

FAILURE DETECTION AND ADAPTIVE CONTROL FOR AUTOMATED
GUIDEWAY TRANSIT SYSTEMS USING THE MULTIPLE MODEL METHOD

by

Jeffrey Garrett Lewis

B.S., Washington University
(1978)

Submitted in partial fulfillment
of the requirements for the
degree of

Master of Science

at the

Massachusetts Institute of Technology

June, 1980

Signature of Author *Jeffrey G. Lewis*
June 27, 1980

Certified by *Alan S. Wilshy*
Thesis Supervisor

Certified by
Thesis Supervisor

Accepted by
Chairman, Department Committee

FAILURE DETECTION AND ADAPTIVE CONTROL FOR AUTOMATED
GUIDEWAY TRANSIT SYSTEMS USING THE MULTIPLE MODEL METHOD

by

Jeffrey Garrett Lewis

Submitted to the Department of Electrical Engineering and Computer Science
on June 30, 1980 in partial fulfillment of the requirements for the degree
of Master of Science.

ABSTRACT

A systematic approach to failure detection using the Multiple Model (MM) algorithm was applied to Automated Guideway Transit (AGT) vehicles, and the ability of an AGT vehicle to adapt to failures was examined using the MM algorithm as a basis for a fault tolerant controller. This was done to determine how the safety and reliability of AGT systems could be improved through detecting and adapting to failures.

Factors influencing the performance of the algorithm such as model selection, robustness, and sensor configuration were examined. The results indicate that failures could be quickly detected and accurately identified using MM. No false alarms occurred even though the models neglected wind and grade and very simply approximated vehicle dynamics.

Under a certain set of communication requirements the fault tolerant controller performed very well. Ways in which these requirements could be relaxed were examined but resulted in degraded performance.

THESIS SUPERVISOR: Alan S. Willsky

TITLE: Associate Professor, Electrical Engineering and
Computer Science

THESIS SUPERVISOR: Paul K. Houpt

TITLE: Assistant Professor, Mechanical Engineering

ACKNOWLEDGEMENTS

There are several people I would like to thank for helping me to get through this trying experience. I would like to thank Professors Willisky and Houpt for the guidance they provided in conducting the research, and Prof. Willisky in particular for his editorial comments.

Ed Chow was always available to discuss ideas with, for which I am grateful. Eric Helfenbein also deserves credit for providing a different perspective on some of the problems we worked on together. I am indebted to Tom Posbergh for providing me a place to stay the last few weeks while I finished my thesis.

I have been supported through the Bell Laboratories One Year on Campus Program.

This work was conducted at the Laboratory for Information and Decision Systems and was supported by the Department of Transportation under Grant DOT/TSC-1685.

TABLE OF CONTENTS

	<u>page</u>
ABSTRACT	2
ACKNOWLEDGEMENTS	3
TABLE OF CONTENTS	4
LIST OF FIGURES	5
LIST OF TABLES	7
1. INTRODUCTION	8
1.1 Background	8
1.2 Thesis Overview	13
2. THE MULTIPLE MODEL METHOD	15
2.1 Multiple Model and Control	19
3. AGT VEHICLE MODELS	23
3.1 Simulation Models	23
3.2 Failure Detection Models	29
4. FAILURE DETECTION PERFORMANCE	35
4.1 Modeled Noise Covariance Selection	35
4.2 Comparison of Failure Detection Models	51
4.3 Detection of Unmodeled Failures	63
4.4 Redundant Sensor Configurations	68
5. FAILURE DETECTION AND CONTROL FOR VEHICLE STRINGS	73
5.1 Effects of Failures Without Failure Detection	73
5.2 Failure Detection Models and Performance	80
5.3 Multiple Model and Control	90
6. SUMMARY AND RECOMMENDATIONS	97
6.1 Summary of Research Results	97
6.2 Suggestions for Further Research	100
REFERENCES	103

LIST OF FIGURES

	<u>page</u>
1.1-1 Failure Detection Filter Block Diagram	11
2-1 Multiple Model Block Diagram	18
2.1-1 MM Controller Using a Decision Rule	21
2.1-2 MM Controller Using Probabilistic Weighting	22
3.1-1 Nonlinear Vehicle Model Block Diagram	24
3.1-2 Linearized Vehicle Model Block Diagram	25
3.1-3 AGT Vehicle Using PI Controller	28
3.1-4 AGT Vehicle Using LQ Controller	28
3.1-5 Vehicle Follower Controller Block Diagram	30
4.1-1 Estimates Tracking Measurements	39
4.1-2 Effect of Bandwidth on Position Residuals	47
4.1-3 Probabilities for Odometer Bias Scenario--No Noise	49
4.1-4 Probabilities for Odometer Bias Scenario--With Noise	50
4.1-5 Probabilities for Speedometer Gain Change Scenario--With Noise	52
4.2-1 Divergence of Position Predicted from Incorrect Measured Velocity	57
4.2-2 True Versus Discretized Acceleration	60
4.3-1 Position Residuals-- $\frac{1}{2}$ m Odometer Bias Scenario	66
5.1-1 Spacing After Various Failures	76
5.1-2 Relative Velocity After Various Failures	77
5.1-3 Steady State Spacing for Various Failures	79
5.2-1 Relative Velocity Underestimated by a Gauss-Markov Model	84

5.3-1	Relative Velocity Estimates	94
5.3-2	Response to Deceleration Maneuver Using Estimates	95

LIST OF TABLES

	<u>page</u>
2-1 Typical Failure Models Used in MM	16
3.1-1 Typical Vehicle Model Parameter Values	26
4-1 Scenario Descriptions	36
4.1-1 Effects of Q and R on Failure Detection	42
4.1-2 Residual and Sensor Noise Covariances	44
4.2-1 Comparison of Failure Detection Model Performance	53
4.3-1 Summary of Unmodeled Failure Detection Performance	64
5.2-1 Multiple Vehicle Failure Detection Performance	86

1. Introduction

For Automated Guideway Transit (AGT) to be a viable form of public transportation it must be safe and reliable. Fault tolerant control systems can help maintain high levels of safety and reliability by detecting and adapting to failures in the vehicle and its subsystems. In this thesis, failure detection and adaptation algorithms have been examined for the longitudinal control of AGT vehicles to determine the safety and reliability they provide. The algorithms that have been examined are based on the Multiple Model (MM) technique and have their roots in modern filtering and estimation theory. They are used in developing a systematic approach to failure detection software. These algorithms have been implemented and their performance evaluated by incorporating them in simulated AGT vehicles on a large digital computer.

1.1 Background

Safety for the longitudinal control of guideway transit systems is primarily a matter of insuring that vehicles stay within service limits on speed, acceleration and jerk and preventing collisions by keeping vehicles separated. Of these, vehicle separation is the most critical safety factor. Inter-vehicle separation, or headway (measured in units of time or distance) has traditionally been maintained using block control. The guideway is divided into sections called blocks and vehicles are allowed to enter only unoccupied blocks; if the next block is occupied the vehicle is commanded to stop. The vehicle must be able to stop before entering the next block so vehicle stopping distance limits the

minimum block size which in turn limits the minimum headway. For headways less than 20 to 30 seconds block control becomes unwieldy, necessitating other control strategies [1].

An alternative to block control which can provide for shorter headways is vehicle follower control, and it is this control strategy that has been used in the research. A vehicle's desired velocity is determined to maintain a given headway using information about the vehicle's motion and the motion of its neighbors. The headway could be specified as a constant time headway, a constant distance headway, or a constant safety factor headway (where the distance between vehicles is proportional to the trailing vehicle's stopping distance). It has been shown that vehicle follower strategies can provide stable control for closely packed strings of vehicles and a variety of control laws have been designed [2-11]. The control law used in this research is described in Section 3.1.

The information provided to the controller by the sensors is an important factor influencing the safety of the system. If a sensor erroneously indicates a large headway between vehicles, the trailing vehicle could collide with the preceding vehicle in attempting to shorten the headway. The conventional approach to this problem would be through the use of redundant sensors. The outputs of two sensors could be compared and if they differed significantly a failure would be indicated. By using three sensors the faulty sensor could be identified and operation could continue using the two unfailed sensors. The replication of hardware can be avoided by using analytic redundancy which employs mathematical relationships to compare the outputs of different sensors. Even if sensors

are triplicated analytic redundancy can benefit the system by still being able to identify which of a pair of sensors has failed if the third has already failed. Unlike triplicated sensors, analytic redundancy can also detect generic sensor failures such as when a temperature change similarly affects all sensors of a physically redundant set. A simple example of analytic redundancy would be to integrate the output of a speedometer to serve as a check on the odometer. Mathematical techniques can also be applied to detect failures in components other than sensors. For example, knowing the input to the propulsion system, a model can be used to compute what the output should be. If the computed output differs significantly from the actual output one can often infer that a failure has occurred.

Even though a host of mathematical techniques have been developed for failure detection (see [12] for a survey) only one has previously been applied to AGT. VanderVelde [13] used the failure detection filter developed by Beard [14] and Jones [15]. The detection filter compares the output of a linear model of the system with the measured output of the system to obtain an error vector, e_0 (see Figure 1.1-1). If the system is described by the equations

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$z(t) = Cx(t)$$

where $x(t)$ is the state vector, $u(t)$ the control vector and $z(t)$ the measurement vector, then the failure detection filter is given by the equations

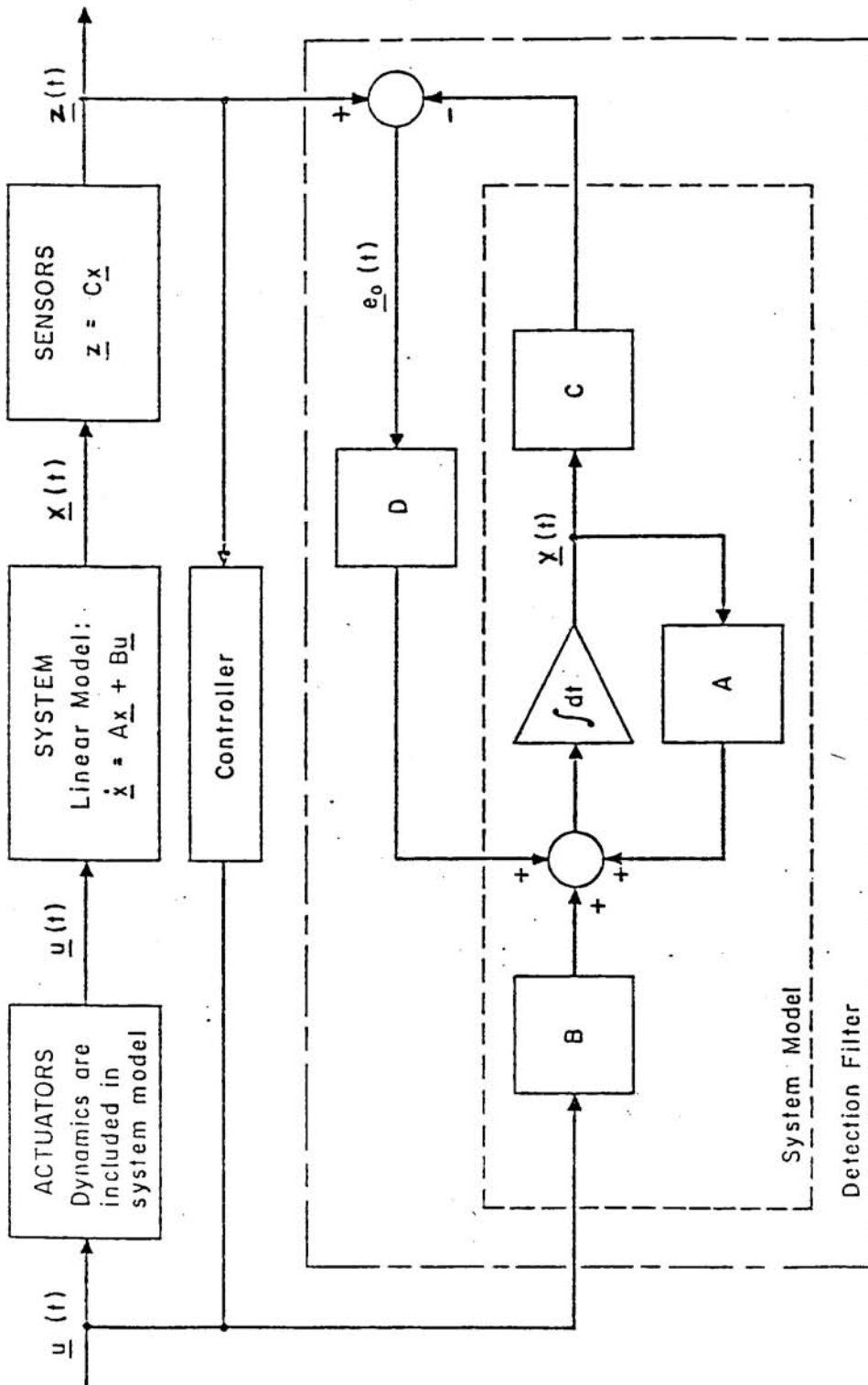


Figure 1.1-1 Failure Detection Filter Block Diagram

$$\dot{y}(t) = Ay(t) + Bu(t) + De_0(t)$$

$$e_0(t) = z(t) - Cy(t)$$

The feedback gain matrix, D , is constructed so that in normal unfailed operation the filter will track the system and the error vector will remain small; but when a failure occurs the error vector will increase in a direction corresponding to the given failure. For example, assume the error vector has two components and that the direction associated with an actuator failure is $(1\ 0)^T$ and the direction associated with a sensor failure is $(0\ 1)^T$ (superscript T denotes transpose). If the actuator fails the error vector will have a large magnitude and be in the $(1\ 0)^T$ direction, i.e., it will have a large first component. A sensor failure would result in an error vector with a large second component. Other failures could be indicated by other directions, but because it is easier to monitor individual components than arbitrary directions additional failure detection filters could be used.

One important feature of the failure detection filter is that its design requires little information about failure modes or how a component fails. The actuator in the example could fail by changing its gain, developing a bias or failing full on but it is immaterial to the failure detection filter. The actuator in the example could fail by changing its gain, developing a bias or failing full on, but all these failures are characterized by an increased magnitude in one component of the error vector. This allows the failure detection filter to use a less detailed model but also prevents it from extracting much information about the

failure. The Multiple Model method uses a more detailed characterization of the failure which allows it to extract more information about the failure, but this could also make it less robust than the failure detection filter. The robustness of the Multiple Model method is examined in Section 4.3.

Information about the failure that could be supplied by the Multiple Model method can help the system in taking corrective action when a failure has been detected. For example, by knowing the gain change of an actuator the input to that actuator could be appropriately scaled to permit continued although possibly degraded operation. The Multiple Model method provides another advantage over the detection filter in that it constantly generates optimal estimates of the system's state. In the event of a sensor failure these estimates could be used to provide information needed by the controller. The Multiple Model method is more fully explained in the next chapter.

1.2 Thesis Overview

The research conducted for this thesis is presented in the next four chapters. In Chapter 2 the Multiple Model algorithm is described. The nonlinear model used to simulate the AGT vehicle is discussed in the first half of Chapter 3. The simulated vehicle was used to generate data for testing the algorithm similar to data that would have come from an actual AGT vehicle. Much simpler linear models were used to describe the AGT vehicle in the failure detection algorithm. The various models used in the algorithm are discussed in the second half of Chapter 3. These simpler models attempt to describe only the aspects of the vehicle

necessary for failure detection, and thus avoid degrading failure detection performance by using complex models subject to parameter uncertainty.

The ability of these various models to detect failures and other factors influencing the performance of the algorithm are discussed in Chapter 4. Specifically, the effects of noise parameters used in the algorithm, the robustness of the algorithm, and redundant sensor configurations were examined. It was found that the algorithm could quickly detect and accurately identify failures with no false alarms when the noise parameters were properly selected. It was shown how to determine when redundant sensors would yield the most benefit and that the algorithm was fairly robust.

The performance of the MM algorithm in detecting failures and in a simple fault tolerant control strategy is examined in Chapter 5. Under one plausible set of conditions excellent control results were obtained. Other sets of conditions were examined under which the control strategy was not as effective.

Chapter 6 summarizes the results of this thesis and presents suggestions for further research.

2. The Multiple Model Method

In the Multiple Model (MM) method failure detection is formulated as a problem of determining which of several linear models most accurately describes the system [16-20]. Each model is based on a different hypothesis about the condition of the system--whether it is unfailed or has suffered a failure of some sort. The basic form of the models in continuous time is

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t) \\ z(t) &= Cx(t) + \zeta(t)\end{aligned}\tag{1}$$

where $x(t)$ is the state vector, $u(t)$ the control vector, $z(t)$ the observation vector, $w(t)$ and $\zeta(t)$ zero mean white Gaussian noise vectors of covariance Q and R respectively. Table 2-1 gives a set of models (neglecting noise) illustrating various failures that can be hypothesized for a simple vehicle model. Vehicle models will be discussed later in more detail. More complex vehicle models could also allow changes in the dynamics with different A matrices.

The MM algorithm generates a set of maximally informative statistics on which to base decisions about which model corresponds to the actual system. For each model, the MM algorithm computes the probability of that model matching the system given observations up to the current time.

The computation of the probabilities relies heavily upon modern estimation theory and the Kalman filter; because the algorithm will be implemented on a digital computer, the discrete-time Kalman filter is used.

Table 2-1 Typical Failure Models Used in MM

Hypothesis 0: Unfailed Vehicle

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \end{bmatrix} f(t)$$

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix}$$

Hypothesis 1: Actuator Amplifier Changes Gain to k

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ k/M \end{bmatrix} f(t)$$

(observation equation unchanged from H_0)

Hypothesis 2: Actuator Bias, DC Offset of b

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M \end{bmatrix} f(t) + \begin{bmatrix} 0 \\ b \end{bmatrix}$$

(observation equation unchanged from H_0)

Hypothesis 3: Speedometer Fails to Zero

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix}$$

(dynamics equation unchanged from H_0)

Hypothesis 4: Speedometer Sticks at Velocity V

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ V \end{bmatrix}$$

(dynamics equation unchanged from H_0)

$s(t)$ distance traveled
 $v(t)$ velocity
 $f(t)$ force applied to vehicle
 M mass of vehicle
 subscript m denotes measured quantity

The structure of the MM algorithm is illustrated in Figure 2-1. Each Kalman filter in Figure 2-1 is of the form

$$\begin{aligned}\hat{x}(k|k) &= \hat{x}(k|k-1) + Hr(k) \\ \hat{x}(k+1|k) &= A\hat{x}(k|k) + Bu(k) \\ r(k) &= z(k) - C\hat{x}(k|k-1)\end{aligned}\tag{2}$$

where H is the Kalman gain, $r(k)$ the residual vector and $\hat{x}(k+1|k)$ the optimal estimate of the state at time $k+1$ given measurements through time k . The matrices H , A , B , and C for the i^{th} hypothesis (H_i) are those consistent with the model corresponding to H_i . If the i^{th} hypothesis is the correct one, the residuals produced by the model corresponding to H_i will be a zero mean white Gaussian process with probability density

$$p_i(k) = p(r_i(k)|H_i) = [(2\pi)^n \det(\Sigma_i)]^{-1/2} \exp[-\frac{1}{2} r_i^T(k) \Sigma_i^{-1} r_i(k)] \tag{3}$$

where Σ_i is the precomputable covariance for the steady state Kalman filter based on H_i . Essentially, $p_i(k)$ is a measure of the probability of the residual produced by model i assuming model i is correct. Using these $p_i(k)$ and Bayes' rule the probability of the system matching model i can be recursively computed according to the equation [20]

$$p_i(k) = \frac{p_i(k-1)p_i(k)}{\sum_{j=1}^N p_j(k-1)p_j(k)} \tag{4}$$

There is an assumption implicit in the above equations that causes a minor difficulty in the implementation of MM. This assumption is that

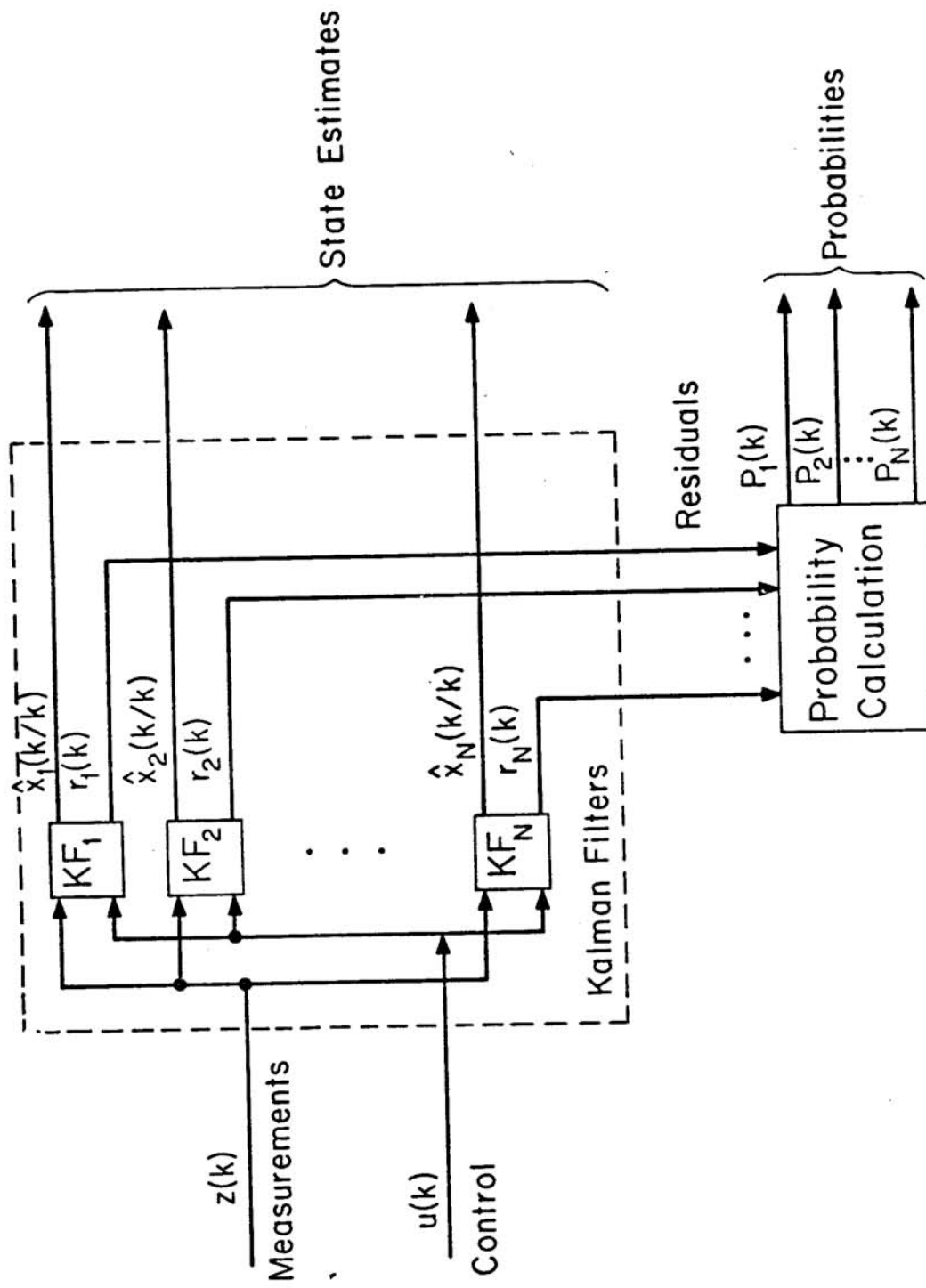


Figure 2-1 Multiple Model Kalman Filter Block Diagram

the system does not change and is always described by the same model. For example, if the set of models in Table 2-1 were used and the past sequence of $p_i(k)$ indicated H_0 was correct, P_0 would approach unity and all other P_i would approach zero. If a failure then occurred equation (4) indicates that P_0 would remain unity and the other P_i would remain zero. This can easily be prevented by setting a lower limit on the P_i so that they cannot approach zero. Another problem is the degradation of the state estimates. For example, while the system is unfailed the state estimate for the model based on a biased actuator (H_2 in Table 2-1) will diverge from the true state. If a bias did develop the residuals for H_2 would remain large because of the error in H_2 's state estimate, and these large residuals would keep P_2 small. This problem can be solved by setting the state estimates of exceedingly improbable models to the estimate of the most probable model.

