recuperator temperature limit, again all plotted against time. Because the actual values of speed and temperature are information proprietary to the General Electric Company, only percent maximum speed is available for the first three figures, and no scale is included for the last three.

As discussed in Section 3.3, the long burst test would begin with an idling of the engine. PCNg and PCNp should start out at their minimum values of 62% and 30%. Five seconds into the test, the transmission would be engaged, and PCNp should drop in response to the increased load. This drop should be as little as possible. Thirty seconds into the test, maximum power would be called for by a maximum PLA of 100%, and the engine should produce a maximum PCNp of 100% as rapidly as possible without violating the maximum PCNg of 95%. Ideally, there should be at most five spikes in PCNp as the transmission shifts from first gear to sixth gear. Four hundred seconds into the test, the engine would be cut back to idle speed, with PCNg and PCNp dropping back to their respective minimums. There should be at most five spikes in PCNp as the transmission shifts down from sixth gear back to first. PCNp would also undershoot, and once again this undershoot should be as little as possible. Once the transmission has returned to first gear, PCNg should rise to force PCNp back up to its minimum value. At six hundred seconds, the test should end, with PCNp at its minimum value and PCNg at such a point that it sustains PCNp at that value. Both speeds should exhibit little or no oscillation.

As for engine temperatures, T6 should remain below its maximum value at all times. Upon transients, spikes in T6 are allowed, but should be avoided if at all possible. In steady state, T6 should be equal to its optimal value, to make the cycle as efficient as possible.

Upon comparison of Figures B-1, B-2, and B-3, the most readily apparent observation is that the conventional control scheme does not cause PCNp to reach its maximum in the allotted 370 seconds. The gradual slope of the PCNp curve also confuses the automatic transmission enough that it shifts back and forth between its fifth and sixth gears repeatedly, more so than either of the other two schemes. However, the fuzzy logic mode selector and conventional low level controllers caused PCNg to overshoot its 95% maximum at around the forty second mark. In addition, PCNp undershot more upon the engagement of the transmission and upon the downshift from second to first gear. The fuzzy logic mode selector and fuzzy logic low level controllers seemed to perform comparably to the conventional scheme, although its overshoot at the 475 second mark was greater than that of the conventional.

Examination of Figures B-4, B-5, and B-6 revealed that the conventional control scheme did not reach optimal T6, whereas the fuzzy logic mode selector with conventional low level controllers did reach that value

briefly. The mode selector with fuzzy logic low level controllers tracked the optimal value for the most of the length of time at maximum power. In all three schemes, there were two spikes in T6 from the engaging of the transmission at the five second mark and the recovery from undershoot at the 475 second mark.

However, in Figure B-3 and Figure B-6, there were some oscillations in the curves after the engine had returned to a steady state from decelerating. This was due to a problem in tuning the Np bottomer. Although some additional effort could have remedied this problem, it was decided that the time would be better spent writing this thesis.

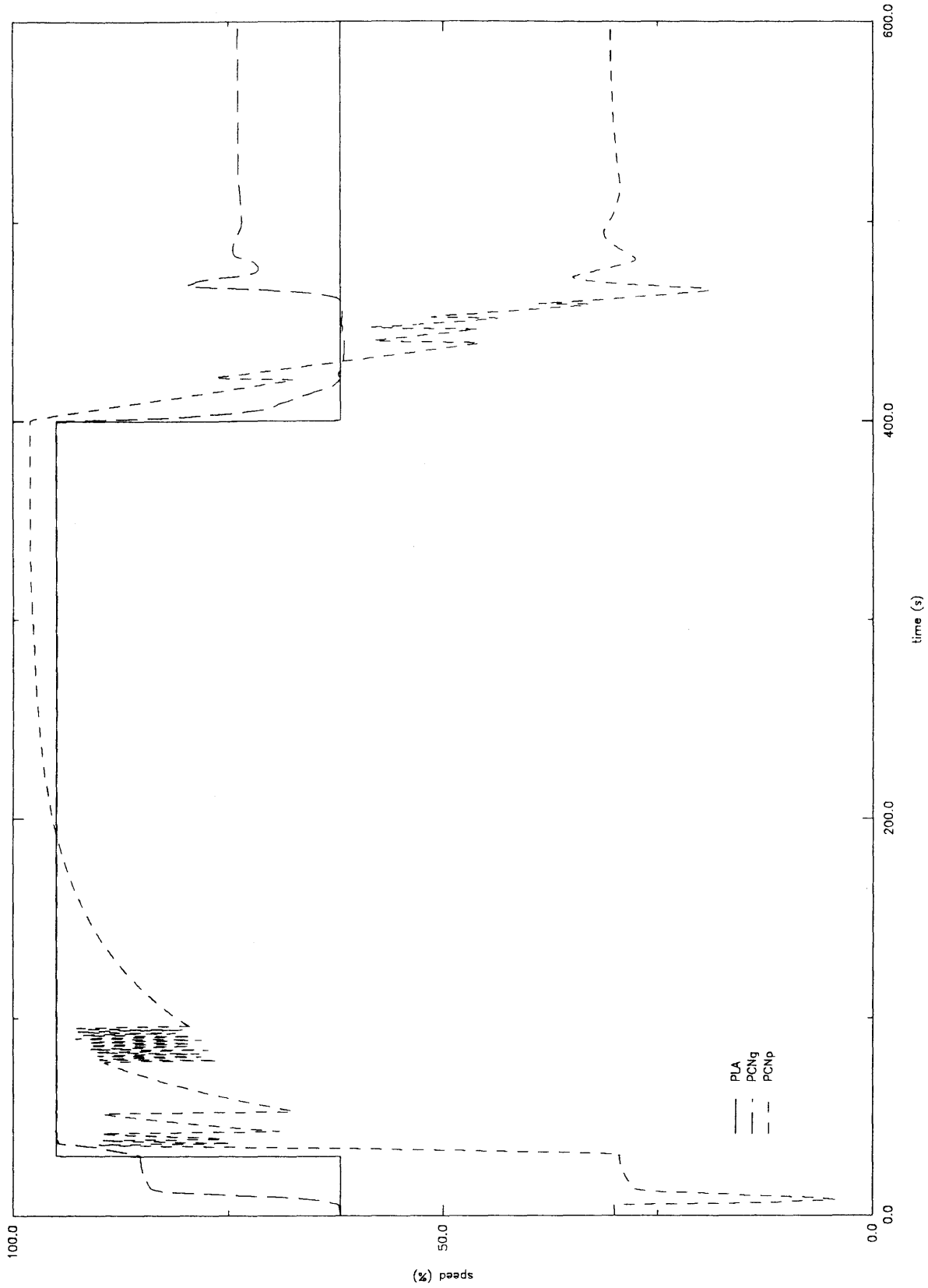

Figure B-1: Conventional controllers and min/max mode selection: PLA and engine speeds for long burst.

