NASA/CR 97- 206027

693187

Final Report

Integrator Windup Protection – Techniques and a STOVL Aircraft Engine Controller Application

K. KrishnaKumar, Associate Professor
S. Narayanaswamy, Graduate Research Assistant

Department of Aerospace Engineering
The University of Alabama, Tuscaloosa, AL 35487

for

NASA Lewis Research Center, Cleveland, OH
Grant # NAG3-1564

June, 1997

Acknowledgments

We would like to thank Dr. Sanjay Garg of the Advanced Controls Technology Branch at NASA Lewis Research Center for his suggestions and support of this project.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Integrators are included in the feedback loop of a control system to eliminate the steady state errors in the commanded variables. The integrator windup problem arises if the control actuators encounter operational limits before the steady state errors are driven to zero by the integrator. The typical effects of windup are large system oscillations, high steady state error, and a delayed system response following the windup. In this study, methods to prevent the integrator windup are examined to provide Integrator Windup Protection (IWP) for an engine controller of a Short Take-Off and Vertical Landing (STOVL) aircraft. An unified performance index is defined to optimize the performance of the Conventional Anti-Windup (CAW) and the Modified Anti-Windup (MAW) methods. A modified Genetic Algorithm search procedure with stochastic parameter encoding is implemented to obtain the optimal parameters of the CAW scheme. The advantages and drawbacks of the CAW and MAW techniques are discussed and recommendations are made for the choice of the IWP scheme, given some characteristics of the system.

Chapter 1.

Introduction

Flight control systems for modern tactical/fighter aircraft are to be designed for enhanced flight maneuvers such as Short Take-Off and Vertical Landing (STOVL) and high angle of attack flights. The forces and moments generated by the conventional control surfaces are usually insufficient for successful completion of such maneuvers under the flight conditions considered, and hence are augmented with the forces from the propulsion system. The engine and its subsystems also play an important role for aircraft with tilt-rotors and for aircraft with engine steering control. Thus the propulsion system of a modern aircraft is not being used to just overcome the atmospheric drag, but also to control the airplane. Hence a need arises to control the engine variables as precisely as possible.

1.1 Integrator Windup Problem

One of the difficulties in controlling the engine variables arises due to the integrator windup, caused due to the control actuators encountering the operational limits when integrators are used in the feedback loop of the control system. Operational limits include both physical limits on the actuator motion as well as the user imposed limits on certain variables of the system to avoid extreme operational conditions. For instance, limiting the maximum fuel flow to the engine to avoid high temperature damage of turbine blades is an operational limit on the fuel flow. Integrators are essential in the feedback loop to drive the steady state errors in the commanded variables to zero. If a control actuator encounters the operational limit before the steady state errors reach zero, the errors cannot be driven to zero by the controller. The integrator would continue to integrate this non-zero error and hence the integrator's output would build-up to a very high value. This phenomenon is referred to as integrator windup, and results in increased system oscillations, degraded system performance, and in some cases even an unstable system [1]. The windup effects also arise if the dynamics of the controller is relatively slower than that of the error when the operational limits of the system are encountered. Following this period of windup, the controller's response to new command inputs might be poor because the integrators must first unwind prior to attempting to drive the steady state errors, due to new command inputs, to zero [2].

In the early phases of feedback control, practicing engineers had developed a good intuition on the effects of integrator windup and methods for handling it. Often, the integral action was achieved due to the motion of the control actuator itself [3]. In such cases, integration stops when the actuator encounters the physical limit, and hence the windup is avoided. These methods had little theoretic foundation but worked well for Single Input-Single Output (SISO) systems that are simple to analyze. However, modern day systems are mostly multivariable, and such Multiple Input-Multiple Output (MIMO) systems are quite cumbersome to analyze due to the complexity of the system. In a MIMO system, the windup problem is further aggravated if there exists a strong coupling between the control variables of the system, which degrades the system performance even if a single control actuator encounters the operational limits. Also, if the MIMO system is sensitive to direction changes of control vector, then the performance of the system is further deteriorated [4]. While much progress has been made in the development of multivariable control theory, some important issues like the integrator windup problems have not been addressed yet in a systematic manner.