These problems have not proven to be serious difficulties in previous applications of MM [16-20]. It has been successfully used to detect freeway incidents [16] and abnormalities in electrocardiograms [17] and was used with moderate success as an adaptive controller for the F-8 aircraft [18,19].

2.1 Multiple Model and Control

The $P_i(k)$ computed by the MM algorithm provide information about the system but the issue of how to use this information to determine a proper control for the system remains and has been the subject of previous investigations [20]. Two different ways of using these probabilities are illustrated in Figures 2.1-1 and 2.1-2. In Figure 2.1-1 the probabilities

are used in conjunction with a decision rule that decides which model corresponds to the system. The control designed for that model is then used to drive the system. In Figure 2.1-2 the probabilities are used to weight the various controls which are then added to produce the control for the system. The latter method avoids the need for a decision rule but can exhibit stability problems as shown in one of its previous applications [19].

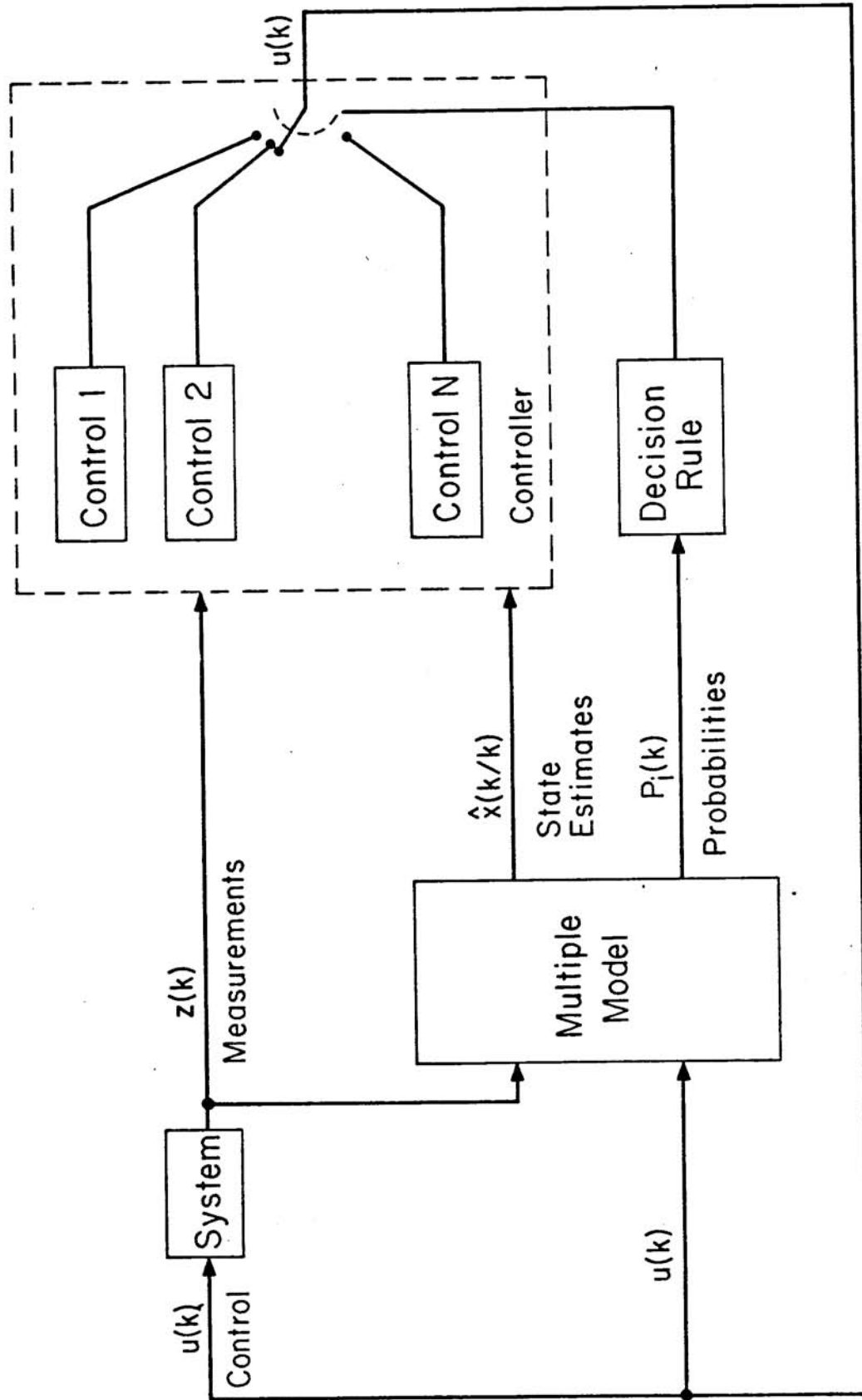


Figure 2.1-1 MM Controller Using a Decision Rule

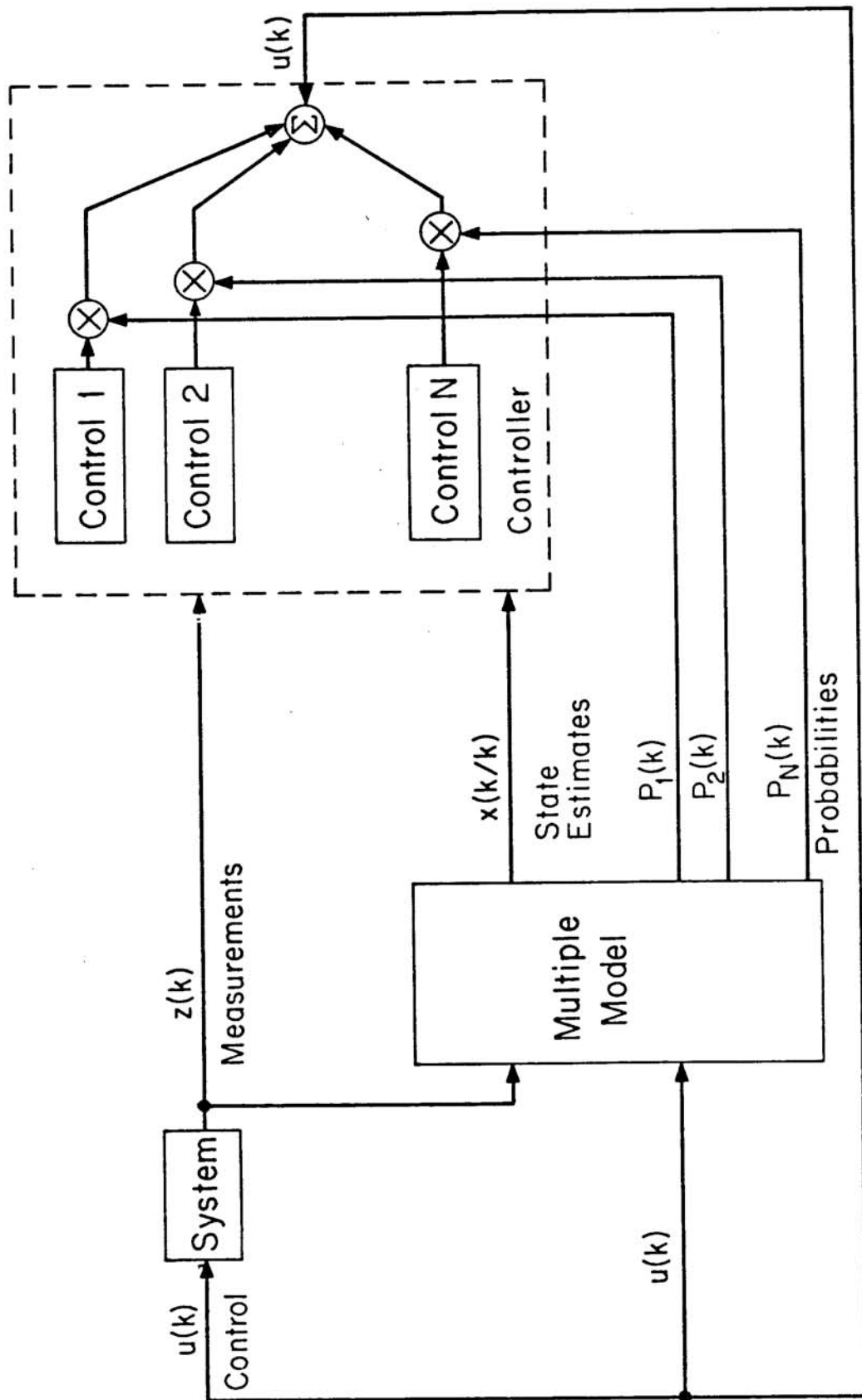


Figure 2.1-2 MM Controller Using Probabilistic Weighting

3. AGT Vehicle Models

In our research, AGT vehicles were simulated on a digital computer to produce sensor outputs similar to those of a real vehicle going through various maneuvers and developing various failures. The simulated vehicle's sensor outputs were used in place of those of a real vehicle in the failure detection algorithm. The sensor outputs were processed by the MM algorithm to demonstrate the various issues involved in failure detection. Different sets of models were employed in the MM algorithm approximating the vehicle at various levels of complexity and using various sensor configurations to determine their relative merits. The failure detection models are not as complex as the simulation model because they attempt to capture only the significant characteristics of the vehicle whereas the simulation model attempts to duplicate a real vehicle at a reasonable level of complexity.

3.1 Simulation Models

A wheeled vehicle driven by a DC electric motor was simulated to produce the sensor outputs used as input to the failure detection algorithm. A block diagram for this type of vehicle developed by Pitts [21] is given in Figure 3.1-1, and typical parameter values for two vehicles are given in Table 3.1-1. In addition to the electric motor this model includes a force due to grade and a nonlinear aerodynamic drag force. Linearizing this model and transforming it to obtain position, s , velocity, v , and acceleration, a , as state variables results in the equivalent linear model of Figure 3.1-2. The constants C_0 , C_1 and K_m depend on the model parameters and the nominal velocity about which the model was

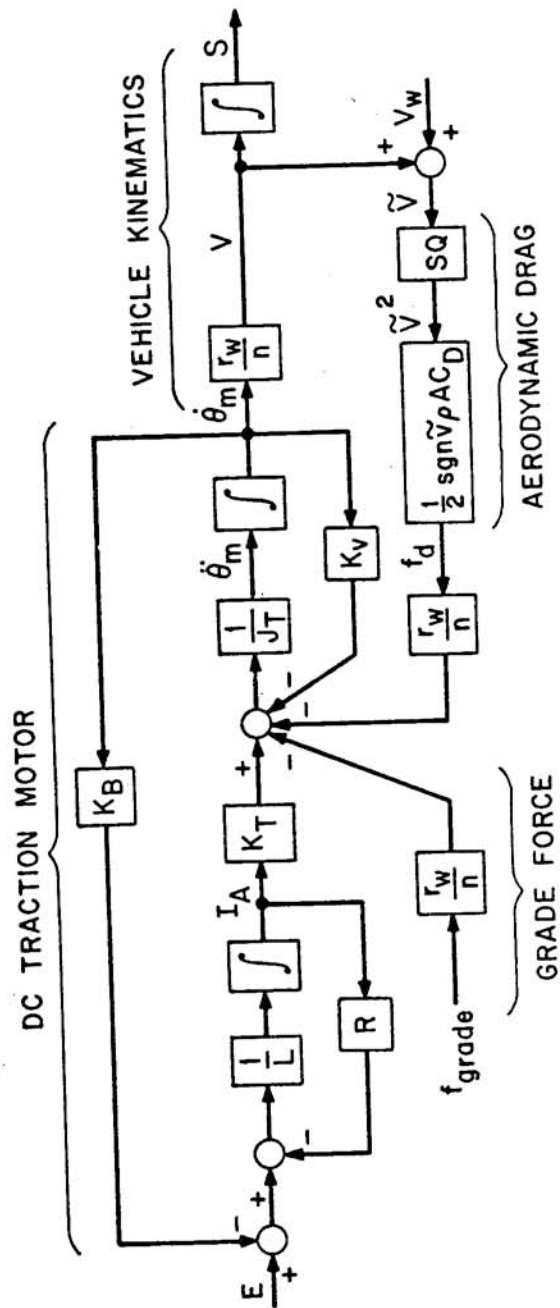


Figure 3.1-1 Nonlinear Vehicle Model Block Diagram

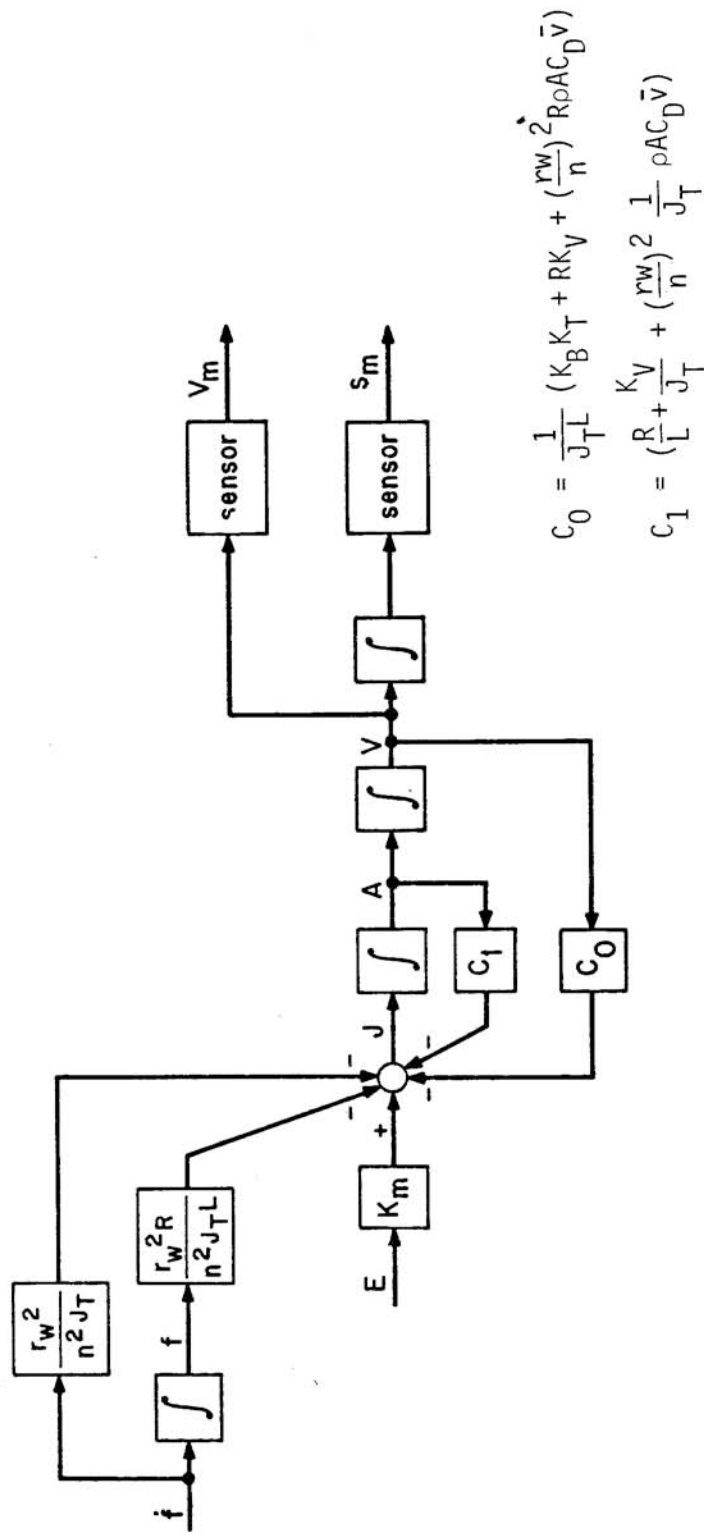


Figure 3.1-2 Linearized Vehicle Model Block Diagram

Table 3.1-1 Typical Vehicle Model Parameter Values

Symbol	Parameter (units)	Vehicle A	Vehicle B
K_T	motor torque constant ($\frac{nt-m}{A}$)	1.18	.827
K_B	motor back emf constant ($\frac{V-s}{rad}$)	1.26	.88
R	armature resistance (ohm)	.0415	.0203
L	armature inductance (henry)	.0011	.00052
r_w	wheel radius (m)	.35	.35
m	gear ratio (dimensionless)	4	3.82
M	vehicle mass (kg)		
	empty	2120	663
	mean	2500	979
	full	2880	1295
J_T	total rotational inertia ($kg-m^2$)	16.23	6.03
	(motor inertia + reflected inertia)	19.14	8.68
		22.05	11.33
A	vehicle frontal area (m^2)	3.4	3.4
C_d	drag coefficient (dimensionless)	0.7	0.7
ρ	density of air ($\frac{kg}{m^3}$)	1.22	1.22

linearized. Simulations were conducted with both linear and nonlinear models, and except in isolated cases to be discussed shortly there were negligible differences in the simulated behavior.

Two different controllers were used to produce a voltage when given a velocity command. Initially the proportional-integral (PI) controller of Pue [22] as illustrated in Figure 3.1-3 was used to control the vehicle described by Column A in Table 3.1-1. The values of k_1 and k_2 from [22] produced a large bandwidth system in which the measured velocity, v_m , closely tracked the commanded velocity, v_c , and was relatively unaffected by grade and wind disturbances. However, when velocity sensor failures were simulated, the abrupt change in v_m produced unreasonably large jerks and accelerations violating motor power constraints given in [23]. By limiting voltage and current in the nonlinear simulator, acceleration was prevented from becoming extremely large and power constraints were not violated, but jerk was still excessive.

The modified form of Chiu, Stupp and Brown's linear-quadratic (LQ) controller [24] used by Draper [25] did not exhibit the same characteristics as the PI controller. The LQ controller is illustrated in Figure 3.1-4 using Draper's notation for the gains. The vehicle described in Column B of Table 3.1-1 using the LQ controller has a much narrower bandwidth than that of the vehicle using the PI controller. Because of its narrower bandwidth it does not respond as violently to speedometer failures but it also responds more slowly in correcting disturbances due to wind and grade. The effects of these two different controllers on failure detection will be discussed later.