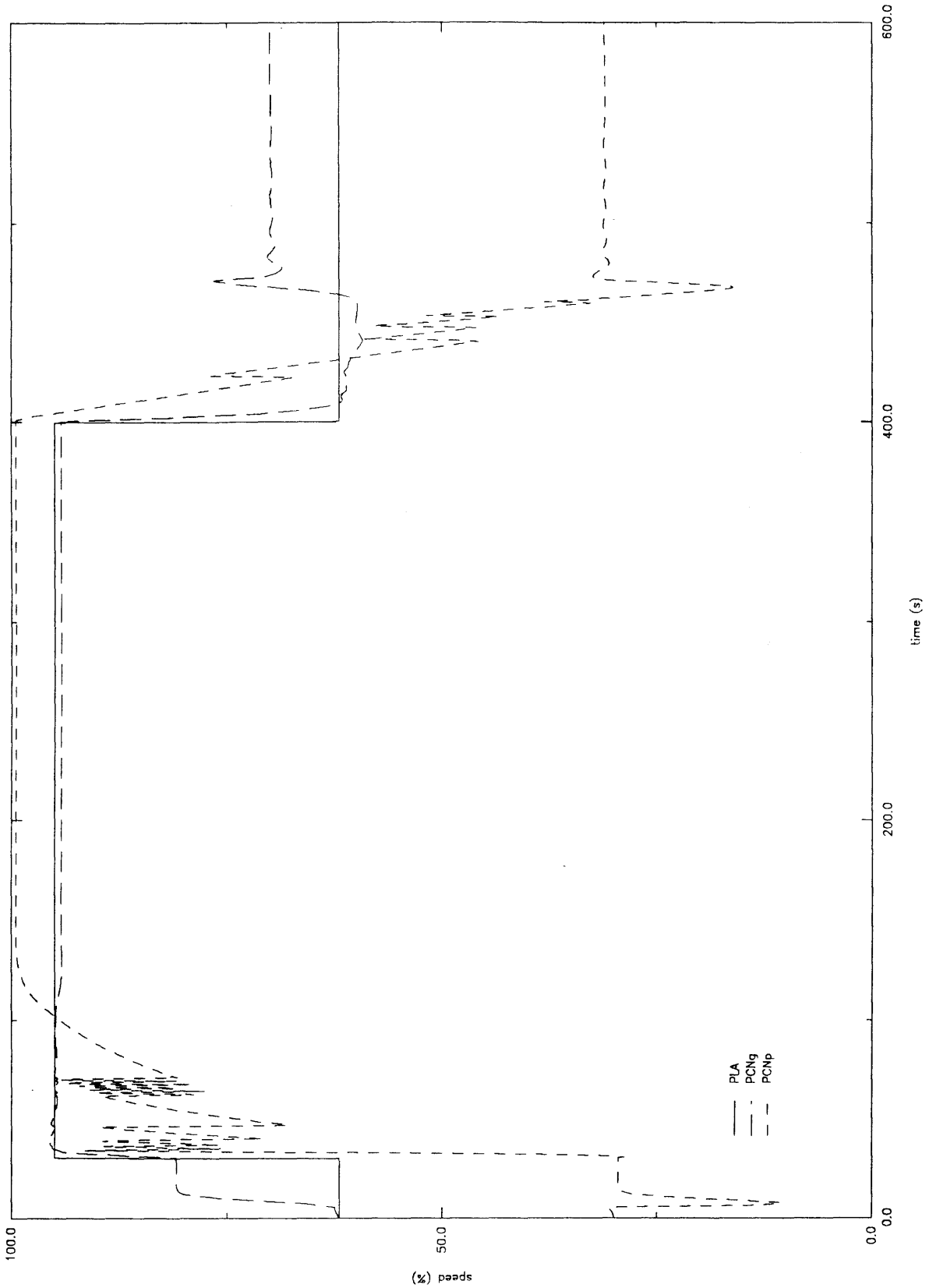

Figure B-2: Conventional controllers and fuzzy logic mode selection: PLA and engine speeds for long burst.

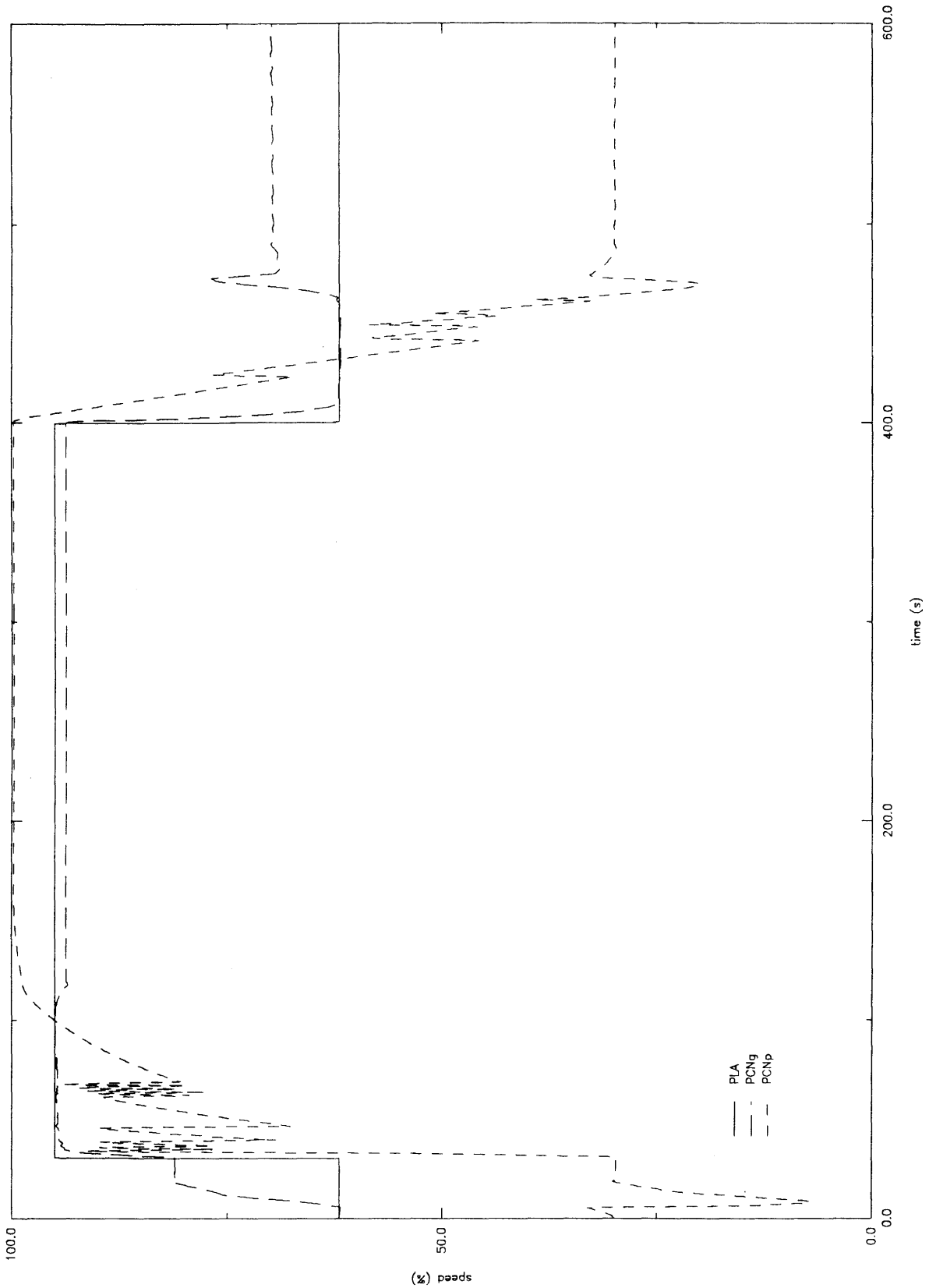

Figure B-3: Fuzzy logic controllers and fuzzy logic mode selection: PLA and engine speeds for long burst.

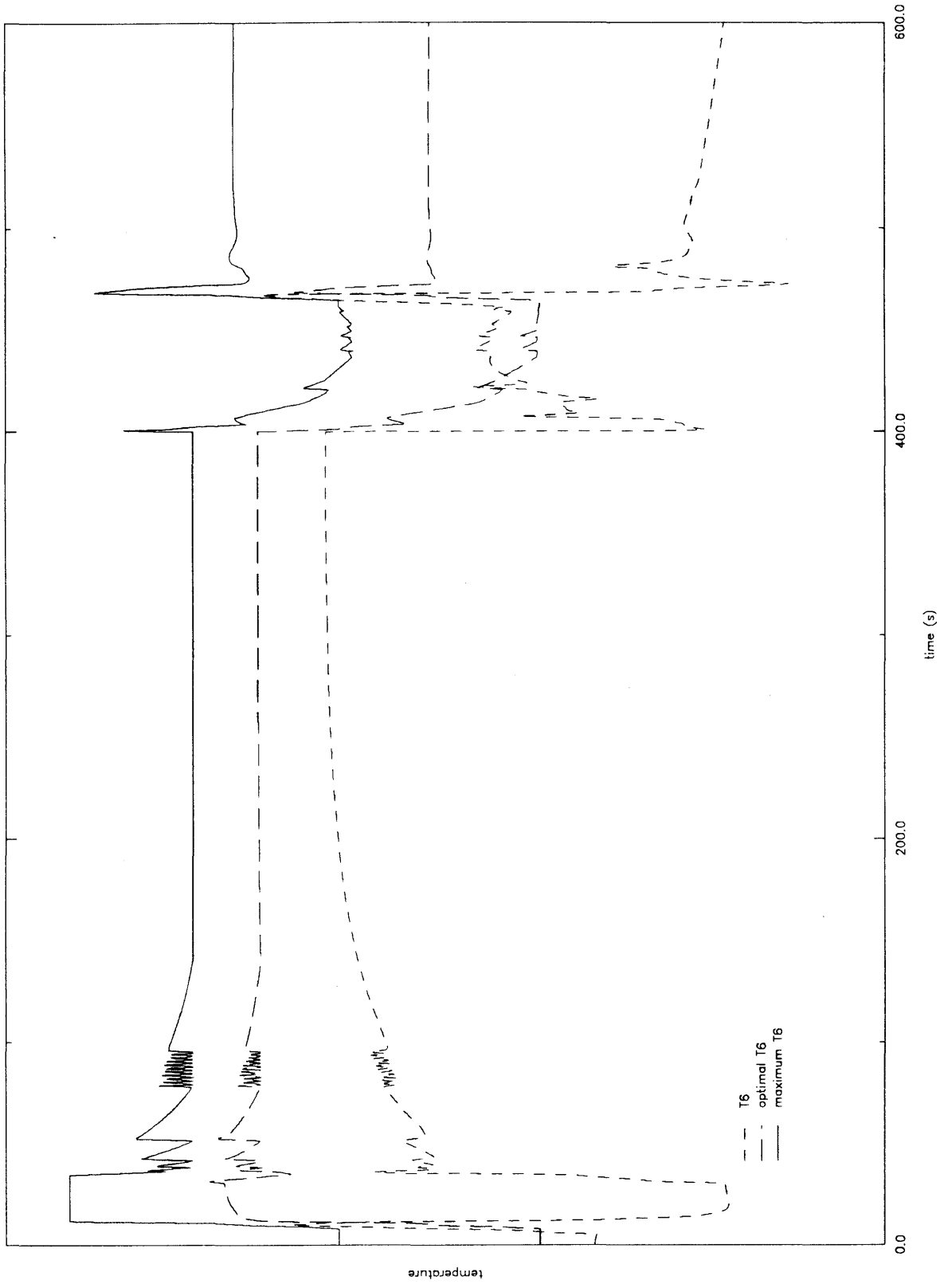

Figure B-4: Conventional controllers and min/max mode selection: recuperator temperature limit, optimal temperature, and actual temperature for long burst.