It might seem to be a rational approach to handle the windup problem at the design stage of the controller by including the plant limits [4]. However, this approach is quite tedious and the resulting control law is often very complicated. Also, the nonlinearities of the actuator are not always known apriori. Hence, the design of the controller is often based on linear theory by neglecting the limits. An extra feedback compensation is then added at the control implementation stage to take the practical limits into consideration. As this compensation aims to diminish the effects of windup, it is referred to as anti-windup or Integrator Windup Protection [5]. This additional compensation leaves the original linear behavior unchanged but provides graceful degradation of system performance when actuator limits are encountered.

1.2 Approaches to providing IWP

Most of the IWP schemes modify the error between the commanded input and the plant output to achieve the anti-windup action. Typically, this error is reduced by adding a non-zero factor that is proportional to the extent of the windup. This decreases the magnitude of the error that would be integrated by the integrator and hence diminishes the effects of windup. This principle is also called back-calculation since the controller states are back-calculated such that the output of the controller is at the actuator limit [3].

In the Conventional Anti-Windup (CAW) scheme, the difference between the actuator limit and controller output is multiplied by a set of gains (referred to as IWP gains) and is added to the above error term when the actuators are limited. The CAW method works well for a SISO system, but for a MIMO system, anti-windup provided by the CAW scheme may not be adequate [6]. Also, the CAW scheme does not maintain the direction of the controller's output vector, and hence is not suitable for systems that are sensitive to changes in control vector direction.

In the Modified Anti-Windup (MAW) approach, controller states are multiplied by a windup factor before being added to the above described error term. The windup factor is such that the direction of the controller output vector is maintained [1] and hence this scheme is suitable for plants that are sensitive to direction changes of the control vector. The main drawback of the MAW scheme is that it has few design variables and hence we do not have sufficient degrees of freedom to design the IWP for time varying actuator limits. Consequently, the performance of the MAW scheme suffers when the system encounters a limit different from the one for which it was designed.

1.3 Optimization Methods

The choice of the optimization method used to determine the set of optimal parameters is an important step in the overall solution process. Genetic Algorithms (GA) have been successfully used in the past to solve a number of non-linear optimization problems [7]. As the GAs do not require any gradient information to aid their search, they can be used even for cases where the derivatives are difficult to obtain. However, GAs require information about the range in which parameters of the optimization problem are expected to lie. This handicaps a regular GA approach in cases where we do not know the expected range of parameters apriori. Also, for problems with large number of optimization parameters, the regular GA needs a large population size to find the optimal solution, and this calls for enormous computational resources [8]. In this study, an improved version of the stochastic GA proposed in Reference [9] is implemented to solve the IWP optimization problem. This modified GA with a stochastic encoding structure overcomes the shortcomings of the regular GA in that it does not need the exact parameter ranges, and it converges to the global optimum quickly with a small population. This search technique starts with a given search region and as the GA population evolves, the search region is modified to capture the global optimum and finally to converge to it. The accuracy of the GA solution is verified by carrying out the optimization using the optimization module supplied with MatrixX [10].

1.4 Study Objectives

In this study, we develop an unified performance index to provide anti-windup compensation using the CAW scheme. This performance index is optimized to provide IWP for the engine control system of a STOVL aircraft. The IWP is implemented on the linear model of the engine control system developed at NASA Lewis Research Center. The objectives of this study include :

- Studying the CAW and MAW schemes for providing IWP and implementing them to example problems to show their merits and demerits.
- Development of an unified performance index for the design of IWP.
- Development of an improved stochastic Genetic Algorithm (GA) method to obtain the optimal IWP gains.
- Application of the CAW and MAW schemes to provide IWP for the engine controller of a STOVL aircraft.
- Analysis of the results obtained using the CAW and MAW schemes.
- Conclusions regarding the suitability of CAW or MAW for a given control system.

1.5 Thesis Organization

This thesis is organized as follows: In chapter 2, the previous literature on the IWP methods is detailed followed by the optimal parameterized IWP formulation for a generic control system. In chapter 3, we examine the optimization techniques used to obtain the optimal IWP gains. In chapter 4, we implement the IWP schemes for the engine control system of a STOVL aircraft, and discuss the results obtained. The conclusions are presented in chapter 5, followed by the recommendations for the choice of the IWP scheme, given some system characteristics.