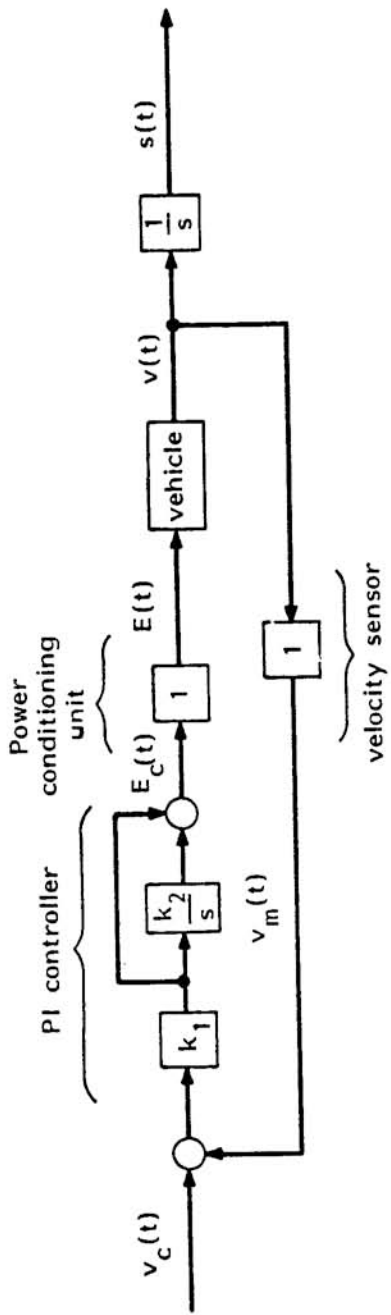


Figure 3.1-3 AGT Vehicle Using PI Controller

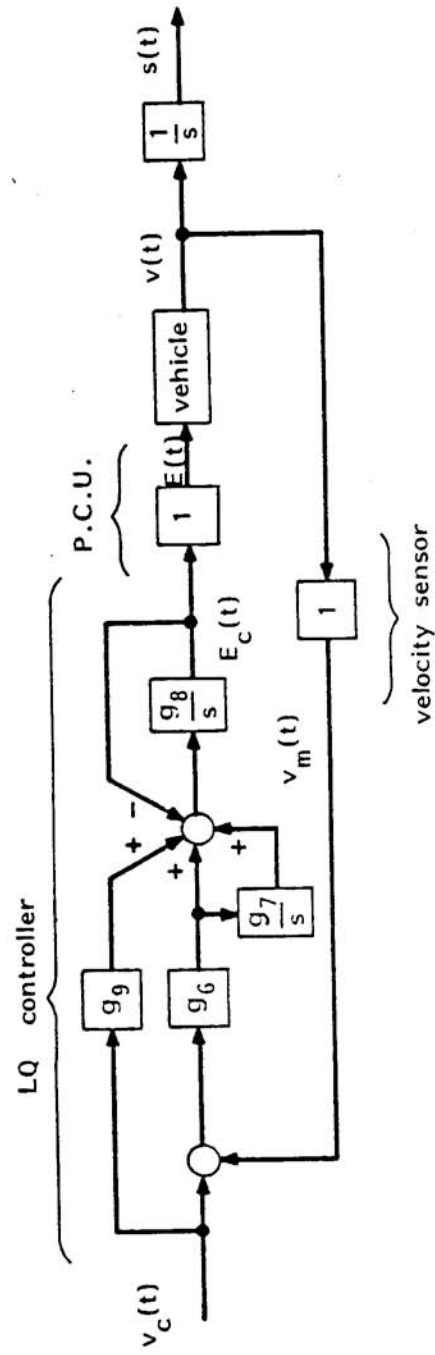


Figure 3.1-4 AGT Vehicle Using LQ Controller

In single vehicle simulations velocity commands were generated by doubly integrating a jerk command. In multiple vehicle simulations velocity commands were generated for tracking vehicles using Draper's safe approach controller [25]. This controller computes a minimum safe spacing according to the equation

$$s_{\min} = c_1(v_2^2 - v_1^2) + c_2(v_2 - v_1) + c_3v_2 + c_4$$

a simplified form of the kinematic constraints developed by Pue [11]. The preceding vehicle's velocity is v_1 , the trailing vehicle's velocity is v_2 and c_1 , c_2 , c_3 and c_4 are constants. Velocity commands are generated to keep the trailing vehicle on the safe approach curve by using a linearization of the curve about the point $(v_2, s_{\min}(v_2))$,

$$v_c = v_2 + \frac{\Delta s - \Delta s_{\min}(v_2)}{\left. \frac{\partial \Delta s_{\min}}{\partial v_2} \right|_{v_2}}$$

The block diagram of this controller is given in Figure 3.1-5. Modifications to this controller in the event of a failure are discussed in Chapter 6.

3.2 Failure Detection Models

A variety of simple models are used in the research to approximate the vehicle in the failure detection algorithm. These models attempt to characterize the significant aspects of the real vehicle in various ways. The models examined are not intended to provide a comprehensive list but

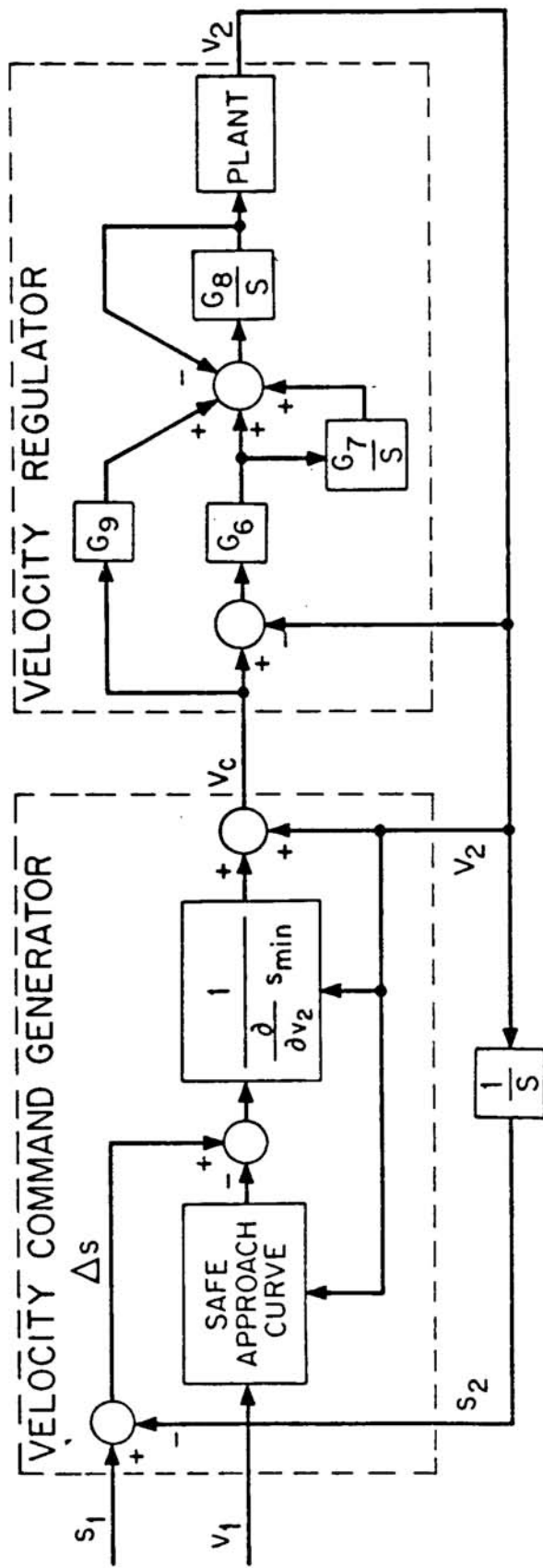


Figure 3.1-5 Vehicle Follower Controller Block Diagram

rather to illustrate the advantages and disadvantages of various types of models. Continuous time models are given here but their discrete time equivalents are used in the research.

The simplest class of models is that based on the kinematic relationships between position, $s(t)$, velocity, $v(t)$, and acceleration, $a(t)$. Assuming that the commanded acceleration, $a_c(t)$, is equal to the true acceleration, the vehicle can be modeled as

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_c(t)$$

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix}$$

KC2--Kinematic Command driven model 2 state

where the subscript m denotes measured quantities. This model allows position and velocity sensor failures to be modeled. In our research position sensor biases and velocity sensor gain changes are modeled. For position sensor biases the measurement equation takes the form

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix}$$

where b is the magnitude of the bias. For a velocity sensor which changes its gain to k the measurement equation becomes

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix}$$

but because the velocity sensor is in a feedback loop it affects the relationship between commanded and true acceleration and therefore the B matrix. Measured velocity tracks commanded velocity, i.e.,

$$v_c(t) = v_m(t)$$

but if $v_m(t) = kv(t)$

then $v_c(t) = kv(t)$

or $\dot{v}(t) = \frac{1}{k} a_c(t)$

so that the state equation becomes

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{k} \end{bmatrix} a_c(t)$$

The MM algorithm when using this model detects failures by detecting discrepancies between commanded and measured vehicle behavior so that it is possible to detect controller failures or that controller failures cause sensor failure models to be indicated when this model is used. For example, suppose that the controller developed a bias so that the applied voltage is larger than it should be. The bias would cause the vehicle's velocity to increase even though there were no commanded acceleration. The measured velocity would be larger than the velocity computed by integrating the acceleration command and this discrepancy could be attributed to a velocity sensor failure if there were no other model that could account for the discrepancy.

However, failures are not the only cause of discrepancies between

commanded and actual behavior. Discrepancies also occur because of plant dynamics that are not included in this model and because of disturbances such as wind and grade.

One way to avoid these discrepancies is to use measured acceleration instead of commanded acceleration as the driving term. The model is identical in form to the previous model,

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_m(t)$$

$$\begin{bmatrix} s_m(t) \\ v_m(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \end{bmatrix}$$

KS2--Kinematic Sensor driven model 2 state

This model uses no command information making it independent of the relationship between commanded and actual behavior. For this reason it also does not reflect possible controller failures. Sensor biases and gain changes are modeled straightforwardly.

A simpler sensor driven model that does not require an accelerometer is

$$\dot{s}(t) = [0]s(t) + [1]v_m(t)$$

$$s_m(t) = [1]s(t)$$

KS1--Kinematic Sensor driven model 1 state

Another more complicated model used in our research that avoids assuming that commanded acceleration equals true acceleration is based on Figure 3.1-2:

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \\ \dot{a}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -C_0 & -C_1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \\ a(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_M \end{bmatrix} E_C(t)$$

$$\begin{bmatrix} s_m(t) \\ v_m(t) \\ a(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \\ a(t) \end{bmatrix}$$

P3--Parameterized model 3 state

which uses plant parameters to explicitly model the dynamics. This model is used in our research as a basis for modeling power conditioning unit failures in addition to sensor failures. There is a drawback to this model in that it is subject to modeling errors. The values of C_0 , C_1 and K_M cannot be known exactly but only approximated. The validity of this approximation is examined later in Chapter 4.

4. Failure Detection Performance

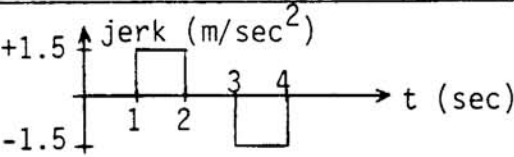
Failure detection performance was examined by using the MM algorithm to process sensor outputs generated by a simulated vehicle. The first two sections attempt to determine how the various parameters in the algorithm affect the performance. The first section examines the effects of varying the modeled noise covariance matrices; in the second section various possibilities for the A, B and C matrices used to parameterize the failures are examined. How well the algorithm can detect failures that are not explicitly modeled is investigated in Section 3. In the fourth section the addition of redundant sensors and how they improve failure detection is discussed. After examining the algorithm and how it can be applied to a single vehicle, the application of the algorithm to pairs of vehicles moving in tandem is explored in Chapter 5.

Several basic scenarios are used throughout the rest of the thesis to examine the properties of the algorithm. These scenarios are described in Table 4-1.

4.1 Modeled Noise Covariance Selection

The modeled plant noise covariance, Q , and the modeled sensor noise covariance, R , are important parameters in the failure detection algorithm because of the influence they have in determining the residual covariance, Σ , and the Kalman gain, H . So that the significance of Q and R can be better understood, the roles that the residual covariance and the Kalman gain play in the algorithm and how they are affected by the modeled noise

Table 4-1 Scenario Descriptions

Scenario	Description*
maneuver	 <p>commanded velocity increases from 15 to 18 m/sec (34 to 40 mi/hr)</p>
grade	<p>at t = 1 sec grade starts increasing linearly until a grade of 10% is reached at t = 2 sec maximum force due to grade: 2450 nt for vehicle A 960 nt for vehicle B</p>
wind	<p>at t = 1 sec wind velocity starts increasing linearly until a wind velocity of 18 m/sec (40 mi/hr) is reached at t = 1.5 sec maximum force due to wind: 1250 nt for both vehicles</p>
+1 m odometer bias	<p>at t = 1 sec measured position becomes 1 m larger than true position</p>
.9 speedometer gain	<p>at t = 1 sec measured velocity becomes 10% smaller than true velocity</p>
.9 accelerometer gain	<p>measured acceleration is 10% smaller than true acceleration as the vehicle goes through a maneuver</p>
1.1 PCV gain	<p>at t = 1 sec voltage applied to motor becomes 10% larger than commanded voltage</p>

*all scenarios begin with the vehicle traveling at a constant 15 m/sec (34 mi/hr)

covariances will be discussed before the results are presented.

Multiple Model is based on using the residuals, the differences between predicted and actual measurements, to detect failures. The residual covariance indicates how close to zero the residuals are expected to be; the smaller the covariance the more closely the predictions and measurements are expected to agree. If the various models accurately predict the various possible states of the system and the measurements accurately reflect the true state of the system then failures can be quickly detected. Inaccuracies in either predictions or measurements must be accounted for with a larger modeled plant or sensor noise covariance resulting in a larger residual covariance and slower detection of failures.

Two equations from Kalman filter theory show how Q and R affect the residual covariance.

$$\Sigma = C\Theta_p C^T + R \quad (1)$$

$$\Theta_p = A\Theta A^T + Q \quad (2)$$

where Θ_p is the predicted state error covariance and Θ is the updated state error covariance. The first term on the right hand side of equation (1) is due to prediction error; the second term is due to sensor noise. Equation (2) shows how the predicted state error covariance is due to plant noise and uncertainty in the state on which the prediction is based. Comparisons of R and Σ are used later to determine the primary influences on residuals as modeled by the algorithm.

Although the effects that the Kalman gain has upon failure detection are not as clearly demonstrated by the results of this section as the effects of Σ , it is still an important parameter in the algorithm. The

Kalman gain, H , is selected to produce optimal state estimates by properly weighting the information about the true state of the system contained in the predictions and in the measurements. The bandwidth of the filter is indicative of how these two sources are weighted. A small bandwidth filter will base estimates primarily on the predictions because the measurements are considered relatively inaccurate. A large bandwidth filter will base estimates primarily on the measurements. Figure 4.1-1 illustrates how a small and a large bandwidth filter would respond if the velocity measurements suddenly indicated an increase in velocity when the model indicated a constant velocity. The large bandwidth filter tracks the measurements more quickly than the small bandwidth filter. The effects of bandwidth on failure detection are discussed later.

Effects of varying Q and R

The effects of varying Q and R were examined using the two state kinematic control driven model, KC2, as a basis for the following set of models:

1. unfailed
2. position sensor bias of -1 m
3. position sensor bias of +1 m
4. velocity sensor gain of .9
5. velocity sensor gain of 1.1

The MM algorithm needs explicit specifications of failure modes for each instrument and we have chosen representative failure modes. It is of course possible that the actual mode of failure of an instrument may differ from our model. For example, a velocity sensor might experience a gain change to .8 rather than .9, or it might get stuck at some value.

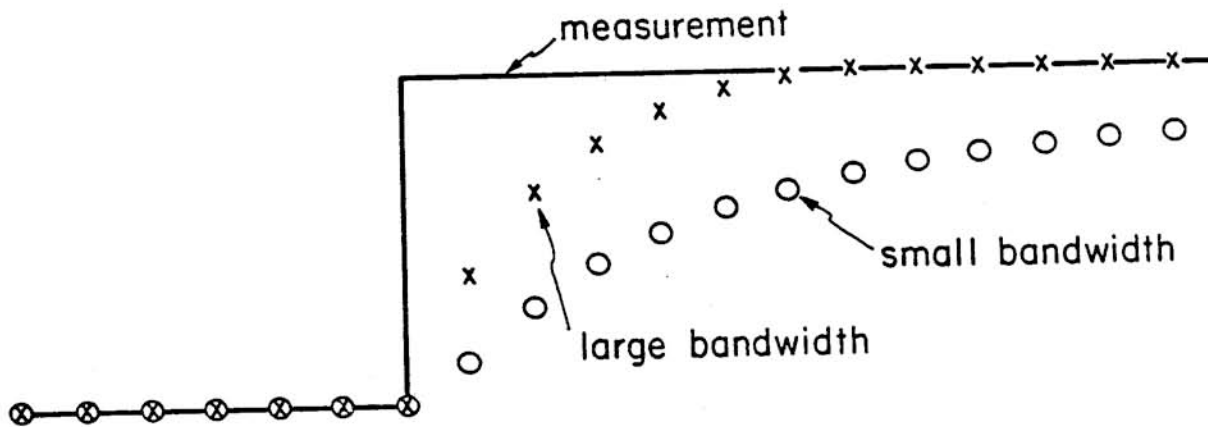


Figure 4.1-1 Estimates Tracking Measurements

Previous experience in other failure detection problems has indicated that MM is reasonably robust to such discrepancies since a real velocity sensor failure will generally look more like a velocity bias failure to .9 than like a position bias failure. We will give some examples of this robustness later in this chapter.

Five sets of models with different plant and noise covariances were used. The nominal value for plant covariance used was

$$Q_c = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad Q_d = \begin{bmatrix} \frac{1}{3} \times 10^{-5} & \frac{1}{2} \times 10^{-4} \\ \frac{1}{2} \times 10^{-4} & 10^{-1} \end{bmatrix}$$

where Q_c is the continuous plant covariance and Q_d is the equivalent discrete time plant covariance used in the digital implementation on the computer. The standard deviations of the position sensor noise and velocity sensor noise were assumed to be $\frac{1}{2}$ m and 0.1 m/sec respectively, yielding a sensor noise covariance of

$$R = \begin{bmatrix} .25 & 0 \\ 0 & .01 \end{bmatrix}$$

which is the same for both continuous and discrete time. Other sets were obtained by reducing and increasing the sensor noise covariance by a factor of 10 and by similarly reducing and increasing the plant noise covariance.

These sets were used on the odometer and speedometer failure scenarios and the maneuver and grade scenarios. All scenarios were based on vehicle A

using the PI controller.* The failure scenarios were used to determine how quickly the various sets of models detect the failures. The maneuver and grade scenarios were used to study the effects of discrepancies between predicted and measured vehicle behavior due to modeling approximations. Sensor noise was not added in any of these scenarios so as not to obscure the issue being examined with random effects. Thus we were primarily interested in these initial tests in determining the effect on MM response time of different assumed noise statistics. The effects of sensor noise present in the simulation are examined separately later in this section.

Results are summarized in Table 4.1-1. The grade scenario is not listed because the grade caused practically no discrepancy between commanded and actual behavior; the PI controller quickly compensated for the grade because of its large bandwidth. The maneuver also had no adverse effects on the failure detection algorithm. It was adequately compensated for even with the reduced plant covariance. Because it had no adverse effects the time it took for the probabilities of the failure models to reach the minimum probability of 1.0×10^{-7} from their initial probability of .05 is given. Time is measured in the table by the number of discrete time steps. Throughout this thesis a discrete time step of 0.1 sec is used. For the failures, the number of discrete time steps given in the table are measured from the onset of the failure. When tested on failures, probabilities for the failure models were initialized to p_{\min}

*The parameters which characterize vehicle A are given in Table 3.1-1. The PI controller is discussed in Section 3.1 and a block diagram is given in Figure 3.1-3.

Table 4.1-1 Effects of Q and R on Failure Detection

Model Set	Scenario		
	Maneuver	Position Bias +1	Velocity Gain .9
nominal	P (odometer failures)* reach min in 9 steps P (speedometer gain .9) reaches min in 6 steps P (speedometer gain 1.1) reaches min in 8 steps	detected in 10 steps (2σ failure)	detected in 7 steps (15σ failure)
reduced sensor covariance	P (odometer failures) reach min in 1 step P (speedometer gain .9) reaches min in 1 step P (speedometer gain 1.1) reach min in 4 steps	immediate detection (6σ failure)	detected in 3 steps (45σ failure)
increased sensor covariance	P (odometer failures) decrease to approx. .001 after 50 steps P (speedometer failures) reach min in approx. 17 steps	P (position bias +1) increases to approx. 1.7×10^{-6} in 20 steps ($\frac{2}{3}\sigma$ failure)	detected in 20 steps (5σ failure)
reduced plant covariance	P (odometer failures) reach min in 8 steps P (speedometer failures) reach min in 1 step	detected in 10 steps (2σ failure)	immediate detection (15σ failure)
increased plant covariance	P (odometer failures) reach min in 13 steps P (speedometer failures) reach min in 10 steps	P (position bias +1) increases to .27 in 20 steps (2σ failure)	detected in 10 steps (15σ failure)

*P (odometer failures) means the probability of the odometer failure models

so that at the time of the failure the failure models would have a correct state estimate. Failures are considered detected when the corresponding model has the maximum probability. Ratios of failure magnitude to sensor noise standard deviation are given in parentheses.

These results will be explained with the aid of Table 4.1-2 which gives the residual covariances and modeled sensor noise covariance for the unfailed model of each set. Residual covariances for the models of velocity gain change failures differed slightly from that of the unfailed model of the same set. Residual covariances for the models of the position sensor bias failures were identical with that of the unfailed model of the same set.