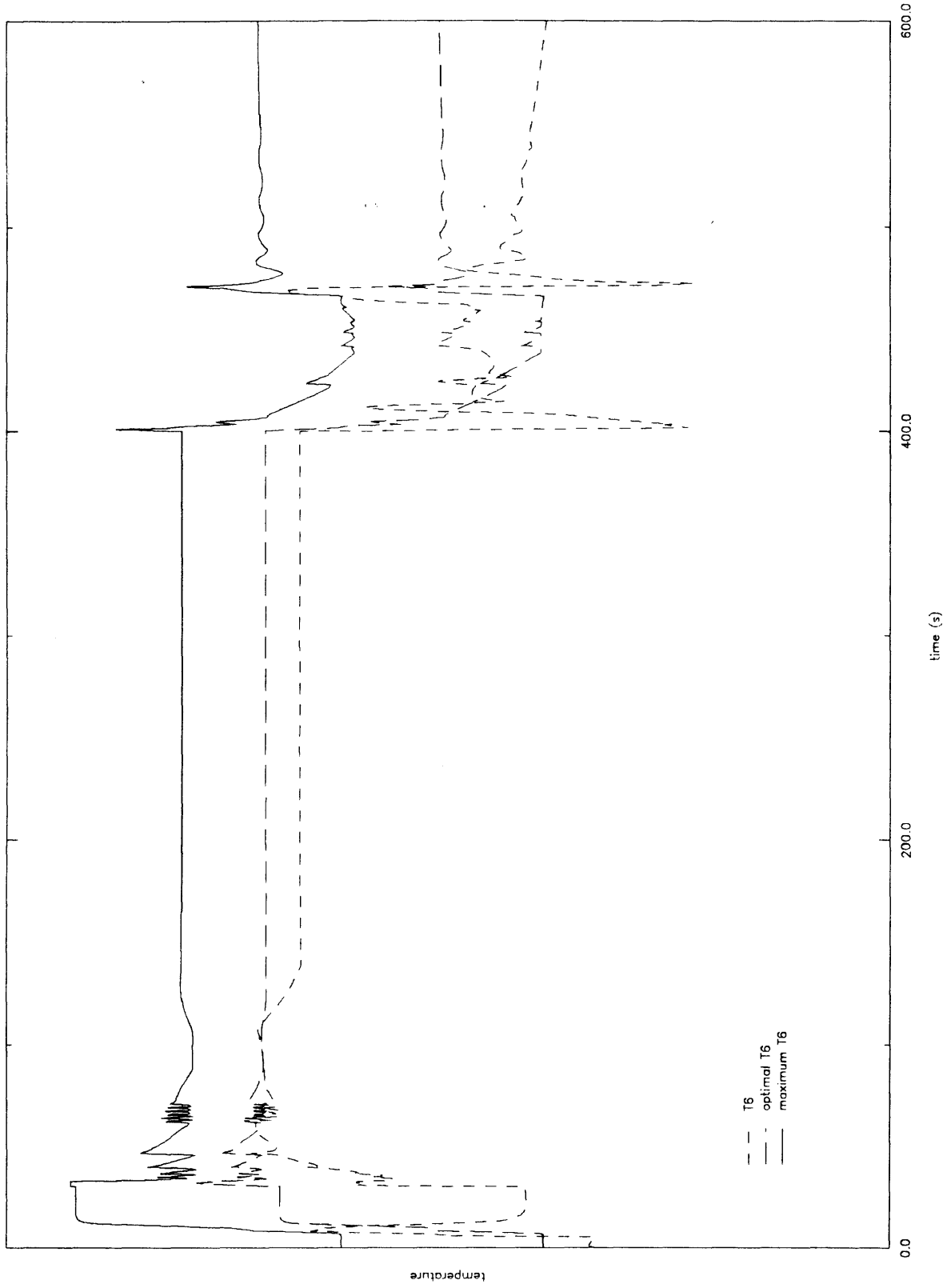

Figure B-5: Conventional controllers and fuzzy logic mode selection: recuperator temperature limit, optimal temperature, and actual temperature for long burst.

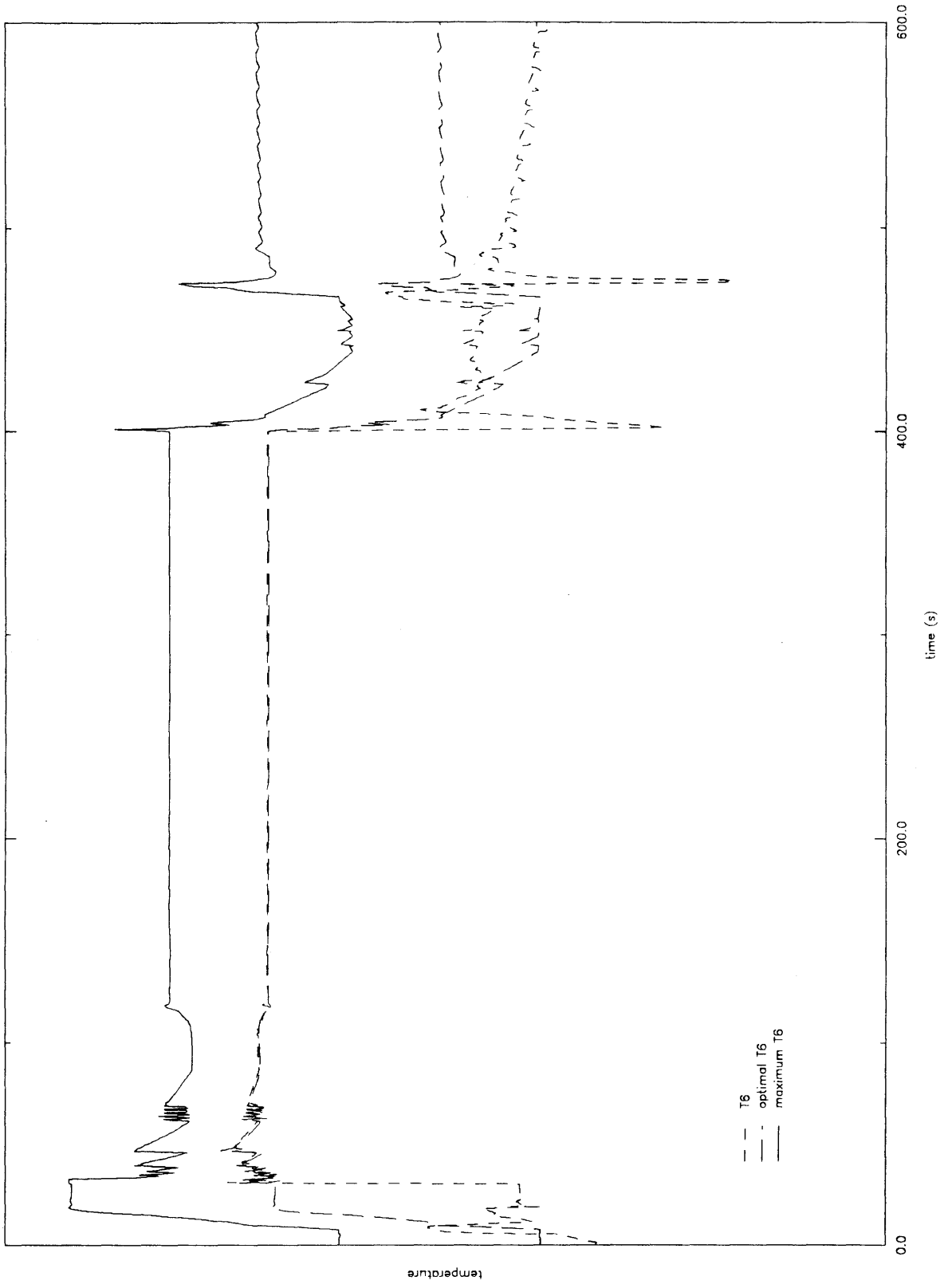

Figure B-6: Fuzzy logic controllers and fuzzy logic mode selection: recuperator temperature limit, optimal temperature, and actual temperature for long burst.