Chapter 2

Optimal Parameterized IWP Formulation

Most natural and man made systems are inherently non-linear, though to a varying degree. It is well known from the non-linear control theory that analysis and design of non-linear control systems is quite complex and mathematically unwieldy. Hence, for the ease of analysis we linearize the non-linear system about a particular operating point and the resulting linear system provides an acceptable solution as long as we stay close to the operating point. However, in most practical cases, large disturbances acting on the system and changes in operating points cannot be avoided, thereby forcing us to face the non-linearities temporarily. Examples of such non-linearities include actuator limits, parameter variations due to changes in operating point of the system, backlash in gears or valves [11], hysteresis [12] etc.

The effect of feedback control is to reduce the system sensitivity to such external disturbances, to minimize the effect of plant parameter variations, and to modify the system dynamics to a desired form. If in addition, the steady state errors on the commanded variables are to be driven to zero, it can be shown that the feedback loop should have an integral control law. However, when integrators are used in the feedback loop, limits on the actuator can cause the integrators to windup thereby deteriorating the performance of the system. Further, the response of the system following the period of windup could be sluggish until the integrator unwinds. Integrator Windup Protection (IWP) is hence included in the feedback loop to minimize the effects due to windup. In this chapter, first we briefly review the available literature on IWP methods. The IWP methods we consider for further study, the CAW and MAW schemes, are then described in detail followed by the description of the performance index that conforms with the requirements of IWP. We then apply these IWP schemes to two example problems to illustrate the effect of directional sensitivity of the system on the choice of the IWP scheme.

2.1 Literature Overview

Early efforts to handle integrator windup were mainly focused on SISO systems. According to the Back-Calculation principle given in Reference [3], when the controller output exceeds the actuator limits, the integral is re-computed such that its new value gives an output at the limit.

Reference [1] examines two methods based on back-calculation principle to provide windup protection for a turbofan engine control system. The first method is an extended version of the Conventional Anti-Windup (CAW) scheme for multivariable controllers. In the Conventional Anti-Windup method, the actuator error is fed back through the IWP gains as an additional contribution to the state derivative calculations. The MAW scheme uses a scalar windup factor to modify the magnitude of the control vector while maintaining its direction. The calculation of this windup factor is a non-linear function of the controller outputs and the actuator limits. The CAW and MAW schemes are described in detail in a later section.

Reference [6] points out that the directional sensitivity of a MIMO system plays a key role in the design of IWP. Examples are given in reference [6] to show the that the CAW scheme does not provide satisfactory protection against windup for directionally sensitive systems. The structured singular value (μ) analysis is used to determine the directional sensitivity of the CAW system. We include these examples at the end of this chapter to illustrate the effect of directional sensitivity of the system on the performance of the CAW system.

Reference [4] includes the issue of directional sensitivity in the IWP design requirements. The structured singular value (μ) is used in [4] to determine the sensitivity of the system to direction changes of the control vector. We adopt this measure later in our performance index definition (section 2.6) to minimize the directional sensitivity of the CAW system. It is pointed out in reference [4] that, if other IWP schemes do not satisfy the μ requirements, the MAW scheme could be used to provide anti-windup compensation.

Reference [13] implements an intelligent limiter based on back-calculation principle for a second order plant and carries out the stability analysis of the limited system using the describing function method and Nyquist stability theorem. Reference [11] implements a modified back-calculation principle for windup protection for cascade controllers, where a limit at secondary actuator may cause windup in the primary controller.

In Incremental Algorithms approach [3], the rate of change of the control signal is first computed and then fed to an integrator. Integration is stopped whenever the output exceeds the actuator's operational limits. Reference [5] implements the discrete-time implementation of Incremental Algorithm for a PID controller.

Conditional Integration technique is implemented in [5] to avoid windup of a PID controller. According to this technique, when the controller works in the linear region, i.e. when the controller output is not limited, the error term is integrated by the integrator and when the actuator is limited, the integration is stopped. Reference [3] introduces the notion of a proportional band and integrates the error only if the predicted process output is in the proportional band. The proportional band is the interval in which the process output lies when the control variable is varied within the control limits. Reference [14] implements an anti-windup method based on conditional integration to provide windup protection for a digital multivariable controller. It, however focuses on a particular system with two inputs and two outputs and the implementation is difficult to generalize for systems with more inputs or outputs.

In the conditioning technique approach proposed by [12], a realizable reference signal is input to the system instead of the actual reference input. The realizable reference signal is computed such that if it is applied to the controller, the controller output would have been at the actuator limit. Reference [4] shows that this technique might provide poor anti-windup performance for some cases.