The first point to note is that the reduced sensor covariance set detects the position sensor bias significantly faster than the nominal set whereas the velocity sensor gain change is not detected significantly faster. This is because the nominal set attributes position residuals to poor measurements whereas it attributes velocity residuals to poor predictions. Comparison of R and Σ of the nominal set shows that the position residual covariance is nearly equal to the odometer noise covariance but that the velocity residual covariance is much larger than the speedometer noise covariance. When the modeled sensor covariance is reduced the position residual covariance decreases significantly but the velocity residual covariance decreases slightly. Because the position residuals are expected to be smaller, large residuals are penalized more heavily in the probability calculation so that the unfailed model's probability decreases quickly and the probability of the +1 m odometer bias model increases

Table 4.1-2 Residual and Sensor Noise Covariances

Model Set	Σ	R
nominal	$\begin{bmatrix} .257 & .006 \\ .006 & .119 \end{bmatrix}$	$\begin{bmatrix} .25 & 0 \\ 0 & .01 \end{bmatrix}$
reduced sensor covariance	$\begin{bmatrix} .027 & .005 \\ .005 & .102 \end{bmatrix}$	$\begin{bmatrix} .025 & 0 \\ 0 & .001 \end{bmatrix}$
increased sensor covariance	$\begin{bmatrix} 2.55 & .018 \\ .018 & .262 \end{bmatrix}$	$\begin{bmatrix} 2.5 & 0 \\ 0 & .1 \end{bmatrix}$
reduced plant covariance	$\begin{bmatrix} .255 & .002 \\ .002 & .026 \end{bmatrix}$	$\begin{bmatrix} .25 & 0 \\ 0 & .01 \end{bmatrix}$
increased plant covariance	$\begin{bmatrix} .268 & .051 \\ .051 & 1.02 \end{bmatrix}$	$\begin{bmatrix} .25 & 0 \\ 0 & .01 \end{bmatrix}$

quickly. The velocity sensor failure is detected slightly more quickly because the velocity residual covariance decreases only slightly.

Similar conclusions are reached by comparing the nominal set's performance with the reduced plant covariance set's performance. In this case the speedometer failure is detected more quickly whereas there is no improvement in detecting the odometer failure. Reducing the modeled plant covariance significantly reduced the velocity residual covariance, allowing the speedometer failure to be detected more quickly; the position residual covariance was reduced marginally so that the odometer failure was not detected more quickly.

For the increased sensor covariance set the increase in R was large enough to increase both the position and velocity residual covariance so that performance in detecting both failures was degraded. However, the position residual covariance increased so drastically that the bias in the position sensor is essentially assumed to be noise.

If the foregoing arguments were applied to the increased plant covariance set, one would expect a slight increase in the time required to detect the odometer failure because of the slight increase in the position residual covariance. Also, it would be expected that the speedometer failure would be detected much more slowly because of the large increase in velocity residual covariance. However, effects due to bandwidth come into play so that the odometer failure takes twice as long to detect and the speedometer failure takes only slightly longer to detect. The odometer failure will be explained first.

Increasing the plant covariance increases the bandwidth of the Kalman filter because the model is trusted less. This causes the filter to track

the measurements more quickly as illustrated previously in Figure 4.1-1. The unfailed model of the increased plant covariance set was able to track the erroneous position measurement more quickly than the unfailed model of the nominal set. This is indicated by its more quickly decaying position residuals illustrated in Figure 4.1-2. Because its residuals decreased more quickly, the unfailed model of the increased plant covariance set does not become improbable as quickly.

The increase in bandwidth also causes the velocity measurements to be tracked more quickly. The unfailed model tracks the sudden decrease in velocity so that its velocity residuals become small quickly. However, because the unfailed model believes the vehicle is actually going slower it predicts that a shorter distance will be traveled than actually will be traveled. The model of the velocity sensor with a gain of .9 is able to accurately predict the distance traveled. The position residuals of the unfailed model grow because of its constant underestimation of the vehicle's velocity. It is because of increasing position residuals that the unfailed model is rejected. The model of the 0.9 gain speedometer is selected because of its more accurate position predictions. Because velocity sensor failures are also detected through position residuals the substantial increase in time required to detect the failure that might have been expected is not seen.

Addition of Simulated Sensor Noise

The previous results show that by reducing Q and R failures can be detected more quickly. However, the modeled noise covariances should not be reduced without consideration of the noise that is actually present in

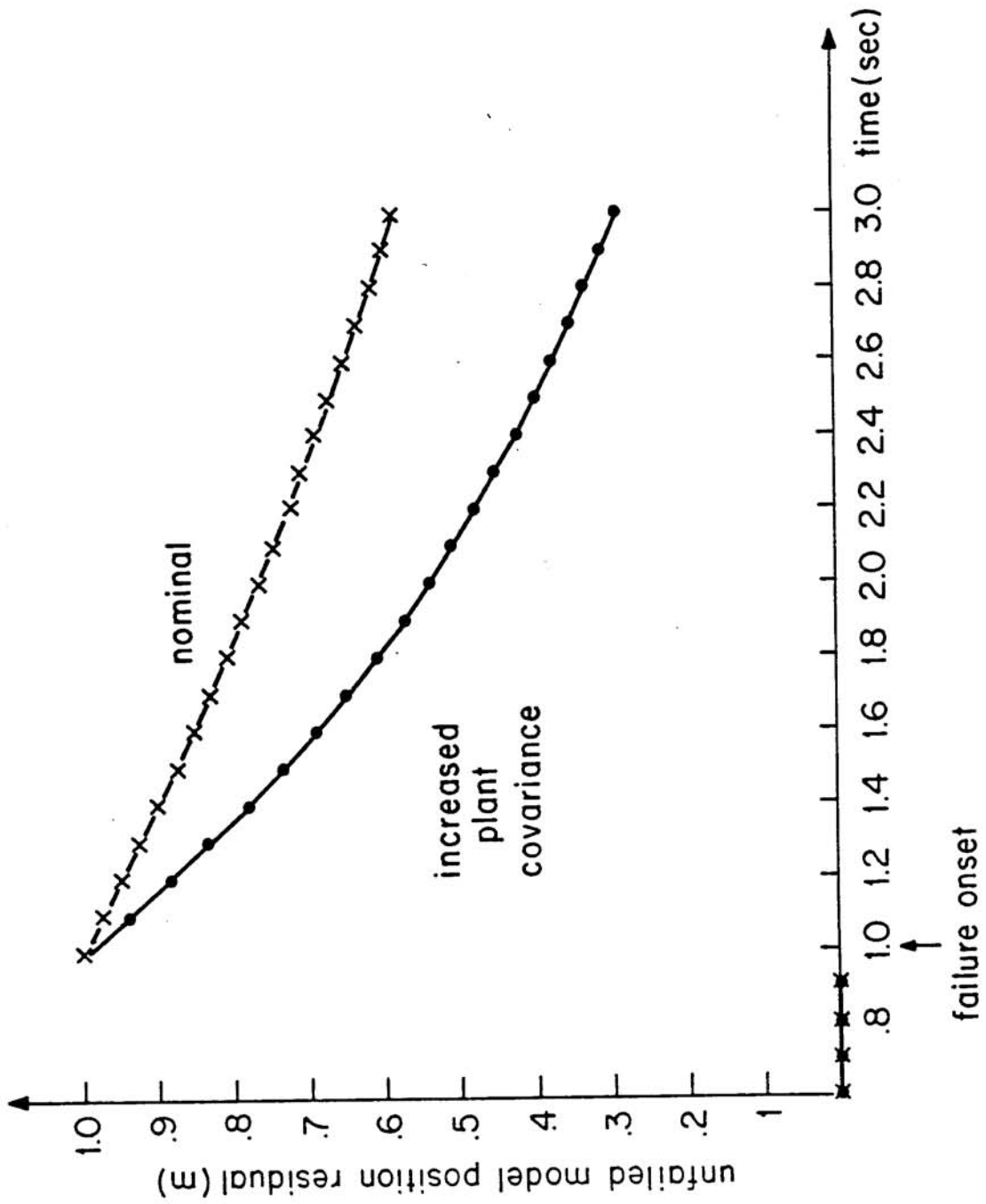


Figure 4.1-2 Effect of Bandwidth on Position Residuals

the system. The noise that is actually present must be accounted for with sufficiently large Q and R matrices or else it can cause false alarms or other degradations in failure detection performance.

Some of the pitfalls of inadequately compensating for noise were demonstrated using scenarios similar to those used previously with noise added to the simulated sensor outputs. The simulated sensor noise had a covariance ten times larger than the nominal modeled sensor noise covariance. The reduced plant noise covariance set was tested on these scenarios because its modeled plant covariance adequately accounted for the modeling approximations and provided for good failure detection performance.

Figure 4.1-3 illustrates how the probabilities behaved when no noise was present in the odometer failure scenario. The initial probabilities for the failure models were .05 and they decreased quickly to the minimum. The initial probability for the unfailed model was .8 and it quickly approached unity. Several steps after the failure occurs there is a smooth transition where the probability of the +1 m odometer bias model approaches unity and the unfailed model's probability decreases to the minimum.

Figure 4.1-4 illustrates how the probabilities behave when excessive simulated sensor noise is added in the scenario. During the one second before the onset of the failure there are problems in locking onto the unfailed model. At first a negative odometer bias is indicated; then a positive odometer bias is indicated. At the time of the failure a velocity sensor gain of 0.9 is indicated. After the failure the probabilities oscillate wildly between the positive odometer bias model and the unfailed

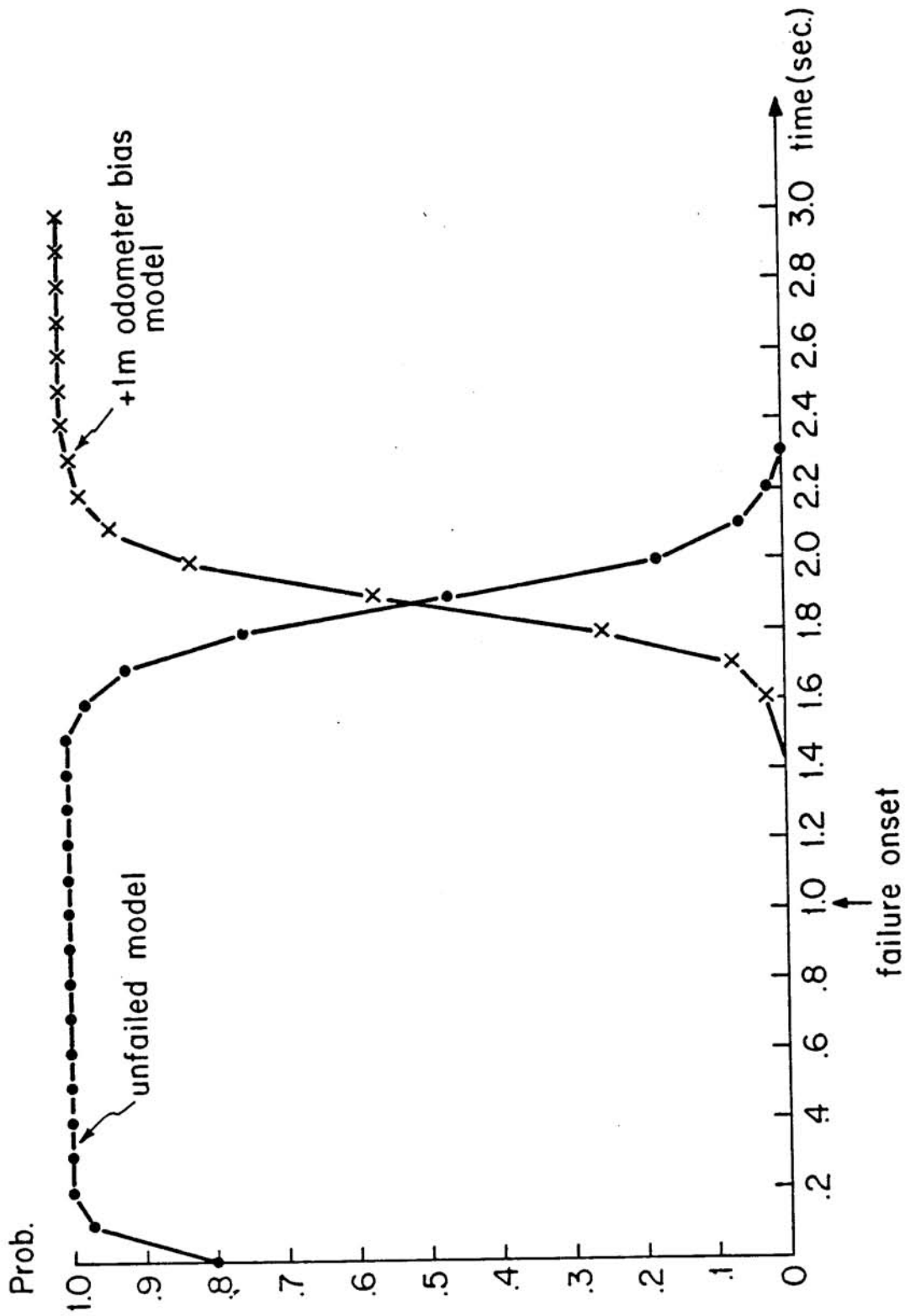


Figure 4.1-3 Probabilities for Odometer Bias Scenario--No Noise

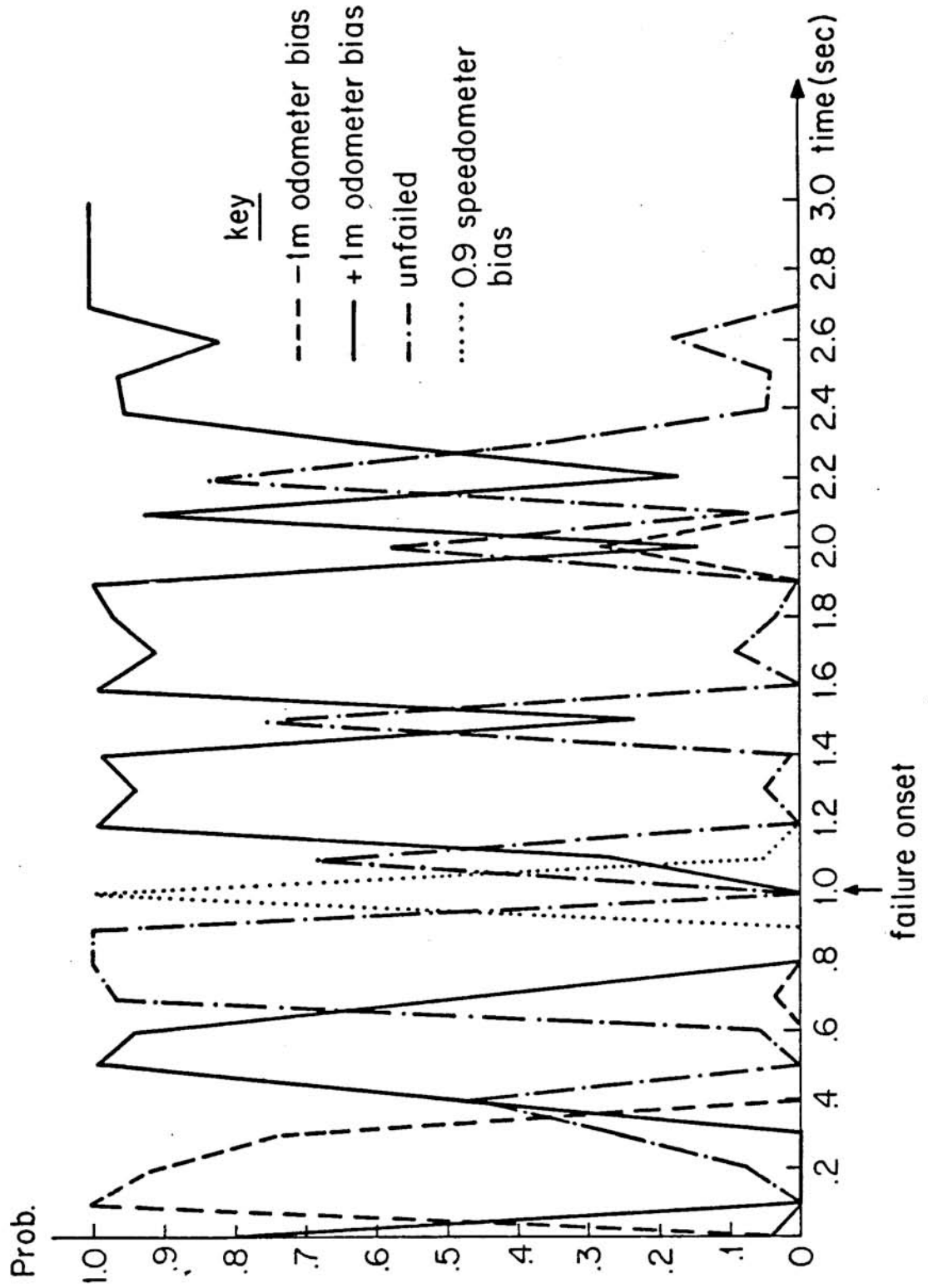


Figure 4.1-4 Probabilities for Odometer Bias Scenario--With Noise

model but finally the odometer bias model is indicated.

When the same noise sequence is added to the speedometer failure scenario the failure is incorrectly identified. This is illustrated in Figure 4.1-5. The same initial chattering of the probabilities was present for this scenario as in Figure 4.1-4 so the first second of data is omitted. Initially the correct model is indicated but the noise causes the 0.9 speedometer gain model to be rejected. For a while the unfailed model is indicated but because it underestimates the true velocity it also underestimates the vehicle's position. The measured position grows larger than the estimated position so that eventually it appears that there is an odometer bias of +1 m.

The results of this section demonstrate that noise must be adequately accounted for with sufficiently large Q and R matrices. By limiting how small Q and R can be, the noise limits how quickly failures can be detected. Minimal amounts of plant and sensor noise are desired. By comparing R and Σ it can be determined whether reducing plant noise or reducing sensor noise will result in the greater improvement in detection speed. If the modeled Q and R correspond to the noise actually present in the system then this comparison of R and Σ indicates whether a more accurate model or more accurate sensors will allow quicker detection of failures.

4.2 Comparison of Failure Detection Models

The results of the tests of the various models are summarized in Table 4.2-1. The scenarios were all based on vehicle B using the LQ controller. The grade scenario was not used because grade and wind have similar effects on the vehicle and wind is the more significant disturbance

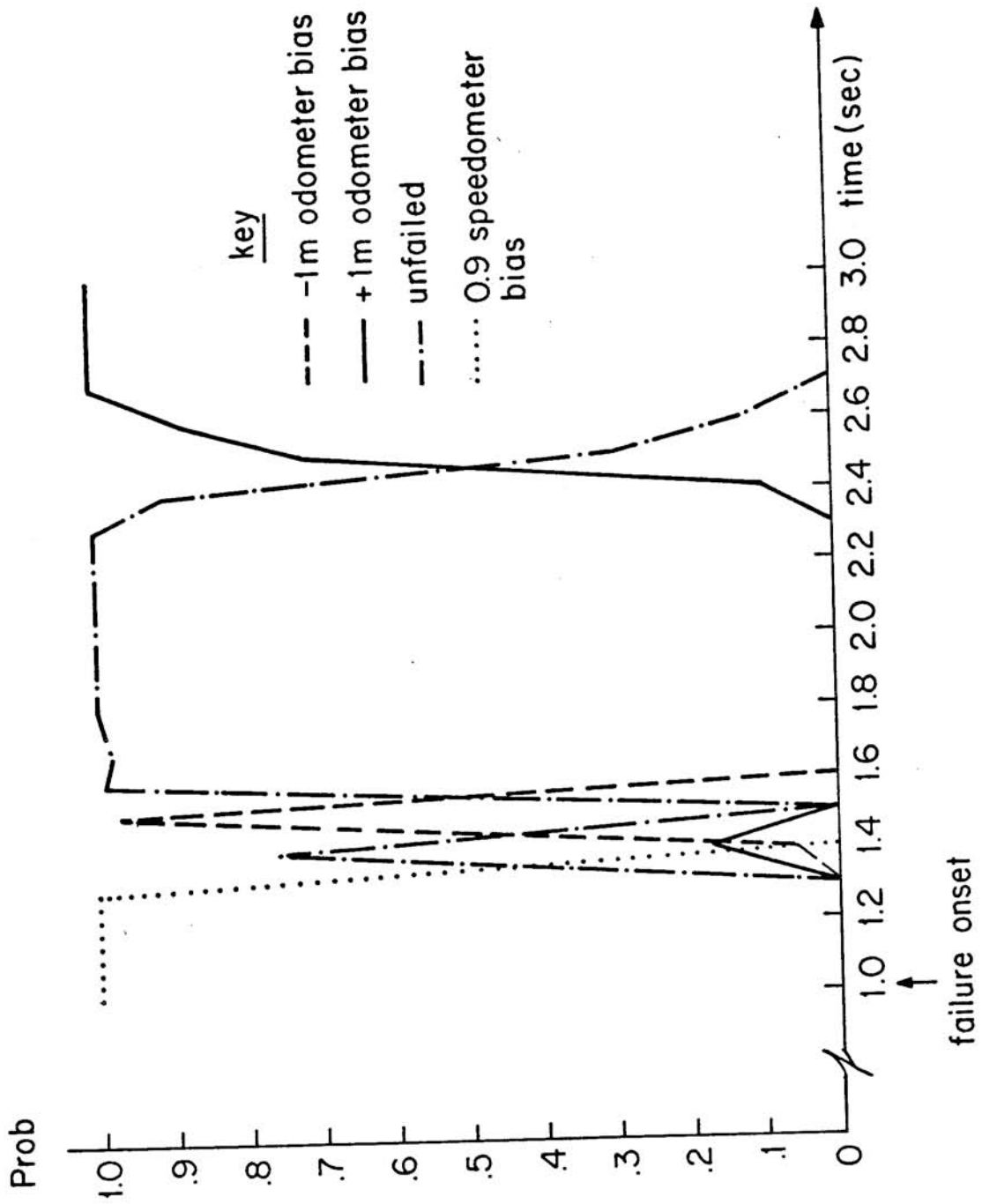


Figure 4.1-5 Probabilities for Speedometer Gain Change Scenario--With Noise

Table 4.2-1 Comparison of Failure Detection Model Performance

Scenario	Model Set			P3
	KC2	KS2	KS1	
maneuver	no effect	.9 accelerometer gain probability increases to .02 while jerk > 0	no effect	no effect
wind	no effect	NF	NF	no effect
+1m odometer bias	detected in one step	detected in one step	detected in one step	detected in one step
.9 speedometer gain	immediate detection	immediate detection	detected in 5 steps; +1m odometer bias probability increases slightly	immediate detection
.9 accelerometer gain	ND	detected in 12 steps	ND	ND
1.1 accelerometer gain	ND	detected in 23 steps	ND	ND
1.1 PCU gain	no failure models indicated	ND	ND	detected in 2 steps
maneuver heavy vehicle	NF	NF	NF	no effect

ND: given failure is not detectable by that set of models

NF: given scenario is not a factor that would cause false alarms for that set of models

for vehicle B (compare the forces given in Table 4-1). No noise was added in the simulations so as not to obscure the other factors influencing performance. All models used a standard deviation of 0.2 m (8 in) for modeled odometer noise and .1 m/sec (.22 mi/hr) for modeled velocity sensor noise.