The three mission profiles are actual segments of a six hundred hour test that each engine is supposed to undergo. Each mission profile gives a desired vehicle speed, the position of the selector— neutral, forward, or reverse— and the braking rate, as can be seen in Tables B, B, and B. A blank entry in the selector column indicates that the selector is not changed from its previous setting. A blank entry in the braking column indicates that the brakes are not engaged. Accelerations would be accomplished with maximum power, and decelerations with minimum power. Ideally, the actual vehicle speed should be equal to the desired speed at all times.

As discussed in Chapter 5, a fuzzy logic PI to simulate a human driver was constructed to test out the three control schemes. Figures B-7 - B-9 concern the first mission profile, Figures B-10 - B-12 concern the second, and Figures B-13 - B-15 concern the last. Each figure has three curves on it— the desired velocity of the vehicle, its actual velocity, and the PLA. Once again, because the actual values are proprietary information, no scale has been included in any of these figures. Unfortunately, the lack of scale does not help in correlating the data in the tables with the figures. As points of reference, note that at $853s$ in the first mission profile the desired vehicle speed is 100% of maximum, at $265s$ in the second mission profile the desired speed is 80% of maximum, and at $955s$ the desired speed is 100% of maximum.

In the first mission profile, the conventional control scheme was unable to meet desired vehicle speeds from $577s$ to $613s$, from $853s$ to $877s$, and from $1015s$ to $1099s$. Both schemes employing the fuzzy logic mode selector were able to reach the desired velocities slightly faster than the purely conventional scheme. From $613s$ to $721s$, however, the fuzzy logic mode selector with conventional controllers exhibited oscillatory action.

In the second mission profile, all three schemes did not meet the desired velocities at times $205s$ to $223s$, $691s$ to $715s$, and $1213s$ to $1225s$. In addition, there was chattering in the vehicle speed from $265s$ to $361s$.

In the third mission profile, all three schemes did not achieve the desired speeds at times $343s$ to $397s$, $751s$ to $787s$, and $955s$ to $1045s$. In addition, the fuzzy logic mode selector with conventional low level controllers exhibited greatly underdamped behavior at intervals from $955s$ to $1045s$ and from $1063s$ to $1177s$. However, the data in Figure B-14 is invalid after time $685s$; at that point, the brakes were engaged, and the power turbine speed became negative.

The fuzzy logic PI for the tank driver was not very well tuned for all three types of control, as time constraints were once again a problem. The chattering in the PLA in the second mission profile was the result of the driver PI's attempts to deal with the shifting of the automatic transmission.

Comparisons of PLA for the engine after a desired velocity had been reached also reveal that PLA levels were lower when control was performed with the fuzzy logic mode selector, an indication of the greater efficiency of the cycle.

time (s)	% max mph	selector	% braking
0.	0.0	neutral	
115.	17.3	forward	
169.	28.0		
349.	58.7		
451.	0.0		50.0
469.	53.3		
529.	10.7		
553.	28.0		
577.	0.0		
583.	86.7		
613.	10.7		
721.	4.0		
745.	17.3		
853.	100.0		
877.	0.0		62.5
895.	8.08		
955.	17.3		
973.	0.0		
1015.	89.3		
1045.	0.0		
1099.	65.3		37.5
1117.	0.0		
1237.	0.0	neutral	50.0
1380.	end		

Table B.1: First mission profile: desired vehicle speed, selector setting, and braking as a function of time.

time (s)	% max mph	selector	% braking
0.	0.0	neutral	
91.	52.0	forward	
151.	0.0		50.0
169.	46.7		
205.	0.0		
223.	38.7		
265.	80.0		
361.	45.3		
673.	0.0		25.0
691.	70.7		
715.	0.0		75.0
745.	34.7		
763.	18.7		
823.	5.3	reverse	
841.	49.3	forward	
883.	0.0		75.0
901.	38.7		
1087.	30.7		
1213.	8.0		
1225.	0.0		
1243.	65.3		
1273.	0.0	neutral	
1440.	end		

Table B.2: Second mission profile: desired vehicle speed, selector setting, and braking as a function of time.

time (s)	% max mph	selector	% braking
0.	0.0	neutral	
91.	58.7	forward	
133.	69.3		
241.	72.0		
301.	58.7		
343.	80.0		
397.	46.7		
433.	0.0		75.0
451.	93.3		
685.	0.0		100.0
703.	41.3		
751.	82.6		
787.	22.7	reverse	
823.	24.0	forward	
895.	10.7		
955.	100.0		
1045.	0.0		25.0
1063.	13.3		
1177.	24.0		
1255.	0.0	neutral	
1273.	10.7	forward	
1285.	0.0	neutral	
1297.	10.7	forward	
1309.	0.0	neutral	
1620.	end		

Table B.3: Third mission profile: desired vehicle speed, selector setting, and braking as a function of time.

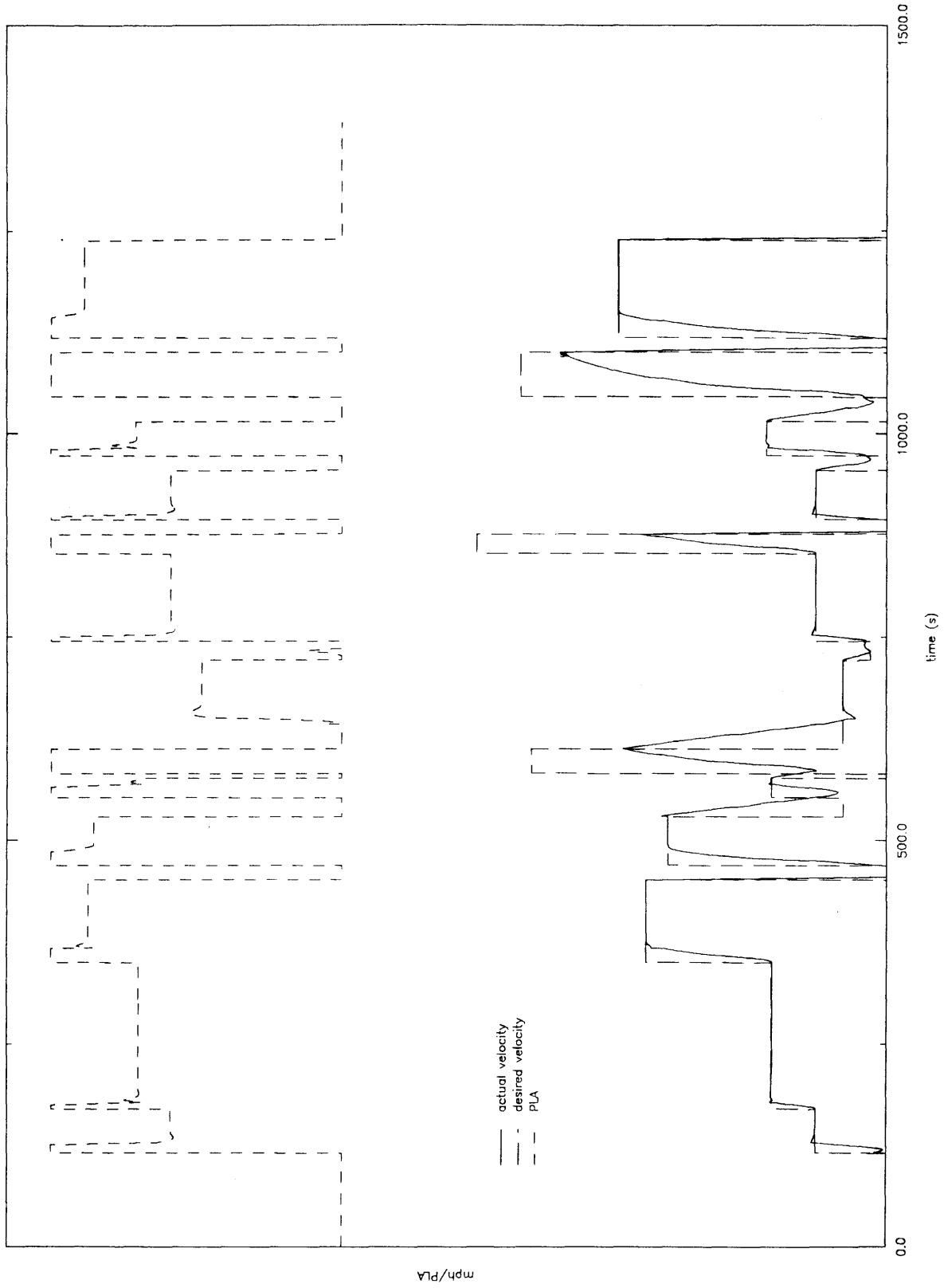

Figure B-7: Conventional controllers and min/max mode selection: actual vehicle velocity, desired velocity, and PLA for the first mission profile.