In the observer based approach, the controller states have the physical interpretation as the estimates of plant states [4]. Hence the objective here is to design such that the controller states assume the correct estimates of the plant states regardless of plant limitations. If the controller has full access to the plant states, this can be achieved by a simple state feedback. In the case where the full plant states are not accessible, an observer is constructed to supply plant states which are used to provide the feedback.

Internal model control (IMC) based anti-windup technique requires an exact plant model for closed loop stability when the actuator encounters operational limits. Reference [4] shows that there are no inherent properties of IMC that provide robustness to diagonal input uncertainties.

Most of the anti-windup methods described above are applicable only for SISO systems. Some of the anti-windup examples have been applied to MIMO system examples, but they are highly dependent on the problem being solved. They also lead to a complex setup when the number of inputs/outputs increase. Hence a need is felt to derive a simple anti-windup solution that is applicable to a generic MIMO system regardless of the number of system inputs/outputs. We begin by stating the general requirements of an IWP scheme.

2.2 IWP Design Requirements

Integrator Windup Protection (IWP), when included in the feedback loop should prevent the integrator's output from building up when the actuator encounters a limit. Some of the important design requirements for an IWP scheme are[1]:

1. A limited actuator must be observable.
2. IWP should be memoryless and should not contribute to the control system when the limits are not encountered.
3. IWP should provide closed loop stability for all possible actuator limit combinations within the system's operating envelope.
4. IWP should attempt to maintain system performance for all possible actuator limit combinations within the system's operating envelope. If the system performance

cannot be maintained, IWP scheme should provide a smooth, stable transition to a minimally degraded operating point.

5. IWP should provide smooth transfers between the unlimited and limited actuators, while providing accurate tracking of the limited actuators.
6. IWP should minimize the sensitivity of system to direction changes of the control vector due to actuator limits.

We now implement an IWP scheme for a generic control system, which, together with the structure of IWP and the definition of performance index would fulfill these requirements.

2.3 A General IWP Structure

A generic structure of a control system with actuator limits and a IWP structure is shown in Figure 2.1. In this figure, G(s) is the plant, the '*Limits*' block represents the limits on the actuator, K(s) is the controller and IWP is the anti-windup compensation added to the controller as shown. The reference command to the control system is 'r', the controller output is 'u_c', the plant inputs are limited by the limits 'u_L' and the plant outputs are 'z'. The error 'e' between the reference inputs and the plant outputs is fed to the controller K(s). Also, the information about the plant inputs, '$u^L{}_c$' , is supplied to the controller in order to implement the IWP. It is necessary that the IWP scheme included in the controller structure stabilizes the controller and the closed loop system. If there is no error between the plant inputs and the controller outputs (i.e. when the limits not encountered), the IWP gains should be ineffective. When the plant inputs are different from the controller outputs (as in the case of actuator limitation), IWP should reduce the magnitude of error between the plant outputs and the reference inputs so that the windup effects are diminished. We assume that for all systems considered in this study, the actuator is observable. This satisfies the requirement # 1 of the IWP design requirements stated in section 2.2.

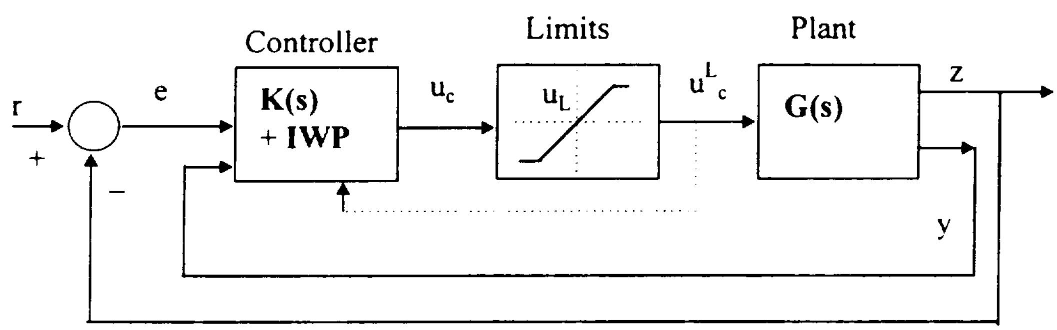

Figure 2.1 General IWP Structure

We now describe two IWP methods that satisfy the requirements stated in section 2.2, and are applicable to a generic MIMO system. These IWP methods, namely the Conventional Anti-Windup method and the Modified Windup method, are based on the back-calculation principle.