Results are discussed on a model by model basis.

KC2 - Kinematic Control Driven Model - 2 State

This model is extensively discussed in other sections and is listed here primarily for comparison. The continuous modeled plant noise covariance used for this set is

$$Q_c = \begin{bmatrix} 0 & 0 \\ 0 & 0.1 \end{bmatrix}$$

The PCU failure is discussed in Section 4.3 with the other unmodeled failures.

One point that should be noted is that the maneuver and wind scenarios do not cause false alarms or even an increase in any of the failure model probabilities even though the LQ controller was used instead of the PI controller. Because of the smaller bandwidth of the LQ controller the vehicle's measured velocity does not track the commanded velocity as well as with the large bandwidth PI controller. The LQ controller allows larger discrepancies between commanded and true velocity so that there are larger discrepancies between commanded and true acceleration. This would imply that a larger modeled plant noise covariance would be needed, but the covariance used adequately accounted for the discrepancies so that no effects on the model probabilities were observed.

KS1 - Kinematic Sensor Driven Model - 1 State

This model allows failures to be detected by integrating the measured velocity for comparison with the measured position. It does not attempt to approximate vehicle dynamics as does KC2, and it does not require an accelerometer that is required by KS2. KS1 allows velocity sensor gain changes and position sensor biases to be modeled.

Because velocity measurements are made discretely the modeled plant noise covariance was not computed in the way that the covariance for KC2 was computed where the driving term is continuous. Instead the noise was assumed to enter the discretized state equations directly:

$$s(k+1) = s(k) + 0.1(v(k) + w(k))$$

where $w(k)$ has the covariance of the speedometer noise. The plant noise is $0.1w(k)$ so that the plant noise covariance is

$$(0.1)^2 \text{cov}(w(k)) = 0.0001$$

This model is extremely simple but its simplicity limits its performance in detecting certain failures. Because it checks only for agreement between the odometer and speedometer it is unaffected by any discrepancies between commanded and measured vehicle behavior regardless of whether these discrepancies are caused by controller failures or wind disturbances. A more important limitation is that it requires longer to detect the speedometer gain change than either KC2 or KS2, even though the velocity sensor is assumed to be equally accurate for all three models. This is because the predicted distance computed by integrating the erroneous velocity

measurement diverges slowly from the true distance indicated by the unfailed odometer. This is illustrated in Figure 4.2-1. Small deviations between predicted and measured position could be caused by noise so the algorithm waits until there is a significant difference between them before making a decision. Models KC2 and KS2 can detect the speedometer gain change more quickly because they have a predicted velocity with which the measured velocity is compared so that the discrepancy is immediately obvious.

The probability of the +1 m odometer bias increases slightly when the speedometer fails because the measured position is larger than the position predicted by integrating the measured velocity. This is symptomatic when failure detection is done by comparing two sensors. There is uncertainty in whether one has failed by indicating a false large quantity or if the other has failed by indicating a false large quantity. This did not prove to be a serious problem in this case. At no time was the +1 m odometer bias model ever more probable than the 0.9 speedometer gain model, and its probability decreased when the discrepancy between predicted and measured position grew larger than 1 m.

Another point to note, which is characteristic of all gain change failures, is that the velocity sensor gain change failure will take longer to detect with smaller velocities. If the true velocity is v and the measured velocity is $.9v$ then the difference between them is

$$v - 0.9v = .1v$$

At lower velocities this difference is smaller so that more data must be

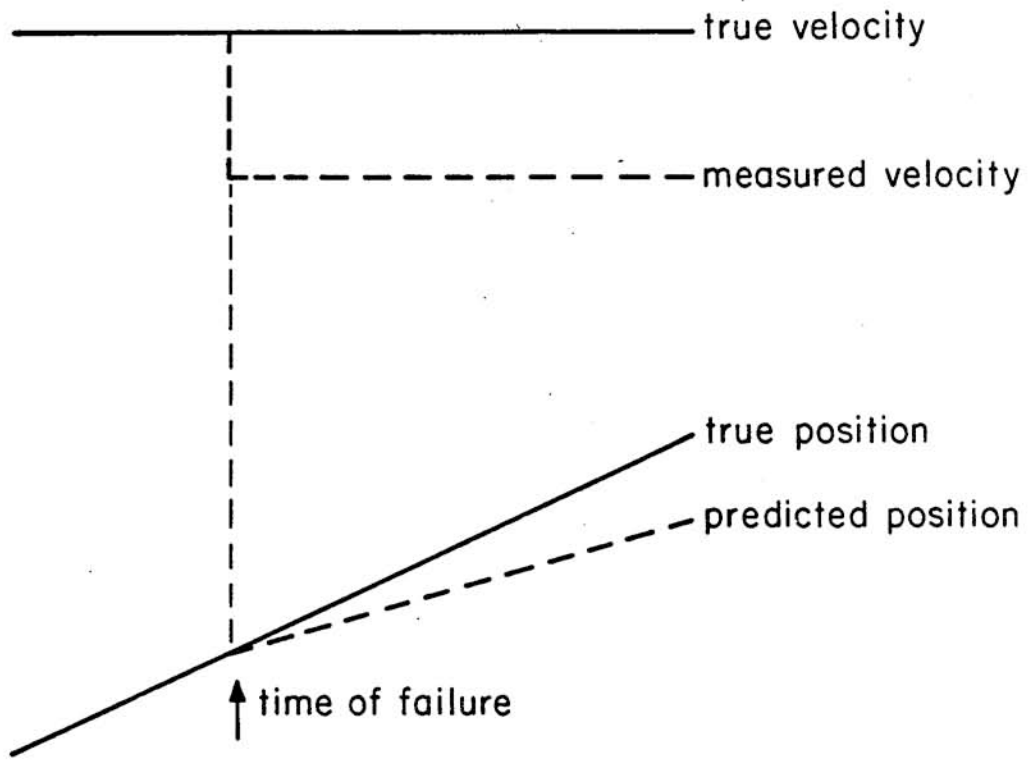


Figure 4.2-1 Divergence of Position Predicted from Incorrect Measured Velocity

collected to distinguish between failed and unfailed. However, this may not be a serious shortcoming because if the vehicle is traveling slowly an extremely quick detection will probably not be required.

KS2 - Kinematic Sensor Driven Model - 2 State

This model is identical to KC2 except that instead of using commanded acceleration as the driving term, measured acceleration is used. This avoids the approximation made by KC2 that the commanded acceleration is close to the true acceleration. In addition to the failures modeled by KC2, accelerometer gain changes to .9 and 1.1 are also modeled by altering the B matrix. The standard deviation of the modeled accelerometer noise used is $.0125 \text{ m/sec}^2$. The modeled plant noise covariance was computed using the method used for KS1.

KS2 performs similarly to KC2 in detecting the odometer and speedometer failures. However, there is a drastic variation in the speed of detecting the accelerometer failures, and a slight sensitivity to the maneuver is displayed. The same cause is responsible for both effects and will be explained by first discussing the results obtained using the maneuver scenario.

During the one second before the maneuver begins, the probabilities of the accelerometer failure models do not decrease from their initial probability of 0.05 because while the acceleration is zero they are indistinguishable from the unfailed model. However, as the acceleration increases it would be expected that the probabilities of the accelerometer failure models would decrease, but the probability of the .9 accelerometer gain increases slightly. This increasing probability is due to discretization

error, the error made by assuming that the control is constant over each discrete time interval. Figure 4.2-2 shows the relationship between the true and discretized acceleration. The probability of the .9 accelerometer gain increases because it indicates when true acceleration is larger than measured acceleration. As illustrated by Figure 4.2-2, the true acceleration actually is larger than the acceleration assumed by the discrete time model. When these two curves are integrated to obtain velocity, the velocity obtained from the discretized curve is smaller. The error caused by discretization is small, but the failure magnitude of this gain change failure is also small because of the small acceleration. Because the discretization error produces a discrepancy similar to that which would be caused by an accelerometer with a gain of 0.9, the probability of the failure model increases.

The discretization error causes the drastic difference in times required to detect the two different accelerometer failures. As the vehicle accelerates in the 0.9 accelerometer gain change scenario, the probability of the 0.9 accelerometer gain model increases in part due to discretization error causing a relatively quick detection. This same error causes the probability of the 1.1 accelerometer gain model to be smaller than it should be, so that even when an accelerometer gain change to 1.1 has occurred it takes longer to detect.

One way to alleviate this problem is to increase the modeled plant noise covariance. This would help account for the "noise" due to discretization but it would also degrade performance in detecting accelerometer failures. A relatively large amount of time is already required to detect the accelerometer failures even though the probability of the

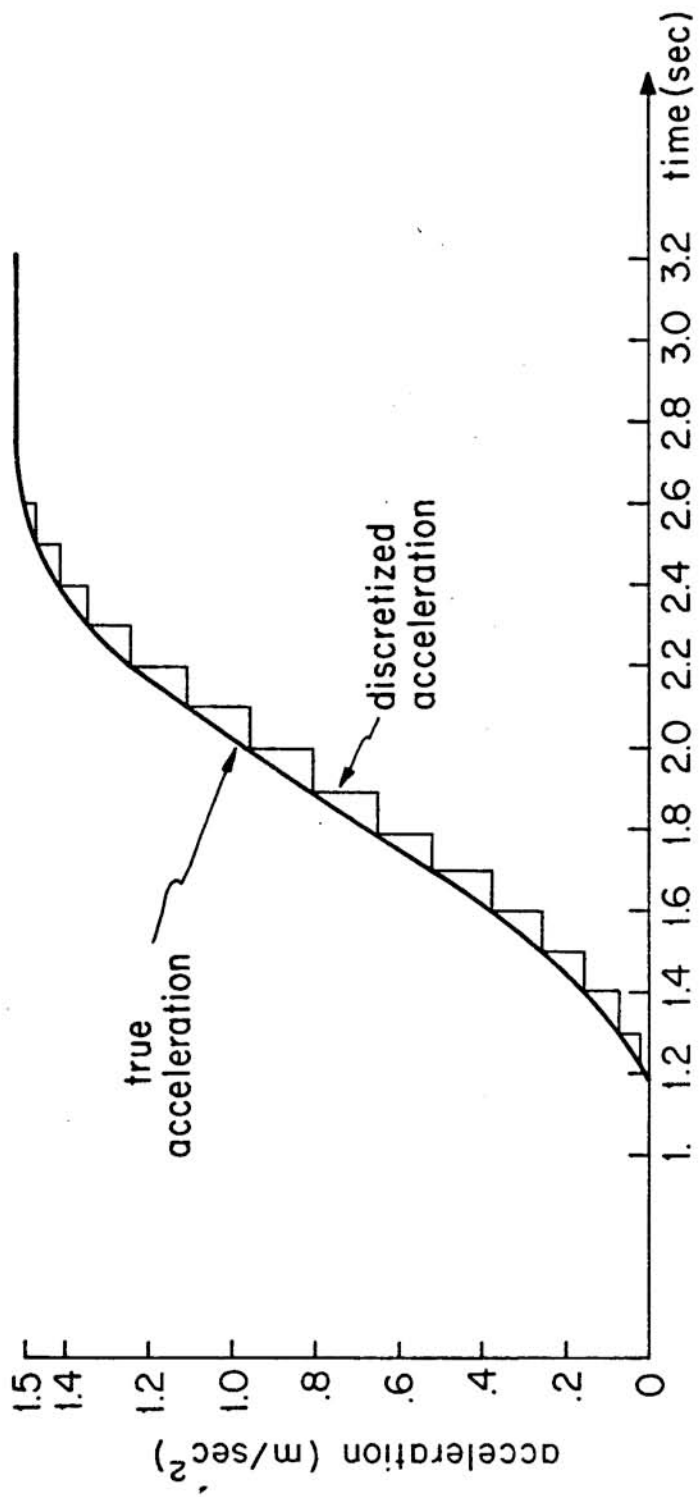


Figure 4.2-2 True Versus Discretized Acceleration

accelerometer failure models began considerably higher than the minimum probability. A further increase in time required for detection because of a larger Q would not be desirable.

Another way to alleviate this problem is to alter the model so that true acceleration is better approximated. This could be done by using the kinematic equations

$$s(k+1) = s(k) + \Delta t v(k) + \frac{\Delta t^2}{2} a(k) + \frac{\Delta t^3}{6} j(k)$$

$$v(k+1) = v(k) + \Delta t a(k) + \frac{\Delta t^2}{2} j(k)$$

where $j(k)$ is the jerk and Δt is the discrete time interval. KS2 ignores the terms due to jerk, but jerk can be approximated by

$$j(k) \approx \frac{a(k) - a(k-1)}{\Delta t}$$

This would lead to the state equations

$$\begin{bmatrix} s(k+1) \\ v(k+1) \\ a(k) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & -\frac{\Delta t^2}{6} & s(k) \\ 0 & 1 & -\frac{\Delta t^2}{2} & v(k) \\ 0 & 0 & 0 & a(k-1) \end{bmatrix} + \begin{bmatrix} \frac{2}{3}\Delta t^2 \\ \frac{3}{2}\Delta t^2 \\ 1 \end{bmatrix} a(k)$$

P3 - Parameterized Model - 3 State

This model attempts to approximate the dynamics of the vehicle more closely than KC2 by explicitly including a model of the vehicle's electric

motor. The state equations are of the form

$$\begin{bmatrix} \dot{s}(t) \\ \dot{v}(t) \\ \dot{a}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -C_0 & -C_1 \end{bmatrix} \begin{bmatrix} s(t) \\ v(t) \\ a(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K_M \end{bmatrix} E_C(t)$$

where $E_C(t)$ is the commanded voltage and C_0 , C_1 , and K_M are constants dependent on vehicle parameters. The continuous plant noise covariance used was

$$Q_C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 10 \end{bmatrix}$$

which can be obtained by assuming a noisy commanded voltage with a standard deviation of $\sqrt{10}/K_M \approx 0.2$ volts.

This model performed well in detecting all failures and the maneuver and wind caused no fluctuation in the probabilities. Discretization error had no effect because the failure magnitudes of the PCU gain change failures was much larger than the discretization error. KS2 had problems with discretization error because the failure magnitudes of the .9 accelerometer gain was on the order of the discretization error.

Because this model depends on vehicle parameters it was tested on a scenario based on a vehicle using different parameters. The most easily identifiable and largest varying parameter is the vehicle's total mass, which varies with the vehicle's load. The mass was increased to the maximum for vehicle B given in Table 3.1-1 to generate the scenario data. The

variation in mass only affects transient behavior and not the steady state relationship of $v = (K_M/C_0)E$ so the heavy vehicle was put through a maneuver. The variation in mass did not cause the probabilities to differ significantly from those in the maneuver scenario using nominal vehicle mass.

The results of this section indicate that KC2 would be a good model to use in the MM algorithm. KS1 imposes a smaller computational burden, i.e., it requires fewer multiplications and additions per time step than KC2, but it requires more time to detect the speedometer failure. Also, KC2 does not require the accelerometer that is needed by KS2. Unlike P3 it did not detect the PCU failure but this failure does not have a drastic impact on safety because it is compensated for by the LQ controller. However, the dynamics of an actual vehicle could differ from those of the simulated vehicle. It is possible that a more detailed model such as P3 could be required to account for the dynamics.

4.3 Detection of Unmodeled Failures

The performance of the MM algorithm in detecting failures not explicitly modeled was examined using the reduced plant covariance set of Section 4.1. The unmodeled failures simulated and the performance of the algorithm is summarized in Table 4.3-1. Most of the simulations were based on vehicle A using the PI controller. The one exception is the PCU failure* which was simulated on vehicle B using the LQ controller because the LQ controller did not compensate for the PCU failure as quickly so that the PCU failure is more detectable. Performance in detecting modeled failures is given at the bottom of the table for comparison.

*The Power Conditioning Unit failure is described in Table 3.1-1.

Table 4.3-1 Summary of Unmodeled Failure Detection Performance

Failure	Performance
+ $\frac{1}{2}$ m odometer bias	no failure models indicated
+ $\frac{3}{4}$ m odometer bias	+1 m odometer bias model probability increases to 0.45 after 25 steps unfailed model probability decreases to 0.55
0.95 speedometer gain	+1 m odometer bias indicated in 18 steps
0.925 speedometer gain	0.9 speedometer gain indicated in 11 steps
+1 m/sec speedometer bias	1.1 speedometer gain indicated in 13 steps
1.1 PCU gain (LQG controller)	no failure models indicated
+1 m odometer bias	detected in 10 steps
.9 speedometer gain	detected immediately

The position bias of $\frac{1}{2}$ m was not detected because it was exactly halfway between the unfailed model and the +1 m odometer bias model. After the failure occurs the position residuals of the two models are equal in magnitude but opposite in sign. Because both models have the same residual covariance their $p_i(k)$ are identical. The two models also have the same Kalman gain so they both track the failure at the same rate as illustrated in Figure 4.3-1 so that their $p_i(k)$ remain the same. The probability of the position sensor bias of +1 m does not increase because its $p_i(k)$ must be larger than that of the unfailed model for it to increase. This can be seen by writing equation (4) from Chapter 2 as

$$P_2(k) = \frac{p_2(k)P_2(k-1)}{p_1(k)P_1(k-1)} \approx \frac{p_2(k)}{p_1(k)} P_2(k-1)$$

using subscript 1 to indicate the unfailed model and subscript 2 to indicate the +1 m odometer bias model. This equation assumes that all terms in the denominator other than that for the unfailed model are negligibly small and that $P_1(k-1) \approx 1$. An intuitive explanation for this is that because present evidence indicates the models are equally probable the decision is made on the basis of past evidence which indicated that the unfailed model is the correct one.

A larger bias tips the balance between the two models in favor of the +1 m odometer bias model as demonstrated with the simulated $+\frac{3}{4}$ m position sensor bias.

The velocity sensor gain change to 0.95 was not detected as a speedometer failure for reasons similar to those given for why the odometer bias

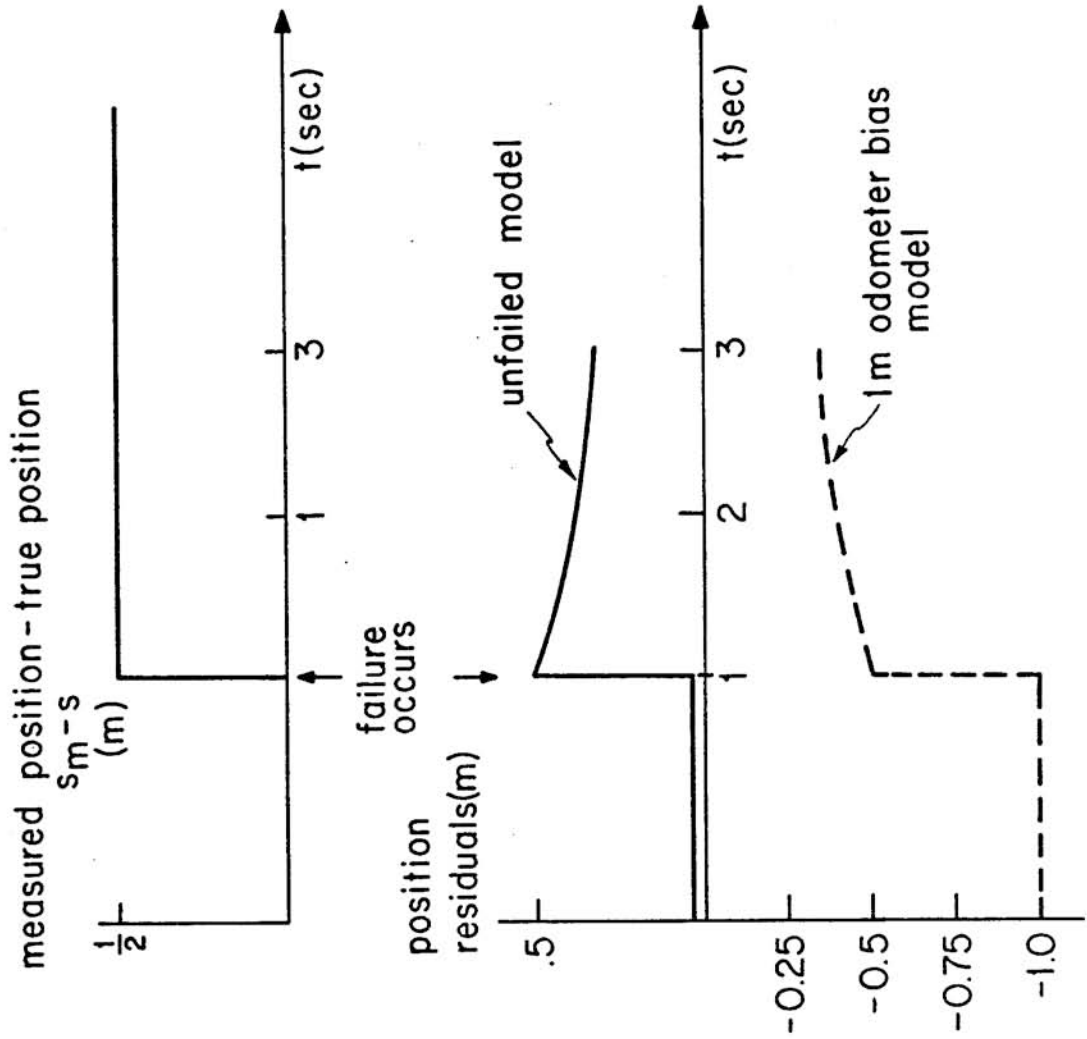


Figure 4.3-1 Position Residuals-- $\frac{1}{2}$ m Odometer Bias Scenario

of $+1\frac{1}{2}$ m was not detected. The simulated failure was halfway between the unfailed model and the 0.9 speedometer gain model. However, a position sensor bias of +1 m was indicated because the models tracked the erroneously low velocity measurements. This caused the predicted position to increase more slowly than the measured position. The discrepancy between measured and predicted position results in the +1 m odometer bias model being indicated. A speedometer gain to 0.925 is detected as a speedometer failure because it is closer to the 0.9 speedometer gain model than the unfailed model.