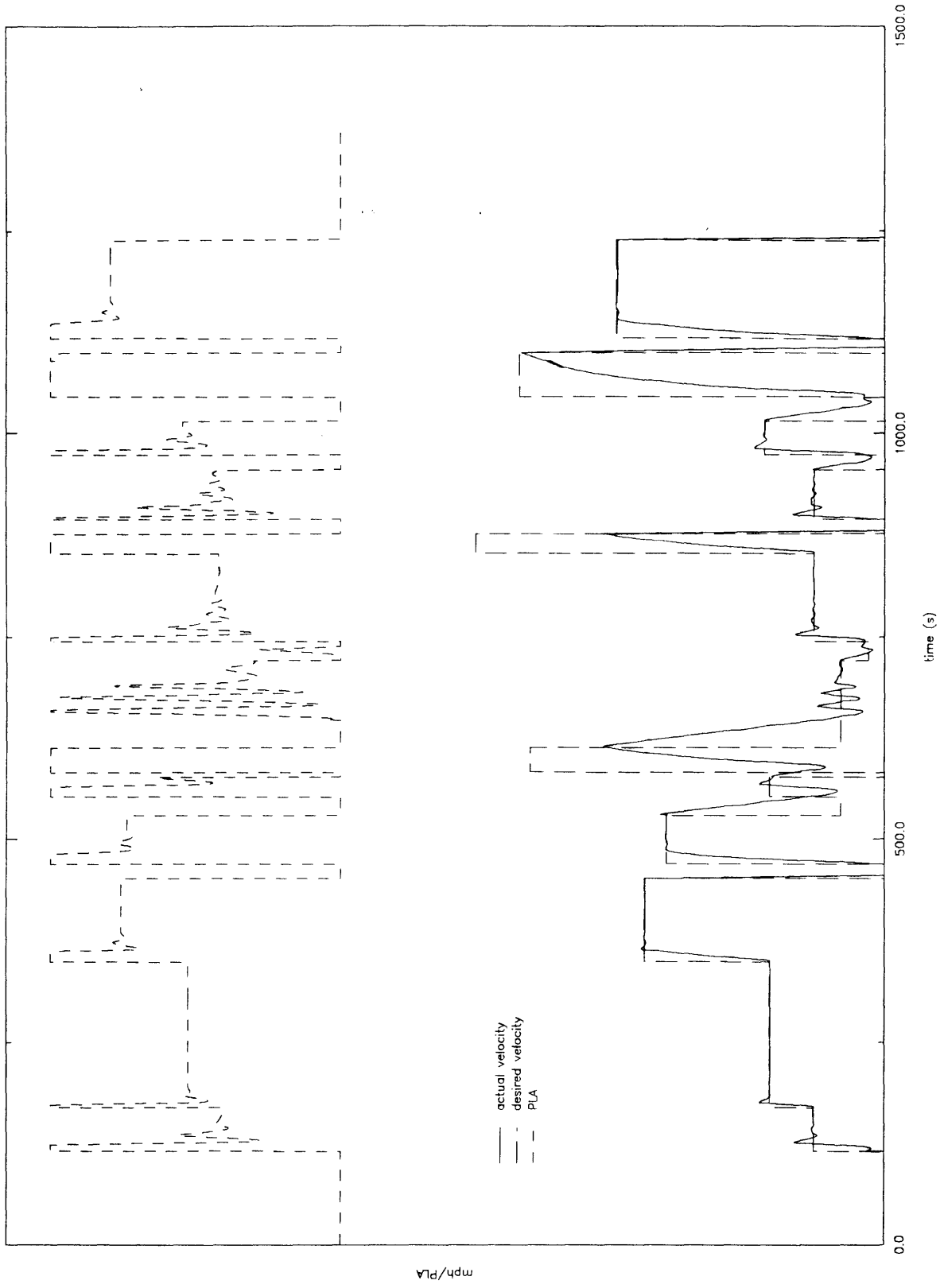

Figure B-8: Conventional controllers and fuzzy logic mode selection: actual vehicle velocity, desired velocity, and PLA for the first mission profile.

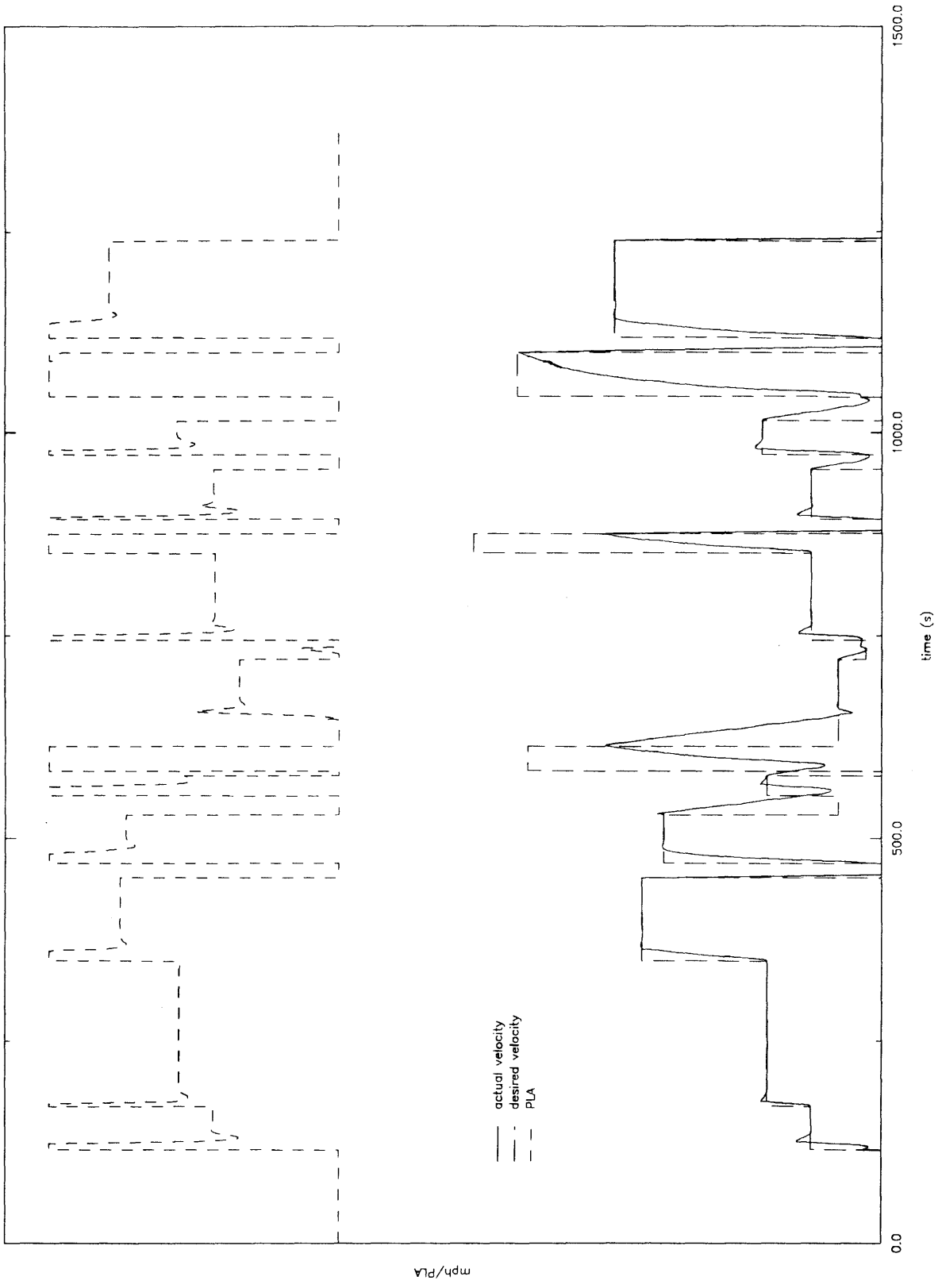

Figure B-9: Fuzzy logic controllers and fuzzy logic mode selection: actual vehicle velocity, desired velocity, and PLA for the first mission profile.