2.4 Conventional Anti-Windup (CAW) Scheme

In the CAW method, the controller states are back-calculated by using actuator error feedback such that output of the controller is within the actuator limits [1]. The actuator error is the quantity by which the output of the controller exceeds the actuator limits. An implementation of the CAW scheme is shown in the Figure 2.2. As seen from the figure, if the controller output is within the actuator limits, then the actuator error 'e_u' is zero and the CAW gains are ineffective. Thus the CAW scheme is memoryless and hence satisfies the second IWP design requirement. When the controller output exceeds the limits, the actuator error 'e_u' is fed back through the gain matrix 'Λ', so that the controller output remains within the limits. The terms in the gain matrix 'Λ', referred to as IWP gains, dictate the stability and performance of this system protected against windup. These are obtained such that they satisfy the stability requirements and optimize the system performance.

The equations for the CAW implementation can be given as[1] :

Original Dynamic Controller :

$$\dot{x}_C = A_c x_c + B_c \begin{bmatrix} e \\ y \end{bmatrix};$$

$$u_c = C_c x_c + D_c \begin{bmatrix} e \\ y \end{bmatrix}$$

Controller with IWP :

$$\dot{x}_c = A_c x_c + B_c \begin{bmatrix} e \\ y \end{bmatrix} + \Lambda L e_u$$

$$= \left(A_c - \Lambda L C_c\right) x_c + \left(B_c - \Lambda L D_c\right) \begin{bmatrix} e \\ y \end{bmatrix} + \Lambda L u_c^L$$

In the above equations, [A_c, B_c, C_c, D_c] are the system matrices for the controller, x_c is the state vector of the controller, $[e\ y]^T$ is the controller input, u_c is the output vector of the controller, u_c^L is the vector of actuator limits, e_u is the actuator error vector, Λ is the constant IWP gain matrix and L is a diagonal matrix that represents the actuator being

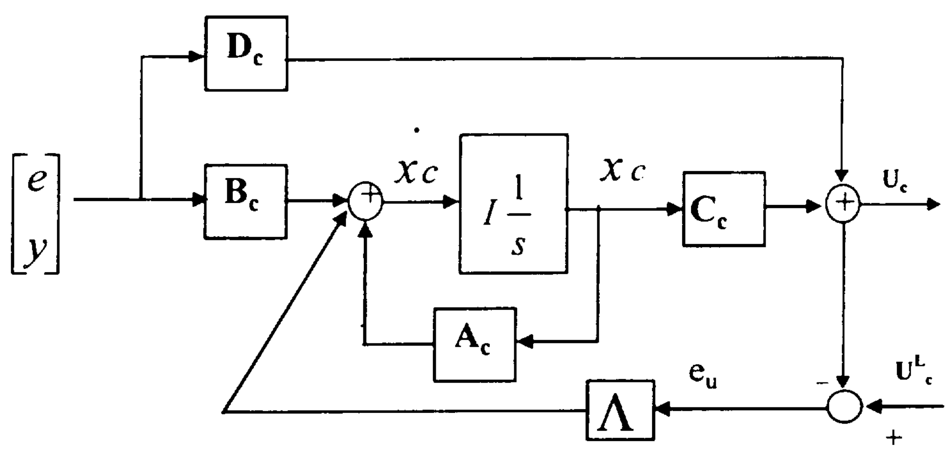

Figure 2.2 CAW Scheme Implementation

limited. If L=0, the nominal control system is obtained. For a two actuator system, L=diag(1,0) represents first actuator being limited, L=diag(0,1) denotes that the second actuator is limited and L=diag(1,1) denotes that both actuators have encountered the limits. As seen in the above equations, the stability of the CAW controller can be obtained from the matrix (A_c-ΛLC_c) for various combinations of L and the chosen IWP gains Λ.

When a control variable encounters the actuator limit, the CAW scheme truncates the variable such that the control is within the limit. This is illustrated in Figure 2.3 [1], in which a two actuator control system is considered. In this figure, u_{c1} and u_{c2} are the two control variables, u_{L1} and u_{L2} are the actuator limits for these control variables, and d_1 is the direction of the unlimited control vector (u_c). As seen from the figure, the control vector modified by CAW scheme u_{CAW} has a direction d_2 different from the unlimited control vector. Some systems are sensitive to changes in the direction of control vector and for these systems, the CAW scheme could lead to a poor closed-loop performance. Hence, for such systems, we must ensure that the CAW gains minimize the sensitivity to the control vector direction changes. A distinct advantage of this scheme is that the number of design variables (terms in the gain matrix Λ) is equal to the product of number of actuators and the number of states of the controller. Hence we have sufficient degrees of freedom to design the IWP for an acceptable performance level.