Speedometer biases can cause speedometer gain change models to be indicated as demonstrated with the +1 m/sec speedometer bias scenario. The +1 m/sec speedometer bias causes the 1.1 speedometer gain to be indicated because both failure and model say that the true velocity is less than the measured velocity. If the measured velocity is a constant 15 m/sec the 1.1 speedometer gain model believes the true velocity to be 13.6 m/sec. As long as the measured velocity remains constant this is equivalent to saying that there is a +1.4 m/sec bias in the measured velocity. The +1 m/sec speedometer bias is relatively close to this model so that the 1.1 speedometer gain is indicated.

The PCU gain change to 1.1 was simulated using the LQ controller because the PI controller would have quickly corrected for the failure, making it much more difficult for KC2 to detect. There was some possibility that KC2 could detect this failure because it would result in an increased measured velocity even though no increase was commanded. Because steady state velocity is directly proportional to voltage the PCU failure should

be similar to a 1.1 speedometer gain. However, the dynamics of the vehicle do not permit the velocity of the vehicle to change quickly in response to this failure so that it is easily tracked by the models. Also, it does not cause a discrepancy between measured position and measured velocity, which helps speedometer gain change models to be indicated. It is possible that by reducing the modeled plant covariance the models will not track the measurement as quickly, permitting this failure to be detected. However, reducing the modeled plant covariance can make the set of models sensitive to wind and grade. A downgrade also causes an increase in measured velocity even though no increase is commanded.

If the MM algorithm is to be used primarily as an alarm to indicate that something has gone wrong, then its ability to detect failures not explicitly modeled is a positive quality. When the speedometer's gain changed to 0.95, the +1 m odometer bias model was selected, indicating that something has gone wrong and corrective action needs to be taken. However, in a fully automated system more than an alarm may be required and an accurate identification of the failure could be crucial. The action taken by the controller when a +1 m odometer bias is indicated may not be appropriate if a speedometer gain change to .95 caused the bias to be indicated.

4.4 Redundant Sensor Configurations

Factors influencing the improvement in failure detection through duplicating sensors were examined using KC2 with two odometers and two speedometers. For this sensor configuration the measurement equation becomes

$$\begin{bmatrix} s_{m1} \\ s_{m2} \\ v_{m1} \\ v_{m2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s \\ v \end{bmatrix}$$

Using the nominal standard deviations for sensor noise given in Section 4.1 yields a sensor noise covariance of

$$R = \begin{bmatrix} .25 & 0 & 0 & 0 \\ 0 & .25 & 0 & 0 \\ 0 & 0 & .01 & 0 \\ 0 & 0 & 0 & .01 \end{bmatrix}$$

The nominal plant noise covariance matrix of Section 4.1 was also used. These modeled noise covariances were used so that the failure detection performance of this configuration could be compared directly with the performance of the nominal set of Section 4.1, a set whose performance left room for improvement. The increased number of sensors requires a greater number of models because of the greater number of failures possible. The models used were

- 1 unfailed
- 2,3 position sensor #1 bias -1m, +1m
- 4,5 position sensor #2 bias -1m, +1m
- 6,7 velocity sensor #1 gain .9, 1.1
- 8,9 velocity sensor #2 gain .9, 1.1

This set of models was tested on the same failure scenarios used in Section 4.1 and performed similarly to the set of models using the reduced plant noise of Section 4.1. The bias of +1 m in position sensor #1 was detected in nine steps and gain change to 0.9 in velocity sensor #1 was detected immediately as compared with the 10 steps and seven steps, respectively, required by the nominal set of models.

The reason for this lies in the fact that both sets of models attribute position residuals primarily to sensor errors whereas velocity residuals are attributed primarily to prediction errors. The nominal set's delay in detecting the velocity sensor failure is primarily due to uncertainty in the prediction, the standard against which the velocity sensor is compared. Because of their accuracy, adding another velocity sensor to the prediction yields a relatively accurate standard against which the other velocity sensor can be compared. This improvement in the standard for comparison for the reduced plant covariance set of Section 4.1 was accomplished by reducing the modeled plant covariance. For the position sensors, the nominal set already considers its standard for comparison relatively accurate; the delay arises because the position sensor is not expected to agree with the standard. The addition of another relatively inaccurate position sensor does not add a significant amount of information about what the other sensor should be indicating.

There is a problem that can arise because of dual sensors that did not prove to be a significant effect for this set of models. The problem is that in comparing two sensors, one indicating a larger quantity than the other, it is difficult to tell whether one has failed by indicating

an erroneous large quantity or if the other has failed by indicating an erroneous small quantity. There was a trace of this indecision in detecting the velocity gain change but the prediction was accurate enough to serve as a decisive third vote. In detecting the gain change to .9 in velocity sensor #1, the probability of a gain change to 1.1 in velocity sensor #2 did rise slightly above the minimum probability to a maximum of 4.8×10^{-5} but returned to the minimum quickly. If the modeled plant noise were larger, the predicted velocity would not be considered as accurate so that this indecision would have been more apparent.

Dual sensors not only improve detection speed but also help prevent false alarms due to modeling errors. For example, if the dynamics of the vehicle causes the true velocity to differ significantly from the commanded velocity this discrepancy could be mistaken for a speedometer failure if a single speedometer is used. However, if two speedometers are used and they both disagree with the commanded velocity but agree with each other it is much less likely that a failure has occurred in either one of them. This effect would help prevent the PCU failure from appearing to be a speedometer failure if dual speedometers are used.

The results presented in this section indicate that triplicating sensors would only marginally improve failure detection performance in this case. The position sensors are relatively inaccurate and provide little additional information beyond that provided by the position estimate. Two velocity sensors allow immediate detection of speedometer failures already so that no improvement in detection speed would be obtained by triplicating them. However, if the modeled plant covariance were larger,

an additional speedometer would help. With a larger Q the predicted velocity would be trusted less so that there would be a greater indecision when a speedometer fails, resulting in a delay if only two are used. A third speedometer would allow immediate detection by permitting the algorithm to essentially use the three speedometers for voting.

5. Failure Detection and Control for Vehicle Strings

The application of the MM algorithm to strings of vehicles is described in three sections of this chapter. In the first section the response of a string of vehicles to various failures is examined to determine the impact of these failures on the system when they are not detected and no corrective action is taken. In the second section models that allow a vehicle to detect errors in measurements it has concerning other vehicles in addition to errors in its own measurements are examined. The performance of the algorithm using these models is also examined in the second section. In the third section the use of the probabilities and estimates generated by the algorithm to determine a proper control for a vehicle in a string is discussed.

5.1 Effects of Failures Without Failure Detection

String response was examined for failures in sensors that supply measurements to the controller. These failures were selected because they are apt to have more significant effects than other failures. String behavior is also affected by other failures, such as the PCU gain change, but it is possible that the controller will be able to determine an appropriate response because it still has accurate information about how the vehicle is moving.

The control law discussed in Chapter 3 generates commands based on three measurements, spacing, Δs_m , preceding vehicle velocity, v_{1m} , and trailing vehicle velocity, v_{2m} . Throughout this chapter it is assumed that the preceding vehicle's own velocity measurements are transmitted to

the trailing vehicle because in [25] it is stated that measurements of relative velocity are unlikely. The equation describing the control law is given here explicitly indicating how these measurements are used to generate the velocity command.

$$v_c = v_{2m} + \frac{\Delta s_m - \Delta s_{\min}(v_{1m}, v_{2m})}{\left. \frac{\partial}{\partial v_2} \Delta s_{\min}(v_1, v_2) \right|_{v_{1m}, v_{2m}}}$$

where

$$\Delta s_{\min}(v_1, v_2) = c_1(v_1^2 - v_2^2) + c_2(v_1 - v_2) + c_3(v_2) + c_4$$

Values of c_1 , c_2 , c_3 and c_4 used in this research are $\frac{1}{3}$, .15, .467 and 0.66, respectively. These values are those used in [25] where this control law is developed. Essentially this controller attempts to keep the trailing vehicle at a minimum safe spacing, $\Delta s_{\min}(v_1, v_2)$, by adjusting its velocity proportionally to the error between measured and desired spacing. The failures examined were a bias of +1m in the spacing sensor, a gain change to 0.9 in the trailing vehicle's speedometer, and a gain change to 1.1 in the preceding vehicle's speedometer.

The short term effects of the failures were examined by simulating the string's behavior for periods up to four seconds (40 time steps) after the failure occurred. This amount of time was considered sufficient because the relatively quick detections demonstrated by the algorithm for single vehicles indicate that the detection algorithm can respond to the failures within this period. The simulations all began with both

vehicles traveling at a constant 15 m/sec with the desired spacing of 7.665 m between them. The preceding vehicle's velocity command, v_{1c} , is kept at a constant 15 m/sec throughout the simulations. Spacing and relative velocity, $\Delta v = v_1 - v_2$, are plotted in Figures 5.1-1 and 5.1-2 for the three failures.

All three failures cause the trailing vehicle to follow more closely than the desired minimum spacing, but of the three the preceding speedometer failure causes the most dangerous response. This failure causes the preceding vehicle to slow down because its measured velocity suddenly becomes larger than the commanded velocity. Although the spacing starts decreasing, the trailing vehicle increases its velocity instead of slowing down because it attempts to match its speed to that indicated by the preceding vehicle. This failure is also the most severe in that unlike the other failures it causes the trailing vehicle to violate the normal operation constraint on acceleration specified by Draper in their design of the controller [25].

An examination of the controller equations reveals that both velocity gain change failures will lead to a collision but the spacing sensor bias will only cause the trailing vehicle to track the preceding more closely. This was determined by computing a steady state spacing using the controller equations and two additional relations. First, in steady state the measured velocities should equal the velocity commands,

$$v_{1c} = v_{1m}$$

$$v_{2c} = v_{2m}$$

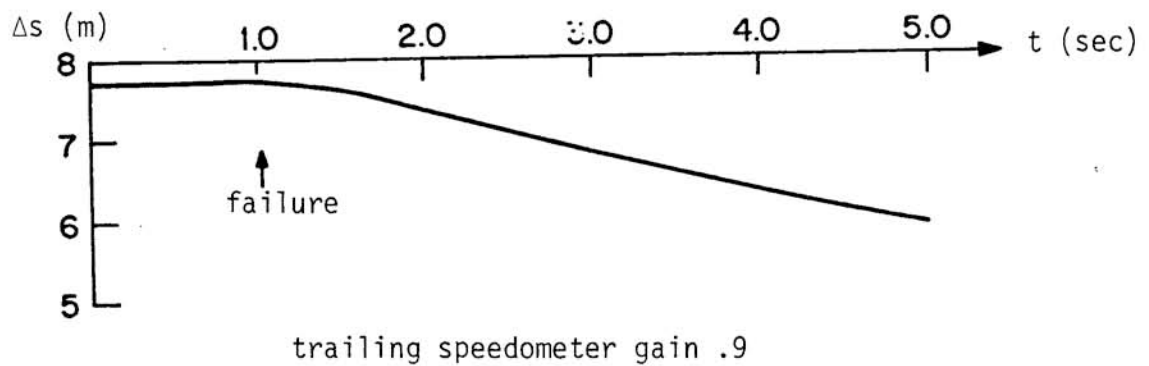
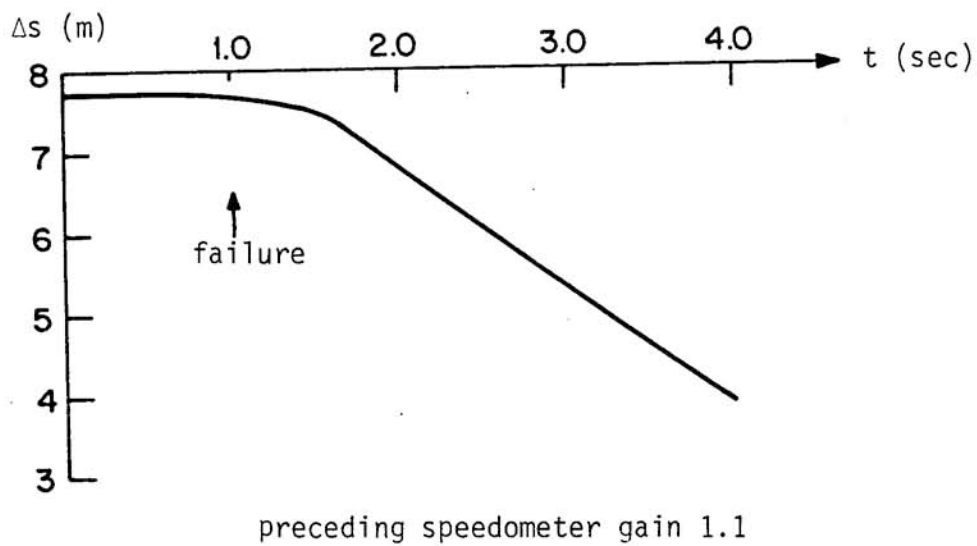
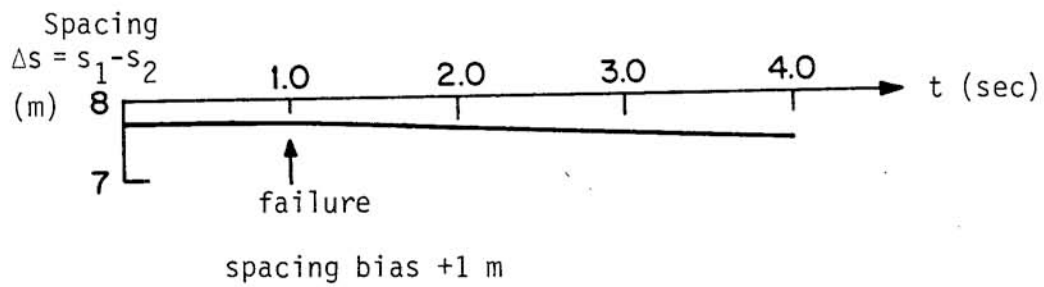


Figure 5.1-1 Spacing After Various Failures

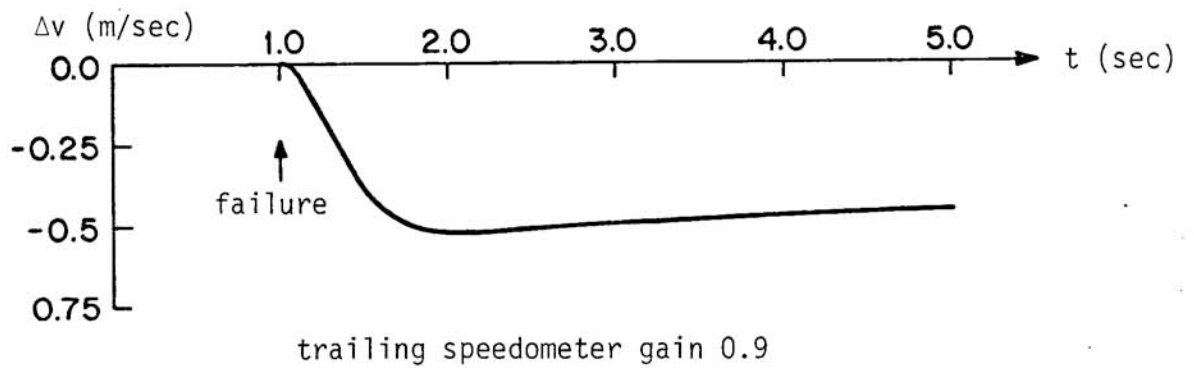
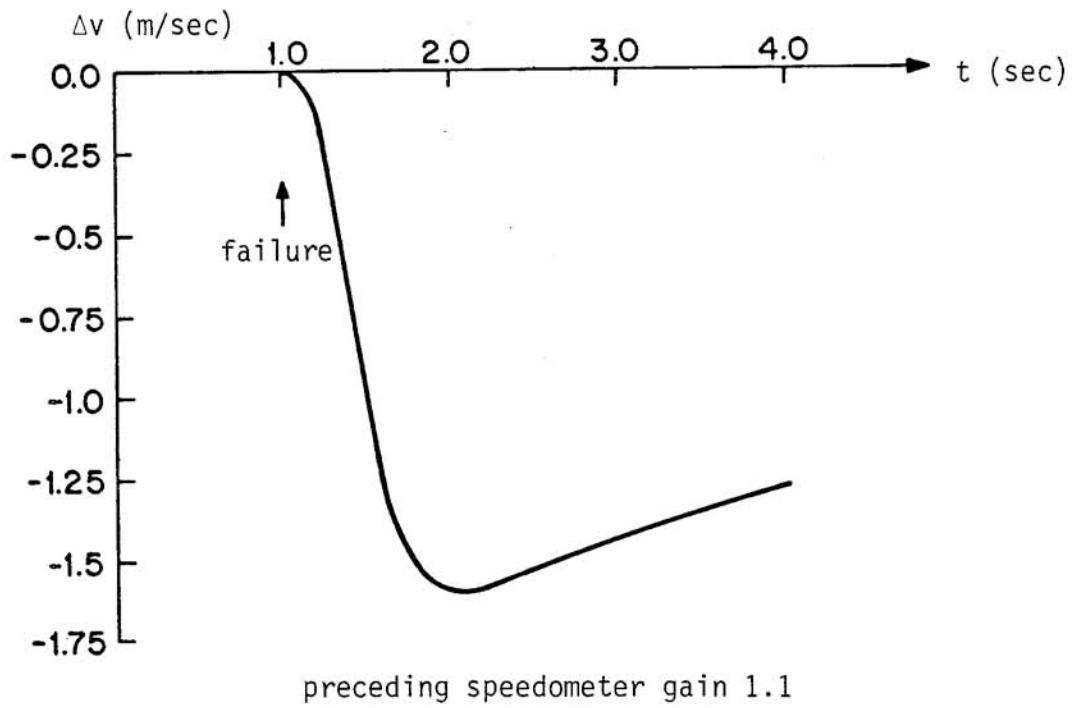
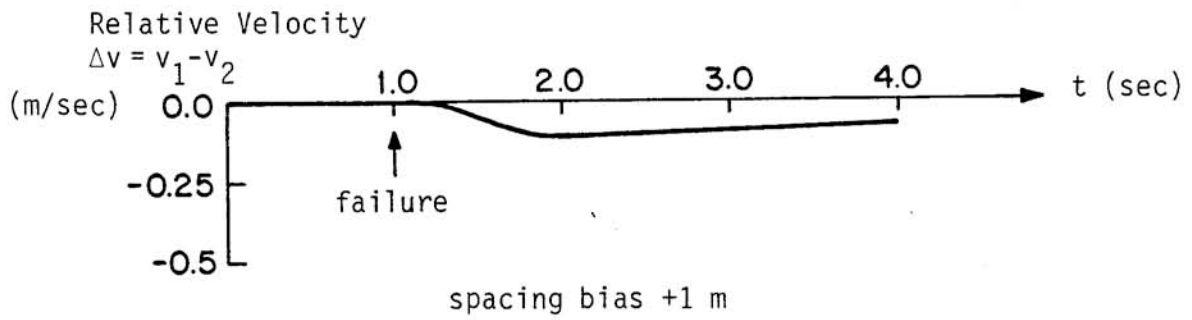


Figure 5.1-2 Relative Velocity After Various Failures

For the trailing vehicle this implies measured and desired spacing are equal,

$$\Delta s_m = \Delta s_{\min}(v_{1m}, v_{2m})$$

Second, in steady state the true velocity of both vehicles should be equal,

$$v_1 = v_2$$

The relationship between measured and actual quantities is determined by the failures suffered by the sensors. For example, for the preceding speedometer gain change the equations are

$$\Delta s_m = \Delta s$$

$$v_{1m} = 1.1v_1$$

$$v_{2m} = v_2$$

These equations can be easily solved to determine Δs as a function of the steady state velocity. Plots of Δs versus velocity are given in Figure 5.1-3 for various failures. Negative steady state spacings indicate that a collision will occur.