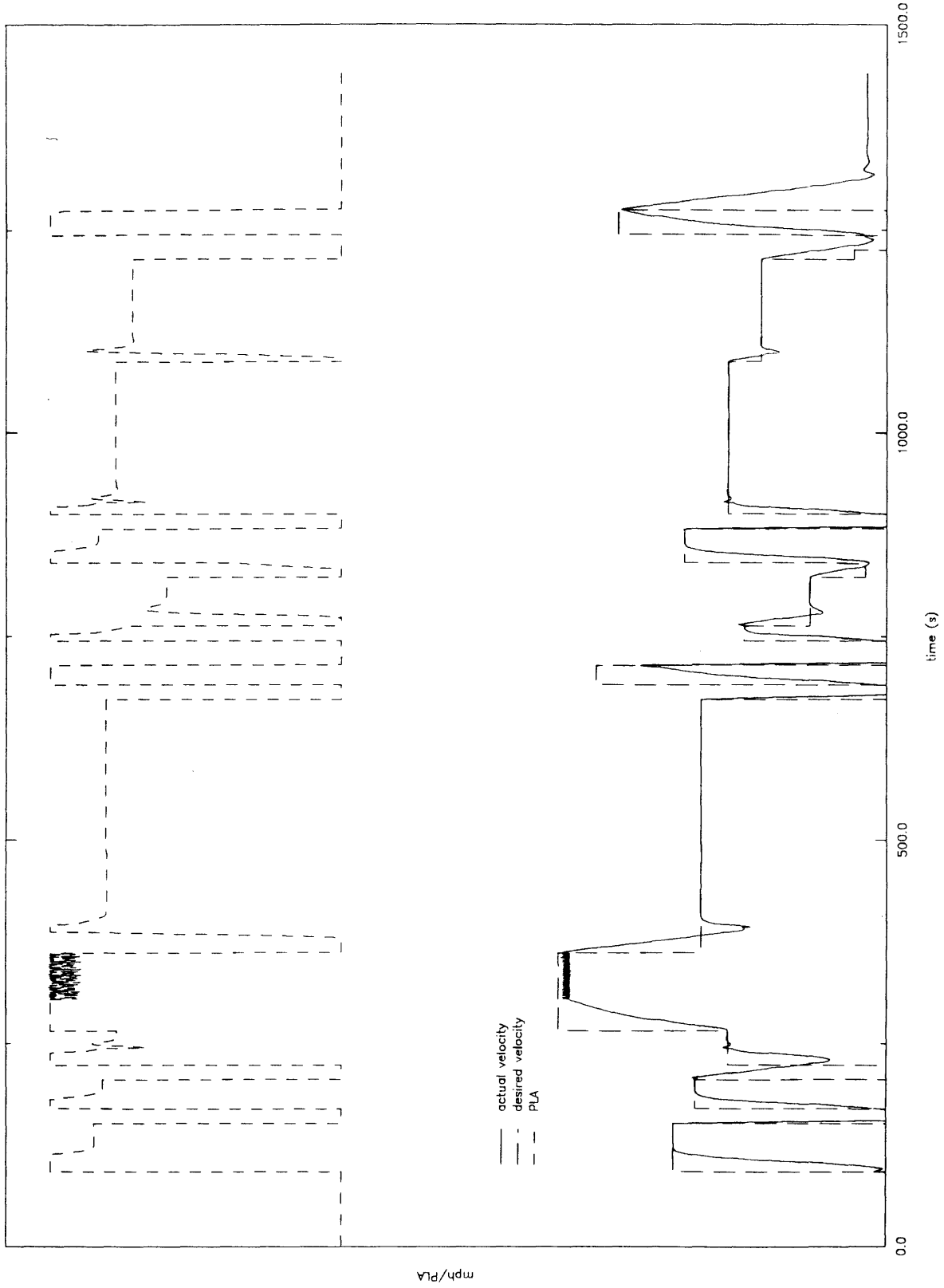

Figure B-10: Conventional controllers and min/max mode selection: actual vehicle velocity, desired velocity, and PLA for the second mission profile.

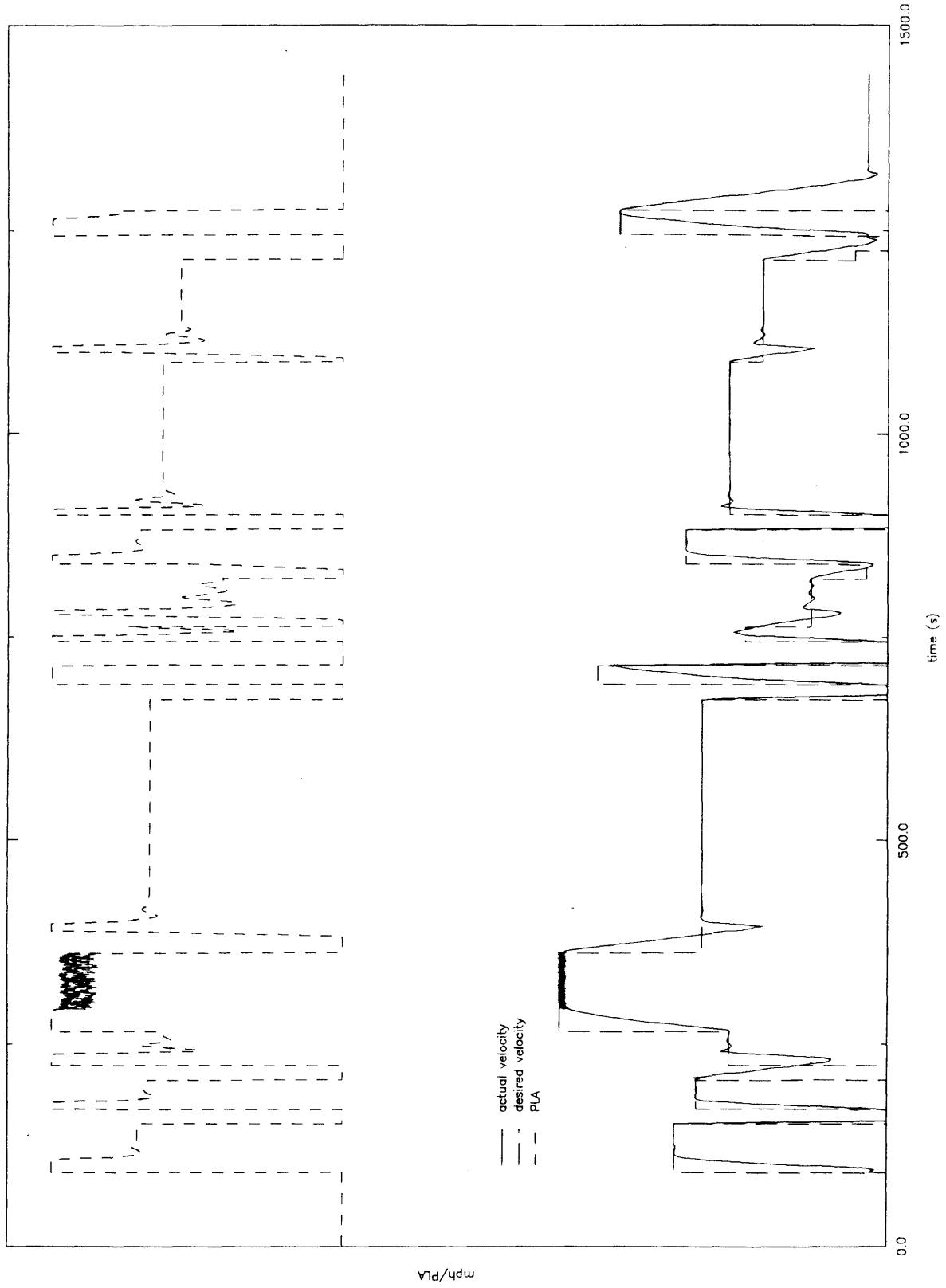

Figure B-11: Conventional controllers and fuzzy logic mode selection: actual vehicle velocity, desired velocity, and PLA for the second mission profile.