2.5 Modified Anti-Windup (MAW) Scheme

In this anti-windup method, modified control vector is obtained by scaling down the unlimited control vector such that the control variables are within the actuator limits.

When the control vector is scaled down by a scalar factor, the direction of the control vector is maintained. In Figure 2.3, u_{maw} is the control vector modified by the MAW scheme and as seen from figure, it has the same direction as the unlimited control vector. Since this scheme maintains the direction of the control vector, it is suitable for systems sensitive to direction changes of control vector. Figure 2.4 shows the implementation of the MAW scheme. In this figure, α is the scalar windup factor that denotes the extent of

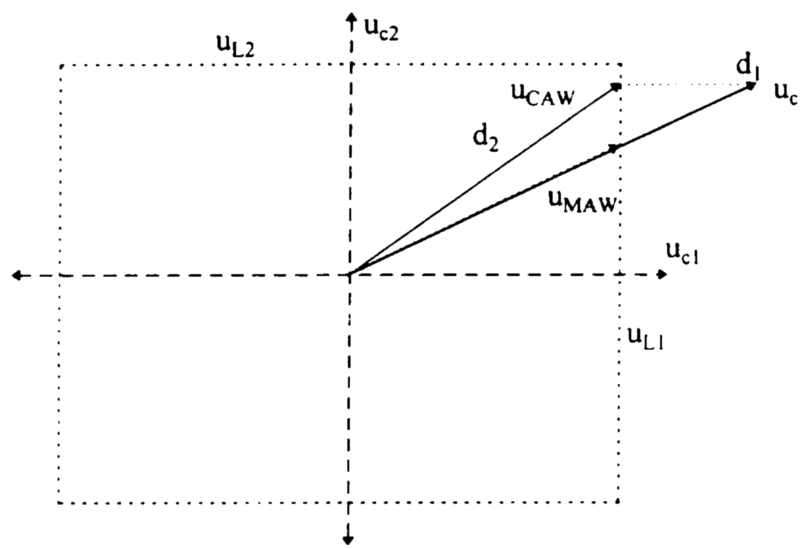

Figure 2.3 Control Vector Directions for CAW and MAW schemes

windup and β is a design variable. The windup factor α is defined as

$\alpha(t) = 1$ if u_c is within limits, otherwise

$$\alpha(t) = \min_i \left| \frac{u_c^L(t)_i}{u_c(t)_i} \right|, \quad u_c(t) \neq 0$$

where $u^L{}_c$ is the limited control vector (plant inputs), u_c is the unlimited control vector (controller outputs), and subscript i denoting the i^{th} component of the vector. The design variable β is to be determined such that the closed loop system is stable and the performance of the IWP system is close to that of the nominal system. As seen from Figure 2.4, the nominal control system is active as long as $\alpha = 1$. When a control actuator encounters the limit, $\alpha \neq 1$, and the nominal controller is modified by the two additional blocks shown in the figure. The state vector x_c is fed back after multiplication by the factor $\beta(\alpha-1)$ so that the output of the controller is eventually brought within the actuator limits.