These curves indicate that unless corrective action is taken the speedometer gain change failures will lead to a collision. The spacing sensor bias will not necessarily cause a collision. The minimum spacing, Δs_{\min} , was determined as the minimum spacing required to avoid a collision if the preceding vehicle decelerates at operational limits. By decelerating

steady state spacing

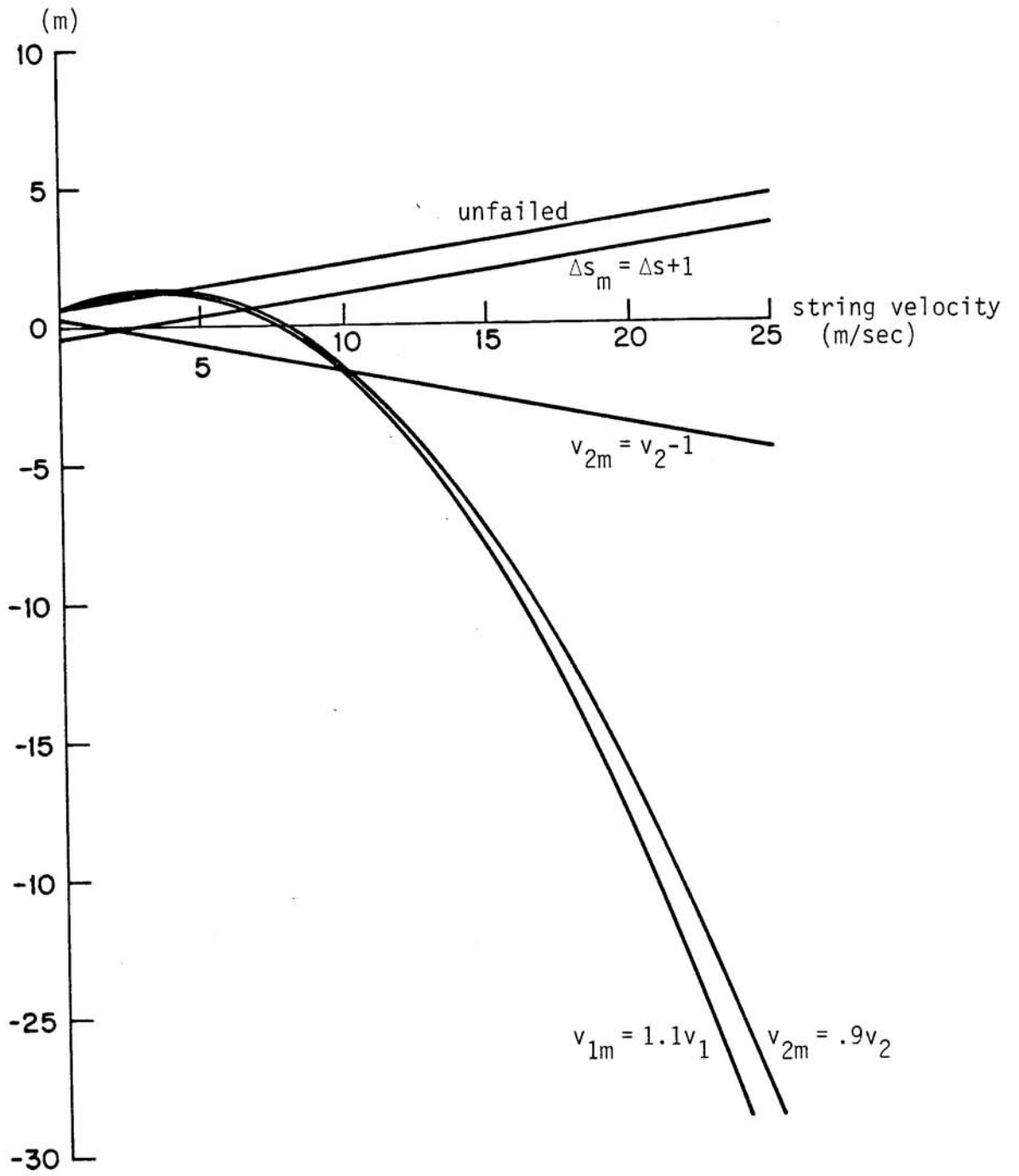


Figure 5.1-3 Steady State Spacing for Various Failures

slowly a collision can be avoided for the spacing sensor failure.

The results of this section indicate that unless these failures are detected and corrective action taken that safety is jeopardized. The two speedometer failures will lead to a collision and the preceding speedometer failure causes an especially severe response in the trailing vehicle. The ill effects of the preceding speedometer failure could be avoided by using the MM algorithm to generate estimates of the preceding vehicle's velocity. However, these estimates would heavily depend on spacing measurements so that spacing sensor failures would become critical.

5.2 Failure Detection Models and Performance

In this section the ability of a vehicle to detect errors in measurements it has concerning other vehicles is examined. In addition to the spacing and preceding velocity measurements required by the controller, preceding position measurements were also used to determine how they might improve failure detection performance if available. Models were examined using various subsets of these measurements to determine how well the algorithm performed with minimum amounts of information. Dual sensor configurations were not examined but the performance of these configurations can be predicted based on the results of Section 4.4. Noise covariances assumed for the various sensors were $.04 \text{ m}^2$ for the odometers and the spacing sensor and $.01 \text{ m}^2/\text{sec}^2$ for the speedometers.

The models examined in this section consist of a single vehicle model used by the trailing vehicle to describe itself, augmented by equations describing the preceding vehicle. The two state kinematic model was used to describe the trailing vehicle. A measure of the trailing

vehicle's true acceleration was used to drive the model and a driving noise of covariance $0.1 \text{ m}^2/\text{sec}^4$ was assumed. If applied to an actual vehicle the true acceleration can be approximated with either commanded or measured acceleration, and thus one of these two quantities is used in the model. The issues involved in which of these are used are not of concern here. The primary concern is how well the trailing vehicle can detect errors in the information required by its controller.

Two different ways were used to describe the preceding vehicle in the dual vehicle models. The first way consisted of essentially using KS1 to describe the preceding vehicle. With measurements of spacing and trailing position and velocity the model is given by the equations

$$\begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{1m} \\ a_2 \end{bmatrix}$$

$$\begin{bmatrix} \Delta s_m \\ s_{2m} \\ v_{2m} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ v_2 \end{bmatrix}$$

D3--Dual vehicle model - 3 state

Spacing residuals affect both preceding and trailing position estimates, and the error covariance of both these estimates affects the ability to detect spacing sensor failures. This coupling has relatively minor effects as will be discussed later. This model was also examined using preceding position measurements in addition the other measurements to determine how this additional sensor could improve failure detection performance. The

set of failures modeled consisted of spacing sensor and odometer biases of ± 1 m and speedometer gains of .9 and 1.1.

The second way in which the preceding vehicle was described did not require measurements of the preceding vehicle's velocity. This was examined to determine how well spacing sensor failures can be detected without velocity measurements and to determine how well relative velocity can be estimated using spacing measurements. By estimating relative velocity, the preceding velocity can be easily estimated as required by the controller if measurements of it are unavailable. The measurements may be unavailable due to a failure or they may be deliberately avoided to avoid the severe effects of the preceding speedometer failure. The state equations used were

$$\begin{bmatrix} \dot{s}_1 \\ \Delta \dot{v} \\ \dot{s}_2 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_1 \\ \Delta v \\ s_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} a_2$$

D4--Dual vehicle model - 4 state

The preceding vehicle's velocity, \dot{s}_1 , is modeled as being equal to the trailing vehicle's velocity plus a correction term, Δv , the relative velocity. Because the vehicles are traveling in a string their velocities should be approximately equal and the relative velocity near zero. This allows Δv to be modeled as a zero mean noise process. The trailing vehicle does not know how Δv evolves so $\Delta \dot{v}$ is modeled as white noise.

Modeling $\Delta \dot{v}$ as white noise is unrealistic in the sense that this model states that the covariance of Δv grows without bound. The relative

velocity is known to remain within certain bounds and a Gauss-Markov model such as

$$\dot{\Delta v}(t) = -\alpha\Delta v(t) + w(t)$$

where $w(t)$ is white noise and α a positive constant could be used to indicate that Δv 's variance is bounded. However, a positive value for α would degrade the Kalman filter's ability to estimate the relative velocity. Figure 5.2-1 illustrates relative velocity for a typical maneuver and how it would be estimated with a positive α . The preceding vehicle decelerates at jerk and acceleration limits and the trailing vehicle attempts to track it. The magnitude of the relative velocity is underestimated because the model predicts the relative velocity will return to zero. Using an α of zero will allow the estimates to better track the true relative velocity.

The covariance of $\dot{\Delta v}$ was selected by examining how quickly Δv could change under normal circumstances. The quickest relative velocity changed in a simulation was when the trailing vehicle tracked the preceding vehicle decelerating at maximum jerk and acceleration. Wind and grade do not cause as great a variation in the preceding vehicle's velocity and also act on the trailing vehicle causing it to similarly change its velocity. The biggest change in relative velocity over any .1 sec time step during this maneuver scenario was about .075 m/sec indicating a relative acceleration of approximately .75 m/sec². Relative acceleration was squared and rounded to one significant digit yielding a covariance of 0.6 m²/sec⁴.

The two sets of measurement equations used with D4 were essentially

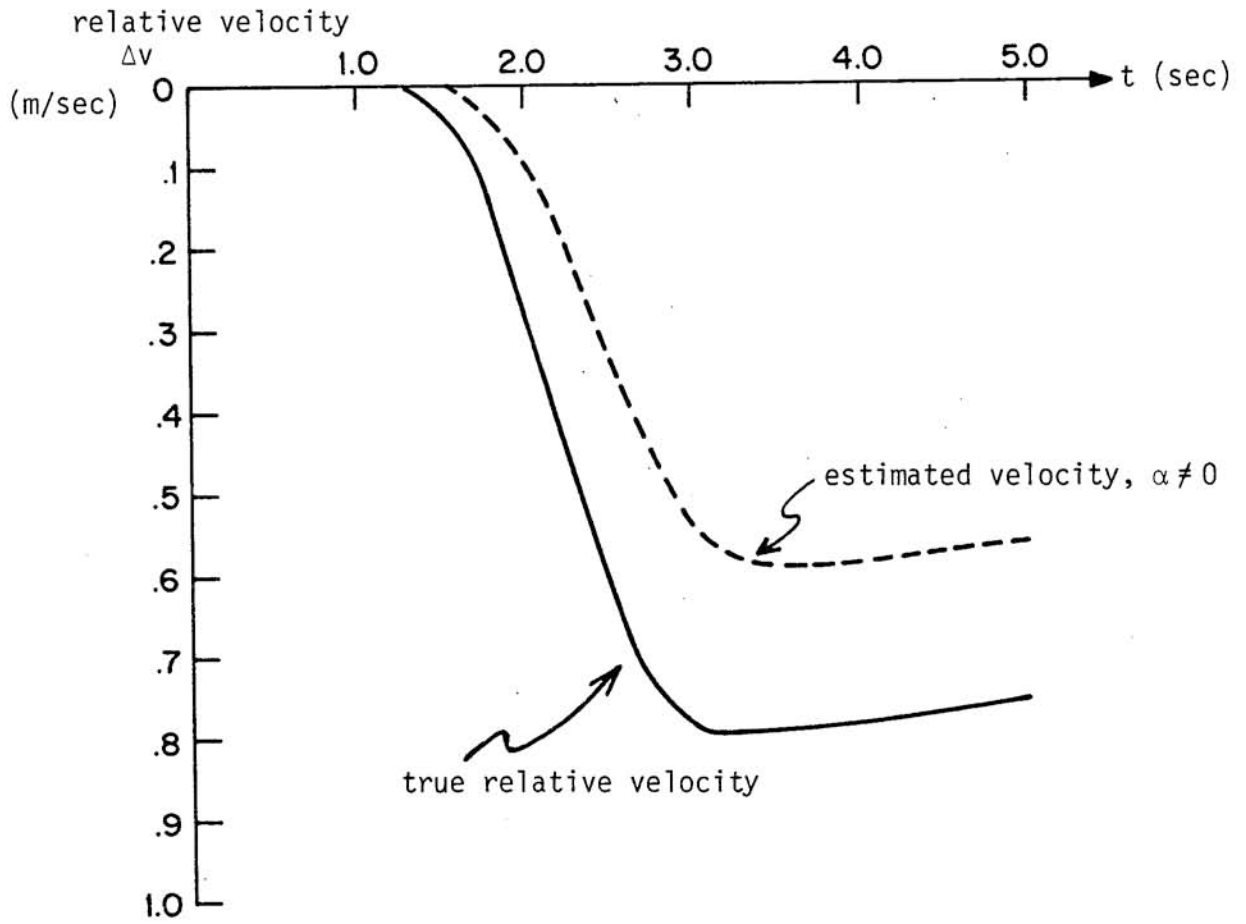


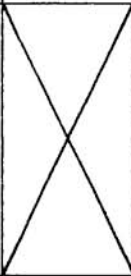
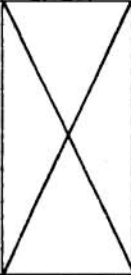

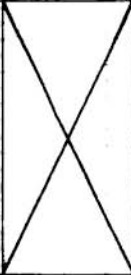
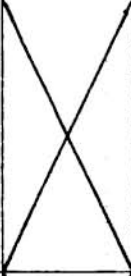

Figure 5.2-1 Relative Velocity Underestimated by a Gauss-Markov Model

the same as those used with D3. One set modeled measurements of spacing and trailing vehicle position and velocity. The other set consisted of the same measurements with an additional measurement of preceding vehicle position. The same set of failures was modeled using D4 as was with D3 except for the preceding vehicle speedometer failures. D4 without the preceding position measurements was also examined with a driving noise covariance of $.06 \text{ m}^2/\text{sec}^4$ for the Wiener process because of its inability to detect spacing sensor failures when using a covariance of $.6 \text{ m}^2/\text{sec}^4$.

The results of the simulations are summarized in Table 5.2-1. The deceleration maneuver where the preceding vehicle decelerates at jerk and acceleration limits was the only scenario used to test for false alarms. Spacing and relative velocity change more quickly for this scenario than with other scenarios so that the models should have the most difficulty adjusting their state estimates to account for these changes. Wind and grade were not used because they would not cause any discrepancies between predictions and measurements for these models. The kinematic relationships between the inputs and outputs of these models are valid independent of the amount of wind or grade.

The trailing odometer bias and the trailing speedometer bias were detected by all the dual vehicle models the same as they were detected by the single vehicle model which was not augmented by the description of the preceding vehicle. The accuracy of the trailing velocity estimates and the low noise of the trailing speedometer allow its failures to be quickly and unambiguously identified. It might have been expected that the odometer bias could have been more quickly detected because of the additional

Table 5.2-1 Multiple Vehicle Failure Detection Performance

Scenario	D3 without s_{1m}	D3 with s_{1m}	D4 without s_{1m}	D4 with s_{1m}	D4 reduced Δv covariance
deceleration maneuver	no effect	no effect	P (spacing errors) don't reach min	no effect	no effect
spacing sensor bias +1 m	detected 1 step	detected 1 step	P (spacing bias +1 m) increases to .034	detected 2 steps	detected 2 steps unfailed doesn't reach min
preceding odometer bias +1 m		detected 1 step		detected 2 steps	
trailing odometer bias +1 m	detected 1 step	detected 1 step	detected 1 step	detected 1 step	detected 1 step
preceding speedometer gain 1.1	detected 6 steps (rel. dist. bias +1?)	detected 5 steps			
trailing speedometer gain 0.9	immediate detection	immediate detection	immediate detection	immediate detection	immediate detection

information about the trailing vehicle's position contained in the spacing sensor. The additional information did cause a decrease in the s_2 residual covariance but this decrease was marginal so that there was no noticeable improvement in performance. A typical decrease was from .0422 for the single vehicle model to .0419 for D3 without s_{1m} .

The performance of D3 in detecting the spacing sensor and preceding speedometer failures is almost identical to that of KS1 in detecting the odometer and speedometer failures. The only difference is that without the preceding position measurement, D3 takes 1 time step longer to detect the speedometer failure than KS1. This increase in time to detect is due to an increase in bandwidth; without preceding position measurements D3 is unable to estimate the preceding position as accurately as KS1.

Spacing and velocity do not yield as much information concerning the preceding vehicle's position as position and velocity did concerning the single vehicle's position because spacing depends on an additional uncertain quantity, the trailing vehicle's position. Because the preceding vehicle's position estimates are more inaccurate they track variations in the spacing measurements more closely so that the effects of the failure are not as readily apparent.

The initial selection of plant covariance led to poor failure detection performance when D4 was used without preceding position measurements. The plant covariance of $0.6 \text{ m}^2/\text{sec}^4$ was large enough to create relatively large bandwidth filters so that the state estimates tracked the spacing sensor bias. The bias caused the probability of the corresponding model to increase from the minimum probability to only .034.

The addition of the preceding position measurements greatly improves failure detection performance. The bandwidth is still large but the additional sensor allows another comparison to be made. When the unfailed model attempts to decrease the spacing residuals caused by the bias, it increases the preceding position estimate. However, this causes the preceding position estimate to diverge from the preceding position measurement. The unfailed model is unable to eliminate both spacing and preceding position residuals and therefore becomes improbable.

Decreasing the plant covariance to $0.06 \text{ m}^2/\text{sec}^4$ also improved failure detection performance. The decrease reduces the bandwidth so that the spacing measurements are not tracked as quickly and the effects of the failure are more apparent. The covariance could be decreased more in an attempt to further increase the detection speed but could make the algorithm sluggish to less abrupt changes in measured spacing which occur in normal operation such as during the deceleration maneuver.

The performance of D4 in estimating the relative velocity is examined in the next section where the effects of using these estimates in the control law are also discussed.

The deceleration maneuver tests the models at the limits of normal operational behavior, and as can be seen, no false alarm problems are observed. However, there is a case when a vehicle exceeds these limits and a failure should not be indicated. As discussed in Section 5.1, when the preceding speedometer changed its gain to 1.1, the trailing vehicle accelerated beyond the normal acceleration limits and corrective action is not taken. A third vehicle following the second vehicle would see a

sudden increase in spacing which could be indicated as a spacing sensor bias. Because this situation occurs as a result of vehicle 2 using preceding velocity measurements in the control, it was examined for vehicle 3 using D3 which also uses the measurement. Simulation demonstrated that it caused no false alarm in vehicle 3. Because the third vehicle responds less violently than the second there is no possibility that its response would cause a false alarm in a fourth vehicle. Other failures do not run the risk of causing false alarms in a third vehicle because the second vehicle does not exceed operational limits in response to them. The third vehicle would only see the second accelerating or decelerating within normal operational limits and would track the second.

The results of this section demonstrate that spacing sensor failures can be detected without any other information from the preceding vehicle. Preceding velocity measurements allow spacing sensor failures to be detected more quickly. The preceding speedometer failure, which causes the most severe response, requires approximately five time steps or 0.5 seconds to be detected by the trailing vehicle. Although it is detected before the spacing has decreased significantly, it may be desired to lessen the severity of the response with a quicker detection. A quicker detection can be provided with dual preceding speedometers. If the preceding velocity measurements are being transmitted to the trailing vehicle from the preceding vehicle then dual sensors would not be required. The preceding vehicle will be able to detect its own speedometer failures as quickly as the trailing vehicle can detect the trailing speedometer failure so that correct velocity information can be transmitted to the trailing vehicle at all times.

5.3 Multiple Model and Control

In this section, the effects of using the information provided by the failure detection algorithm to control an AGT vehicle is examined. The primary concern is whether or not the failure detection algorithm will interact with the control law so as to produce an unstable system, or other less drastic but nonetheless undesirable effects. The ability of the algorithm to quickly and correctly identify failures when it is not in the control loop has been established previously.

The two block diagrams in Chapter 2 illustrating how MM could be used in the control loop suggest that a separate control law be designed for each model, but a simpler approach was taken in this section, in which only a single controller was used. The failures examined in previous chapters can affect the system by causing incorrect information to be supplied to the controller. However, the MM failure detection algorithm can deal with these failures by detecting them quickly and then by supplying estimates of spacing and vehicle velocities to the controller rather than the raw, possibly faulty measurements. Because of the MM failure detection performance discussed in the previous chapters, these estimates are relatively unaffected by sensor failures and because they are generated using Kalman filters, will minimize the effects of sensor noise.

As suggested by the block diagrams in Chapter 2, there are two ways the state estimates from each model could be used to determine an estimate for the controller. The estimate could be computed by weighting the state estimates from each model with the model probability and adding, or the estimate from the most probable model could be selected. The behavior

of the probabilities in detecting the failures indicates that these two ways would be equivalent in most cases. Selecting the most likely estimate is equivalent to adding the estimates from all the models where the estimate of the most likely model is weighted with a factor of unity and all other estimates are weighted with a factor of zero. When a failure is detected, we have seen that the weights will change over a few steps so that the correct failure model's weighting factor will change from zero to unity and the unfailed model's weighting factor will change from unity to zero. The probabilities come very close to weighting the state estimates this way. For almost all failures the probability of the failure model jumps from less than .02 to greater than .98 in two steps or less. A correspondingly quick decrease occurs in the probability of the unfailed model. Because there is so little difference, probabilistic weighting was arbitrarily selected for the simulations conducted in this section.

The performance of D3 in detecting the failures made it unnecessary to simulate the vehicle string using the algorithm based on this model in the control loop. In general, the failures were detected so quickly and accurately that they would have no effects on the system if D3 were used in the control loop. The trailing speedometer failure was detected immediately so that the estimates always agreed with the true state of the system in the simulations where the algorithm was not used in the control loop. The trailing odometer bias of +1 m was detected one time step after the failure occurred so that at the time of the failure the bias did cause the estimates to deviate from the true state, but the deviation was minimal. The spacing estimate decreased because the odometer indicated

that the trailing vehicle had moved farther than predicted, and the trailing velocity estimate increased because a greater velocity would be required to account for this greater distance traveled. However, the spacing estimate was 7.64 m as opposed to the true spacing of 7.66 m and the velocity estimate was 15.02 m/sec as opposed to 15.00 m/sec for the true velocity. These deviations are much less than normal random deviations that would be caused by noisy sensors with the assumed noise standard deviations of 0.2 m for the odometer and 0.1 m/sec for the speedometer. Deviations similarly small in magnitude were noted in the one time step while the spacing sensor failure was undetected.

The preceding speedometer failure caused the preceding velocity estimates to deviate significantly from the true preceding velocity for about half a second. During this time the spacing decreased somewhat below the minimum safe spacing, and it would decrease still further before the controller could bring it to the proper spacing. One way to decrease the trailing vehicle's detection time of this failure and thus avoid the decrease in spacing would be to use redundant preceding speedometers. However, the trailing vehicle should not have to detect this failure.