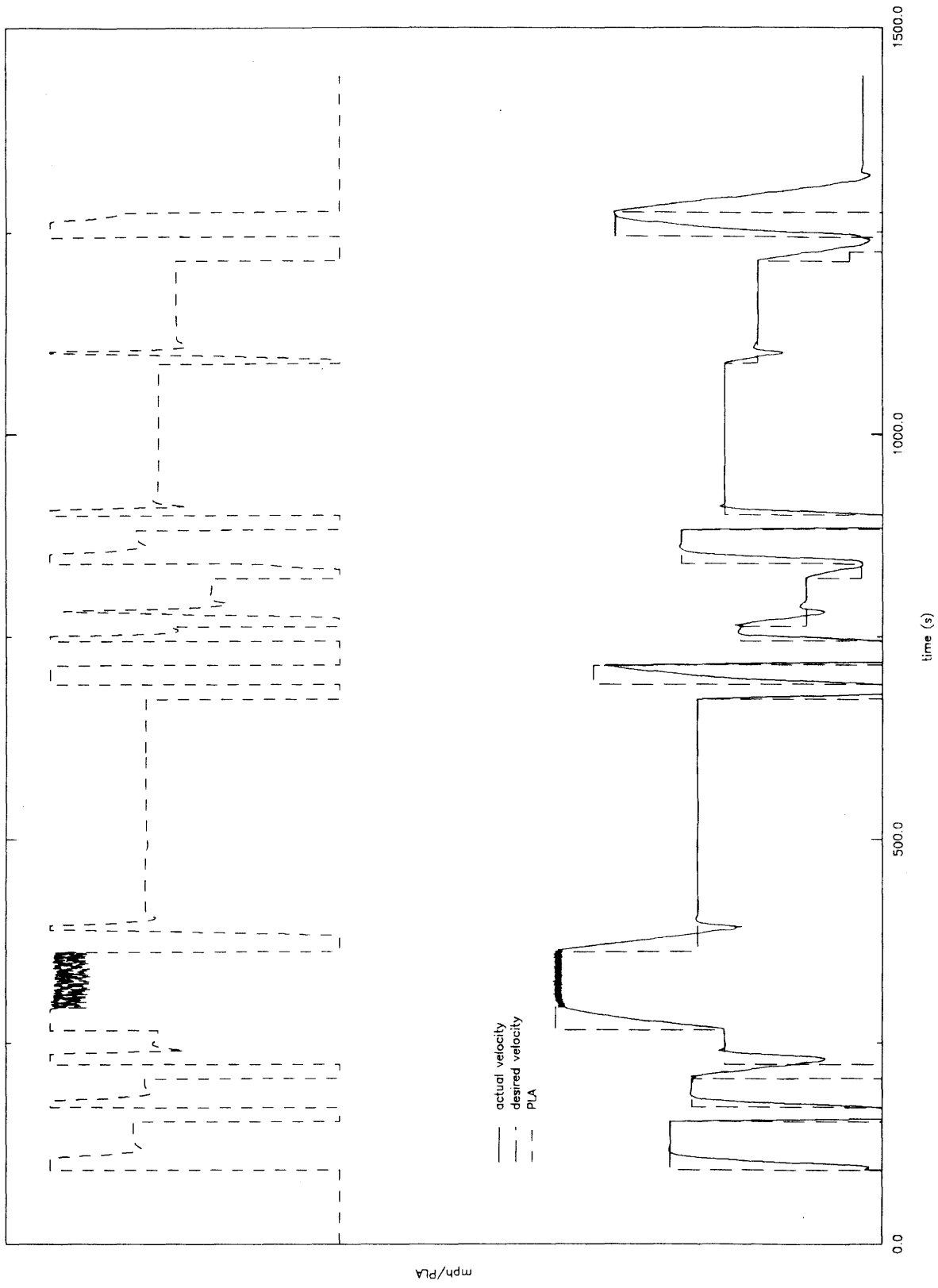

Figure B-12: Fuzzy logic controllers and fuzzy logic mode selection: actual vehicle velocity, desired velocity, and PLA for the second mission profile.

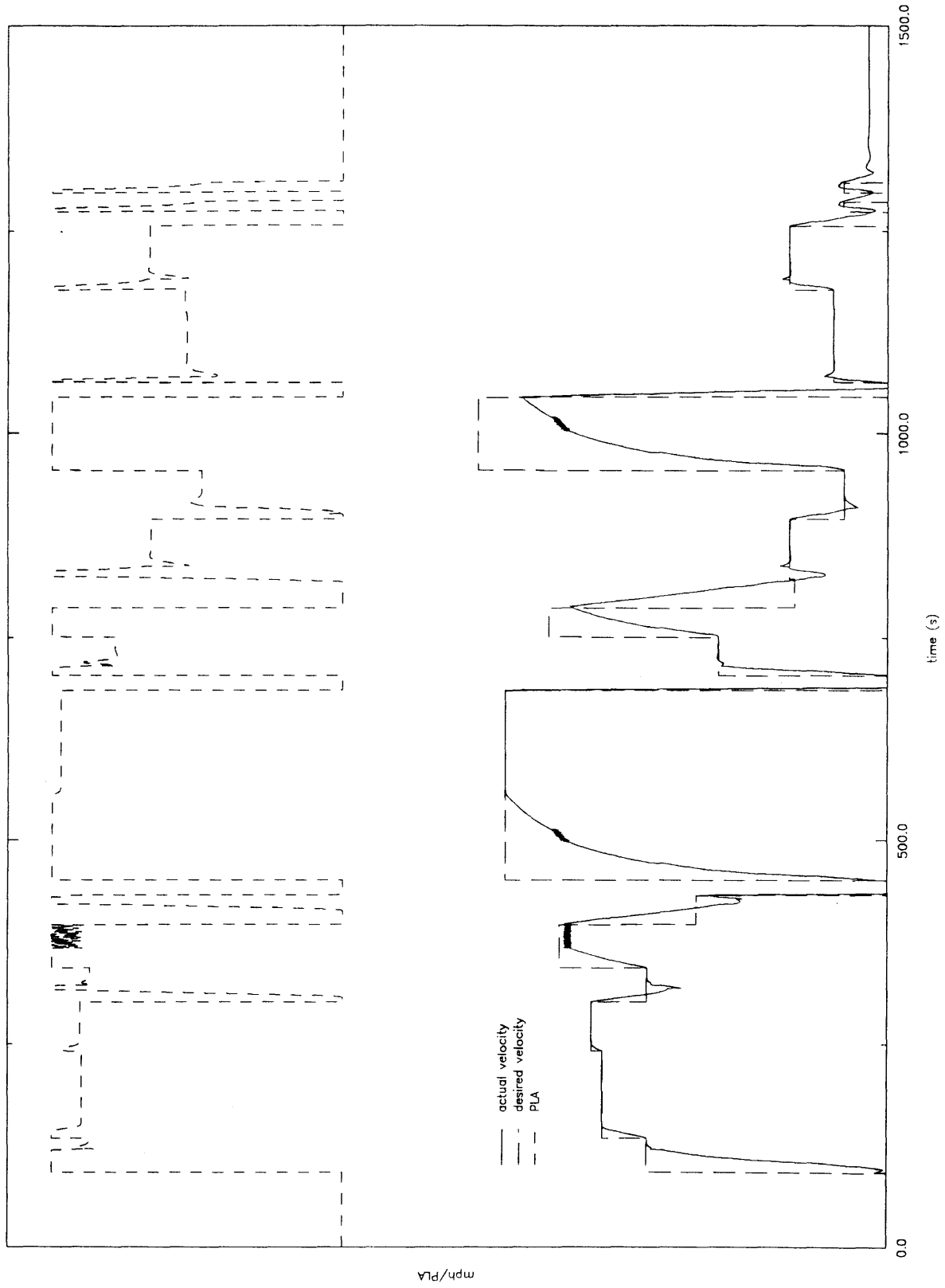

Figure B-13: Conventional controllers and min/max mode selection: actual vehicle velocity, desired velocity, and PLA for the third mission profile.