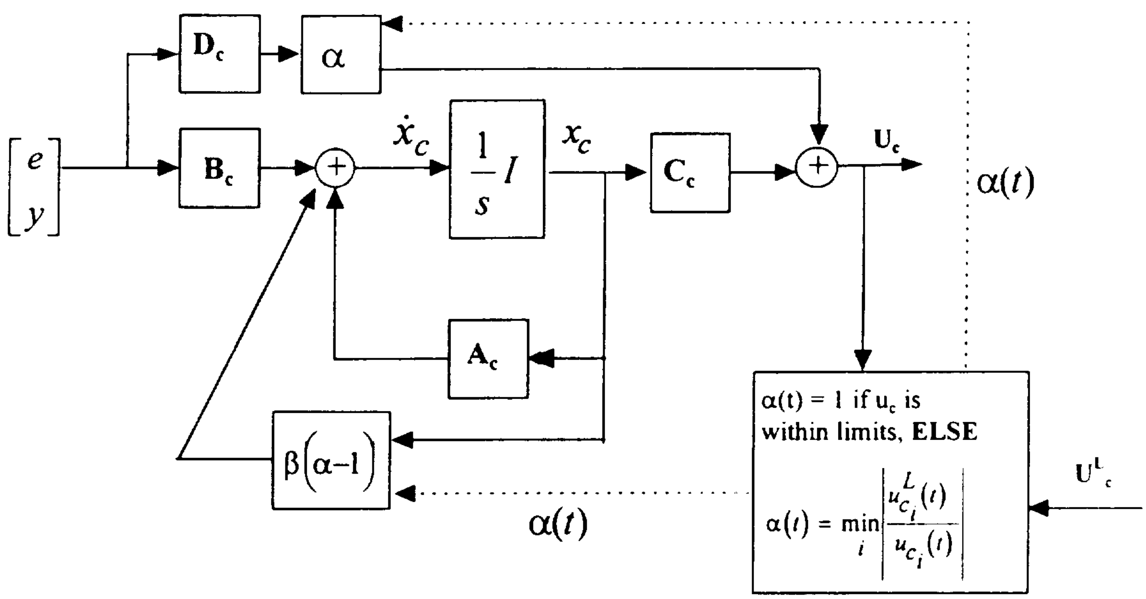

Figure 2.4 MAW Scheme Implementation

The equations that implement the MAW scheme can be given as [1]:

$$\dot{x}_C = \left(A_C + \beta(\alpha - 1)I\right)x_C + B_C\begin{bmatrix} e \\ y \end{bmatrix}$$

$$u_C = C_C x_C + \alpha D_C\begin{bmatrix} e \\ y \end{bmatrix}$$

The system matrix $(A + \beta(\alpha - 1)I)$ in the above equation can be rewritten using similarity transformation as

$$T^{-1}(A + \beta(\alpha - 1)I)T = T^{-1}AT + \beta(\alpha - 1)I$$

Since $0 < \alpha \leq 1$ and $\beta > 0$, it follows that MAW scheme always shifts the real part of the controller eigen values to the left, thereby making the controller more stable.

As described earlier, MAW scheme works well for systems sensitive to control vector direction changes. This is illustrated by an example provided at the end of this chapter. However, since the design parameter of this approach is a single scalar variable (β), we do not have a wide degree of freedom to tune the system for an optimal performance. Also, the optimal performance might require a high value for the gain β that is not be feasible to implement in practice. The non-linearity associated with the MAW scheme while computing the scalar windup factor might degrade the system performance though the effects of windup are eliminated.

2.6 .Performance Measures for IWP

In this section we discuss the performance index chosen to optimize the performance of an IWP scheme. The design parameters of the IWP scheme are to be determined before we could use it to provide windup protection. The parameters have to be chosen such that the stability of the closed loop IWP system is maintained and the IWP system performance is close to the nominal system performance. The performance requirements can be achieved by defining a suitable performance index for the IWP system such that when this performance index is optimized, the performance of the IWP system is close to that of the nominal system. In Reference [2], the root mean square (rms) error between the nominal and IWP system due to a zero mean, unit variance white noise input signal, is considered to be the performance measure for the IWP scheme. The IWP gains are parameters of the optimization problem, and hence can be determined using an appropriate optimization technique. Stability requirements are imposed as constraints while optimizing the performance of the IWP system. If a set of parameters result in an unstable system, a poor performance index is assigned for that set of gains and we proceed to design the IWP with a different set of parameters. Hence we ensure that the IWP gains result in a stable controller and a stable closed loop system before attempting to evaluate the system performance. This is to conform with the IWP design requirement # 3.

The IWP optimization structure used to obtain the performance index in [2] is illustrated in Figure 2.5. As shown in the figure, the nominal system and the IWP system are augmented with signal conditioning blocks that provide appropriate scaling and frequency weighting for the inputs and the errors. We adopt the performance definition structure for our study from reference [2]. The data for the signal conditioning and frequency weighting blocks were obtained from the NASA Lewis Research Center. The explanation for each of the blocks is given below:

1. The "Command Loop Shaping" block consists of a loop scale factor and a first order lag for each of the controller loops. The external commands Z_c and U_L are white noise signals that are scaled and filtered by the command loop shaping block such that the inputs to the nominal and IWP systems have appropriate magnitude and frequency spectra.
2. The "Performance Error Weighting" block weights the performance errors between the nominal and IWP systems such that the low frequency (steady state) errors are given a higher weighting. This choice of weighting can be accomplished by choosing a weighting function that has a large magnitude at low frequencies and drops off to small magnitudes at high frequencies. This block also has a scale factor that is just the inverse of the scale factors used in the "Command Loop Shaping" block.
3. The "Actuator Position Weighting" block weights the actuator position errors so that the limited actuator is tracked accurately. This ensures smooth transfers between limited and unlimited actuator. This satisfies the IWP design requirement # 5.