The preceding vehicle will be able to detect its own speedometer failure as quickly as the trailing vehicle can detect its speedometer failure. Consequently, the preceding vehicle can transmit its accurate velocity estimates to the trailing vehicle in the event of a failure. This removes the task of detecting failures in this performance critical sensor from the trailing vehicle and allows this failure to be detected so quickly that it will have negligible effects on the system.

With regard to trailing vehicle failures and estimates of the trailing vehicle's position and velocity, D4 performs as well as D3. However, without preceding velocity measurements it cannot estimate preceding velocity as well and this has a significant effect on its performance in the control loop, even in the absence of any failures. Figure 5.3-1 illustrates how well D4 estimated relative velocity when it was not used in the control loop. Because trailing velocity is accurately estimated, knowledge of relative velocity is equivalent to knowledge of preceding velocity. The curves shown are those generated for the two variations of D4 which could detect the spacing sensor failure, the large bandwidth D4 with preceding position measurements (large Δv driving noise) and the small bandwidth D4 (small Δv driving noise). Without preceding position measurements a small filter bandwidth is needed so that the spacing sensor bias will not be tracked, but the assumption of small driving noise also prevents the true relative velocity from being tracked quickly. The larger driving noise results in a larger bandwidth filter so that the true relative velocity is better tracked but the estimates still respond sluggishly.

Using these sluggish filters to provide estimates to the controller results in the trailing vehicle responding drastically to a deceleration maneuver. Figure 5.3-2 illustrates the variations in relative velocity when the measurements are used to drive the controller and when estimates from the large and small bandwidth filters are used to drive the model. The small bandwidth filter causes the trailing vehicle to respond more severely because it is slower in estimating the true relative velocity. The actual relative spacing must decrease more before the estimate responds

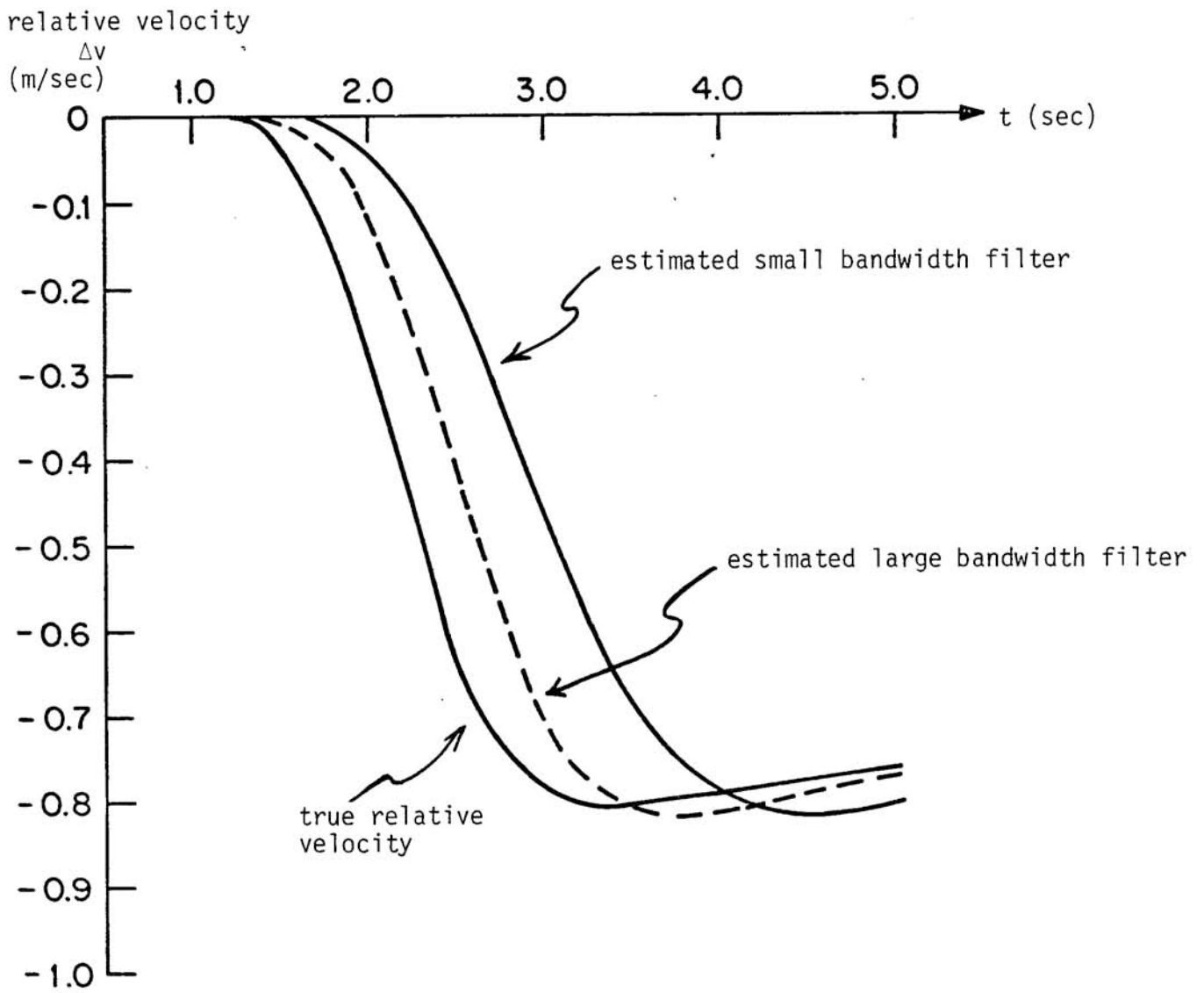


Figure 5.3-1 Relative Velocity Estimates

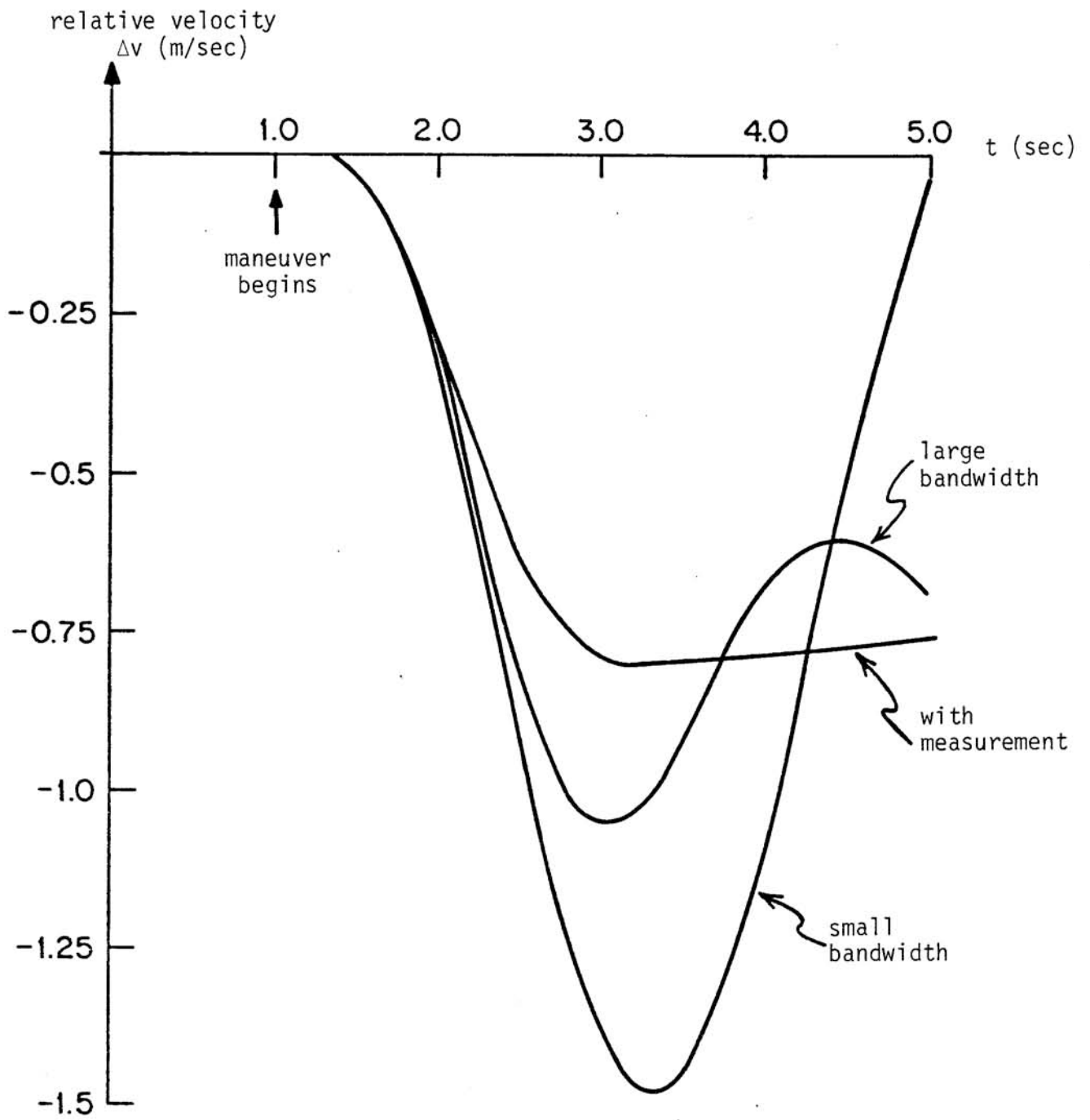


Figure 5.3-2 Response to Deceleration Maneuver Using Estimates

substantially for the small bandwidth filter than it must for the large bandwidth filter. This causes the system to overreact later to compensate for its initial delay, and leads to oscillations. The larger bandwidth filter responds more quickly to the decreasing spacing and so does not overcompensate later.

This behavior would lead to an unstable string. The deceleration of the first vehicle causes a more severe deceleration in the second vehicle which would cause an even more severe deceleration in a third vehicle and so on down the string. Therefore this is an unacceptable controller.

The results of this section demonstrate that the trailing vehicle requires accurate velocity information from the preceding vehicle for good controller performance. Because the preceding vehicle can detect its own speedometer failures quickly and accurately it can provide accurate velocity information to the trailing vehicle at all times. When these conditions are met the MM-based fault tolerant controller performs very well. If the trailing vehicle is required to detect failures in the preceding speedometer spacing decreases somewhat below the minimum safe spacing before the failure is detected. Without preceding velocity measurements the trailing vehicle was unable to estimate preceding velocity well enough for acceptable control.

6. Summary and Recommendations

In this thesis the MM algorithm was examined to determine how it could be used to detect failures and consequently improve safety and reliability for AGT systems. A fault-tolerant control strategy based on the MM algorithm was also examined to demonstrate how MM could be used to make AGT vehicles fail operational. The basic conclusion that can be drawn from this research is that failures can be quickly detected and accurately identified using the MM algorithm and that it can allow AGT vehicles to continue operating safely after a failure has occurred.

6.1 Summary of Research Results

The research was conducted by first examining the performance of the algorithm in detecting failures and the factors influencing that performance. Specifically, the issues examined were:

1. the ability of various dynamical models to adequately describe the vehicle and the various failures
2. the selection of noise statistics and their effects on detection performance
3. the ability of the algorithm to detect failures not explicitly modeled or improperly parameterized
4. the improvement in detection performance through the use of redundant sensor configurations.

The central feature of the algorithm is the set of models used to describe the system in various failed and unfailed conditions. Several different sets of models and their effects on failure detection performance were examined. It was shown that basic kinematic relationships driven by commanded acceleration modeled the vehicle well enough to provide good

failure detection capability. Speedometer and odometer failures were detected very quickly. Wind and grade did not cause any false alarms, and vehicle dynamics were modeled well enough so that maneuvers did not cause false alarms. More detailed models that incorporated more of the dynamics of the AGT vehicle than basic kinematics or that avoided the difference between commanded and actual acceleration were also examined but they had various drawbacks. One required an accelerometer to drive the kinematic equations, another modeled only the relationship between velocity and position measurements but required longer to detect speedometer failures, a third used a more complex model which required more computation and relied on uncertain vehicle parameters. Because the use of the simple approximation of vehicle dynamics did not result in false alarms and allowed failures to be quickly detected, these other models are not recommended.

The modeled sensor and plant noise covariances were shown to be important model parameters with significant effects on failure detection performance. These covariances should be chosen as small as possible to provide for the quickest detection of failures, but they must be large enough to account for the sensor noise actually present and modeling inaccuracies, or false alarms will result.

The robustness of the algorithm was examined by determining how it performed in detecting failures that are not modeled or improperly parameterized. It was shown that a failure that does not correspond exactly to any of the failure models can cause a failure model to be indicated. One specific demonstration showed that a speedometer gain change to 0.925

caused the 0.9 speedometer gain model to be selected. This capability allows MM to indicate that a failure has occurred, but more accurate identification of the failure magnitude may be required if the system is to be fail operational. A more accurate identification can be achieved through using more models in the MM algorithm, or by using a parameter identification algorithm once MM indicates that a failure has occurred. Whether a fine set of MM models (e.g., gain change to .9, to .925, to .95) is worthwhile or not needs to be examined. It should be noted that a two-mode procedure--i.e., detect using one model and then estimate the gain change--will actually involve a variable set of models in the second stage in order to determine the gain value. Therefore, it is not clear a priori which of the two methods will yield acceptable performance with a computationally less complex algorithm.

Failure detection performance was shown to improve through the use of redundant sensors. Redundancy yields the greatest improvement when modeling inaccuracies limit detection performance. This was demonstrated by showing how the time required to detect failures decreased with redundant sensors, but they can also prevent false alarms by verifying one another when the model incorrectly predicts what the measurement should be.

After the algorithm's ability to detect failures was examined, the use of MM in a fault tolerant control strategy was examined. The particular vehicle follower control law examined required a vehicle to know the immediately preceding vehicle's velocity and the spacing between the two vehicles. Under reasonable assumptions concerning how the trailing vehicle would obtain this information, excellent control results were

obtained. Specifically, it was assumed that the preceding vehicle would relay its own velocity measurements to the trailing vehicle, and that the preceding vehicle would detect its own speedometer failures, allowing it to supply accurate estimates of its own velocity to the trailing vehicle in the event of a failure. These assumptions allowed the failures to be quickly and accurately detected so that the effects of the failures on the system were insignificant. When the trailing vehicle was required to detect the preceding speedometer failure the spacing decreased somewhat below the minimum safe spacing before the failure was detected. One possibility for relieving the trailing vehicle's dependence on receiving accurate velocity information from the preceding vehicle was examined. This possibility involved using MM to estimate the preceding velocity from spacing measurements and its own velocity measurements. The preceding velocity estimates differed significantly from the true velocity even under no fail conditions so that when used by the controller they resulted in unacceptable string behavior.

6.2 Suggestions for Further Research

In this thesis the MM algorithm was tested using data from a simulated AGT vehicle. The algorithm needs to be tested using data from an actual AGT vehicle.

The failure magnitudes were significantly greater than the noise levels in this research, allowing failures to be quickly detected and accurately identified. Failures smaller in magnitude will be more difficult to detect and identify with these noise levels. Further research should be conducted to determine the minimum failure magnitudes that must

be detected in order to avoid safety problems and the minimum failure magnitudes that can be detected with given noise levels.

It has been demonstrated previously that a failure can cause a failure model to be indicated even though it does not correspond exactly to the model. Although the model describes the failure accurately enough for the purposes of failure detection, it may not be accurate enough for the purposes of fault tolerant control. For example, the speedometer gain change to 0.925 caused the 0.9 speedometer gain model to be selected, but the drastic change in steady state spacing caused by variations in speedometer gain indicates that a more accurate identification of the gain may be necessary. Methods for more accurately identifying failures need to be examined. One method would be to use parameter identification algorithms once MM has indicated a failure has occurred. Another method would be to use a larger set of models for MM.

Additional methods for avoiding or alleviating the trailing vehicle's dependence on the preceding vehicle for accurate information concerning the preceding vehicle's velocity need to be examined. It is possible that through modifying the control law that the preceding velocity estimates generated by the algorithm would be acceptable substitutes for preceding velocity measurements. Specifically, the velocity estimates resulted in an unacceptable control because of the delay involved in estimating the true velocity as discussed in Section 5.3. A different selection of constants used in the control law could be made to account for this delay. Also, there are other control laws that do not require knowledge of preceding vehicle velocity as mentioned in [25]. For these control laws,

spacing will be the critical quantity. The spacing estimates provided by the algorithm may be sufficiently accurate in both failed and unfailed operation so as to provide good fault tolerant control when used by these control laws.

References

1. Garrard, W. L., R. J. Caudill and W. B. Reed, "Control Considerations for Automated Guideway Transit Systems," Proc. IEEE Conf. on Decision and Control, pp. 567-527, Dec. 1975.
2. Athans, M., W. S. Levine and A. H. Levis, "On the Optimal and Suboptimal Position and Velocity Control of a String of High Speed Moving Trains," MIT Electronic Systems Laboratory, Report PB 173640, Nov. 1966.
3. Brown, S. J. Jr., "Characteristics of a Linear Regulator Control Law for Vehicles in an Automatic Transit System," AIAA Guidance, Control and Flight Mechanics Conf., Hempstead, N.Y., Aug. 1971.
4. Brown, S. J. Jr., "Design of Car Follower Type Control Systems with Finite Bandwidth Plants," Proc. 7th Annual Princeton Conf. on Information Sciences and Systems, pp. 57-62, March 1973.
5. Chiu, H. Y., G. B. Stupp Jr. and S. J. Brown Jr., "Vehicle Follower Control with Variable Gains for Short Headway Automated Transit Systems," J. Dynamic Systems, Measures and Control, pp. 183-189, Sept. 1977.
6. Chu, K. C., "Decentralized Control of High Speed Vehicular Strings," Transportation Science, Vol. 8, pp. 361-384, 1974.
7. Cunningham, E. P. and E. J. Hinman, "An Approach to Velocity/Spacing Regulation and the Merging Problem in Automated Transportation," Joint Transportation Engineering Conf., ASME paper 70-Tran-19, Oct. 1970.
8. Garrard, W. L., R. G. Hand and R. Raemer, "Suboptimal Feedback Control of a String of Vehicles Moving on a Single Guideway," Transportation Research, Vol. 6, pp. 197-210, 1972.
9. Garrard, W. L. and R. J. Caudill, "Dynamic Behavior of Strings of Automated Transit Vehicles," Soc. of Automotive Engineers, International Automotive Engineering Congress and Exposition, Detroit, Mich., March 1977.
10. Levine, W. S. and M. Athans, "On the Optimal Error Regulation of a String of Moving Vehicles," IEEE Trans. Automatic Control, Vol. AC-11, pp. 355-361, 1966.
11. Pue, A. J., "A State-Constrained Approach to Vehicle-Follower Control for Short Headway AGT Systems," U.S. Dept. of Transportation, Report UMTA-MD-06-0022-77-2, Aug. 1977.

12. Willsky, A. S., "A Survey of Design Methods for Failure Detection in Dynamic Systems," Automatica, Vol. 12, pp. 601-611, 1976.
13. VanderVelde, W. E., "Application of Failure Detection Theory to Reliable Longitudinal Control of Guideway Vehicles," U.S. Dept. of Transportation Report, Contract DOT-TSC-1445, March 1979.
14. Beard, R. V., "Failure Accommodation in Linear Systems Through Self-Reorganization," PhD thesis, Dept. of Aeronautics and Astronautics, MIT, Feb. 1971.
15. Jones, H. L., "Failure Detection in Linear Systems," PhD thesis, Dept. of Aeronautics and Astronautics, MIT, Aug. 1973.
16. Greene, C. S., P. K. Houpt, A. S. Willsky and S. B. Gershwin, "Dynamic Detection and Identification of Incidents on Free-ways, Volume III: The Multiple Model Method," MIT Electronic Systems Laboratory, Report ESL-R-766, 1977.
17. Gustafson, D. E., A. S. Willsky, Y. Y. Wang, M. C. Lancaster and J. H. Triebwasser, "A Statistical Approach to Rhythm Diagnosis of Cardiograms," Proc. IEEE, Vol. 65, pp. 802-805, 1977.
18. Athans, M., Y. Baram, D. Castanon, K. P. Dunn, C. S. Greene, W. H. Lee, N. R. Sandell Jr. and A. S. Willsky, "Investigation of the Multiple Model Adaptive Control Method (MMAC) for Flight Control Systems," NASA Contractor Report 3089, 1979.
19. Greene, C. S., "Application of the Multiple Model Adaptive Control Method to the Control of the Lateral Dynamics of an Aircraft," S.M. thesis, Dept. of Electrical Engineering, MIT, June 1975.
20. Greene, C. S., "An Analysis of the Multiple Model Adaptive Control Algorithm," PhD thesis, Dept. of Electrical Engineering, MIT, Aug. 1978.
21. Pitts, G. L., "Control Allocation Investigation: Sampling Rate Selection," U.S. Dept. of Transportation, Report UMTA-MD-06-0018-74-2, April 1974.
22. Pue, A. J., "Implementation Trade-off Study for a Short Headway AGT System," APL/JHU, FIC(2)-77-U-013, May 25, 1977.
23. Petrino, E. et al., "Volume 3 - VLCR Longitudinal Control Analysis and Design, Part A - SLT and GRT Systems," U.S. Dept. of Transportation, Report UMTA-IT-06-0148-79-7, May 1979.
24. Chiu, H. Y., G. B. Stupp Jr. and S. J. Brown Jr., "Vehicle Follower Controls for Short Headway AGT Systems - Functional Analysis and Conceptual Designs," U.S. Dept. of Transportation, Report UMTA-MD-06-0018-76-4, Dec. 1976.

25. Schumacher, P. (ed.), "Volume 3 - Longitudinal Control Analysis and Design (Part B - PRT Systems)," U.S. Dept. of Transportation, Report UMTA-IT-06-0148-79-8, May 1979.