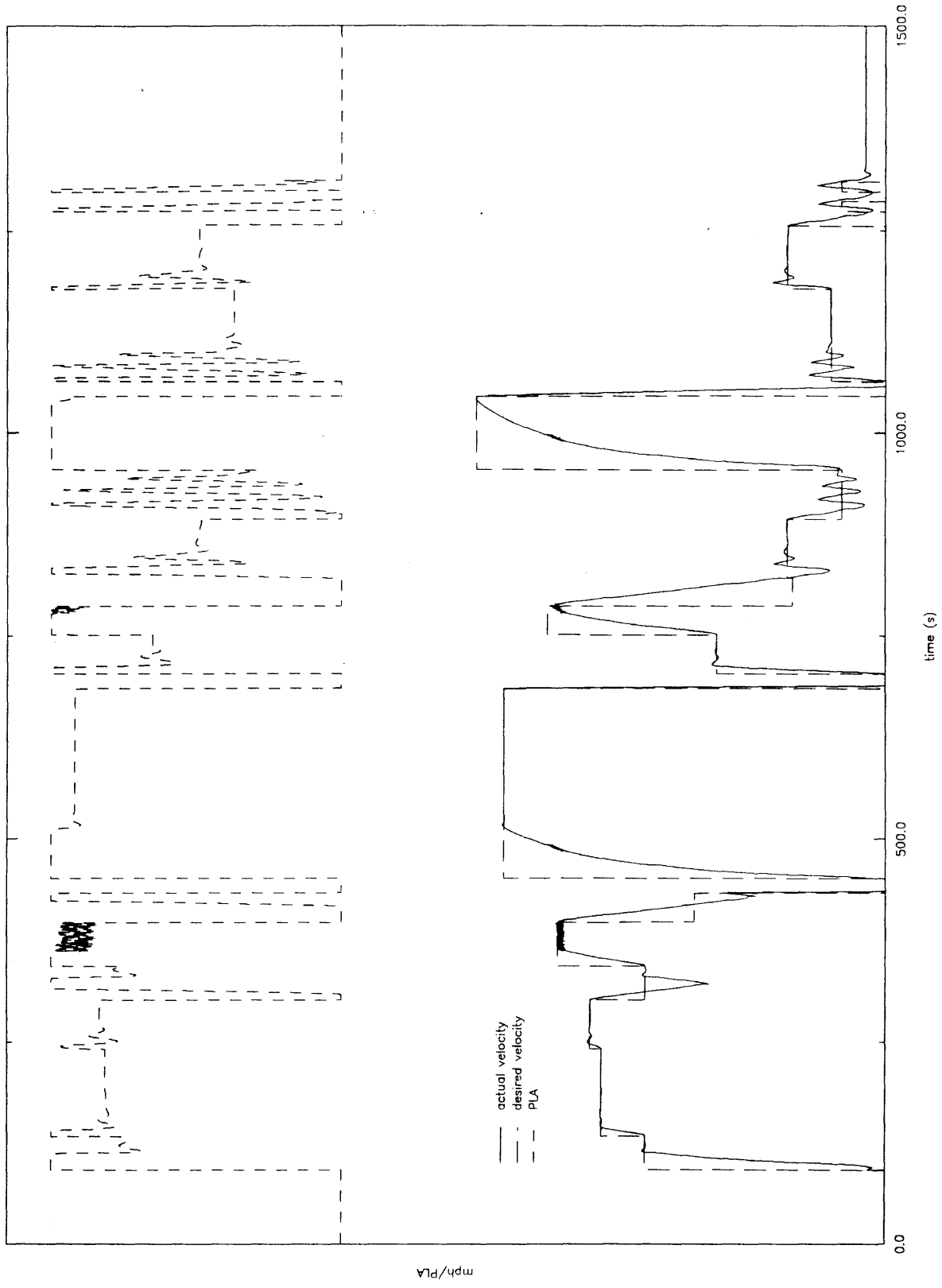

Figure B-14: Conventional controllers and fuzzy logic mode selection: actual vehicle velocity, desired velocity, and PLA for the third mission profile.

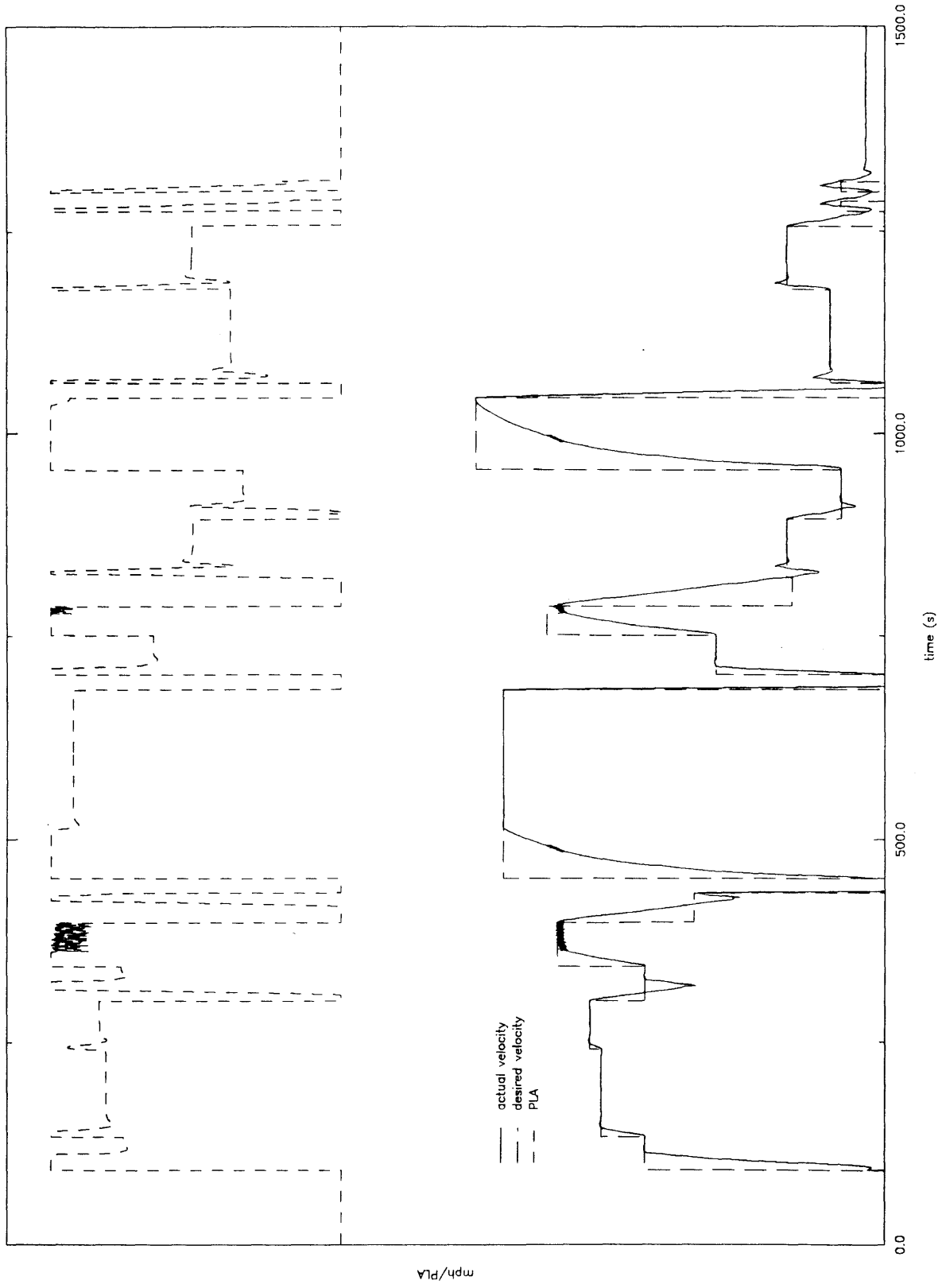

Figure B-15: Fuzzy logic controllers and fuzzy logic mode selection: actual vehicle velocity, desired velocity, and PLA for the third mission profile.

Bibliography

[AIPS] "AIPS Final Report." General Electric document, 1990.

[Baker] Caleb Baker. "Engine Technologies Duel in AIPS Bid." *Defense News*, December 11, 1991, p. 37.

[Bevilacqua] James Bevilacqua. "An Adaptive Update Method for a Regenerative Turboshaft Cycle." Master's thesis, MIT, 1989.

[Bonissone] Piero P. Bonissone. "A Compiler for Fuzzy Logic Controllers." *Fuzzy Engineering toward Human Friendly Systems: Proceedings of the International Fuzzy Engineering Symposium '91, November 13th-15th, 1991*, vol. 2, pp. 706-717.

[Lee] Chuen Chien Lee. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller— Part I" and "Fuzzy Logic in Control Systems: Fuzzy Logic Controller— Part II." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, March/April 1990, pp. 404-435.

[Medeiros] Charles Medeiros, Katherine Koenig, Michael Idelchik, Albion Fletcher, Charles Shortlidge, and David Faymon. "LV100 Engine Control System Requirements." General Electric internal document, 1990.

[Palm] Rainer Palm. "Sliding Mode Fuzzy Control." Siemens AG document. To be presented at the 1992 IEEE International Conference on Fuzzy Systems.

[Slotine] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control.* Englewood Cliffs: Prentice Hall, 1991.

[Sugeno] Michio Sugeno, Toshiaki Murofushi, Junji Nishino, and Hideaki Miwa. "Helicopter Flight Control Based on Fuzzy Logic." *Fuzzy Engineering toward Human Friendly Systems: Proceedings of the International Fuzzy Engineering Symposium '91, November 13th-15th, 1991*, vol. 2, pp. 1120-1121.

[Takahashi] Hiroshi Takahashi, Kenji Ikeura, and Takahiro Yamamori. "5-Speed Automatic Transmission Installed Fuzzy Reasoning." *Fuzzy Engineering toward Human Friendly Systems: Proceedings of the International Fuzzy Engineering Symposium '91, November 13th-15th, 1991*, vol. 2, pp. 1136-1137.

[Tzafestas] Spyros Tzafestas and Nikolaos P. Papanikolopoulos. "Incremental Fuzzy Expert PID Control." *IEEE Transactions on Industrial Electronics*, vol. 32, no. 5, October 1990, pp. 365-371.