In Reference [2], the performance index is solved as a white noise covariance optimization problem. According to this idea, the rms value of the performance errors and actuator position errors due to zero mean, unit variance white noise signal inputs is minimized. It can be shown that this rms norm is equivalent to solving the LQG problem with the performance index given by :

$$J_2 = E\{\lim_{T\to\infty} \frac{1}{T} \int_0^T (W_Z Z_e^2 + W_U U_e^2)dt\}$$

where the weights W_Z and W_U are chosen to penalize the performance errors and the actuator position errors respectively. This choice of the performance index ensures that IWP system performance matches closely with the nominal system performance, and ensures accurate actuator tracking.

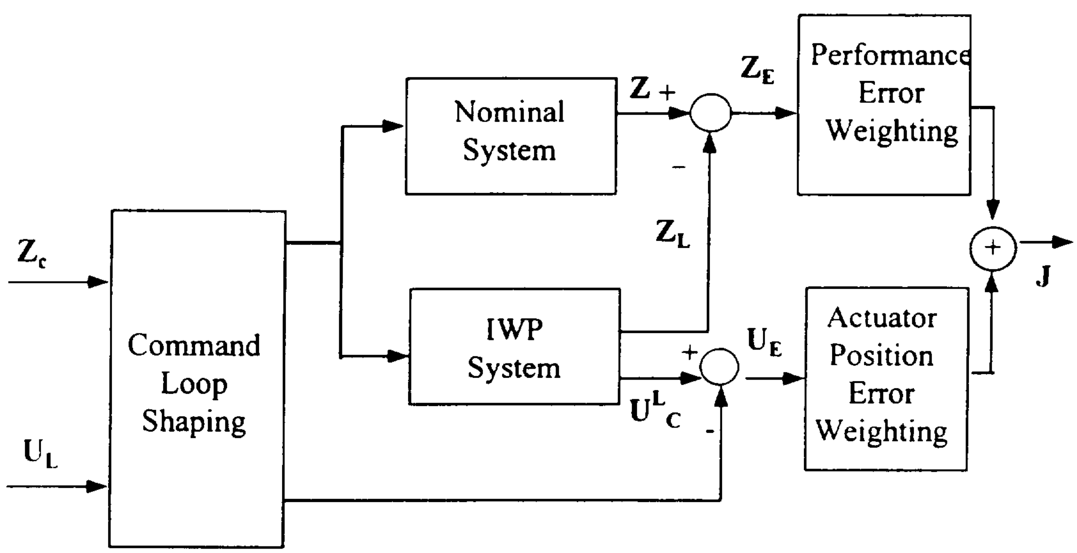

Figure 2.5 IWP Optimization Structure

The system performance for all actuator limit combinations is ensured by carrying out the optimization for all possible limit combinations. Thus, the optimization structure and the rms performance measure adopted from reference [2] satisfy the IWP design requirement # 4.

However, when the CAW gains are obtained by optimizing the rms performance measure as in reference [2], the closed loop IWP system might be directionally sensitive. In that case, the CAW scheme results in a deteriorated performance when actuator limits are encountered. Hence, we have to include the directional sensitivity measure in the performance index such that the directional sensitivity of the CAW system is minimized.

The directional sensitivity is the sensitivity of the system to the direction changes of the control vector, caused when the actuators encounter the operational limits. The directional sensitivity of the IWP system can be obtained by treating the actuator limits as uncertainties (with uncertain limits, the direction of control vector is uncertain). We can then reformulate the IWP system such that the limits on the system appear as diagonal input uncertainties to the system [6] as shown in Figure 2.6. Now the problem is reduced to determining the sensitivity of the system to structured uncertainties (in this case, the uncertainties have a diagonal structure) which is addressed by the structured singular value (μ) theory. Thus robustness of the IWP system to changes in control vector direction is given by its structured singular value (μ).

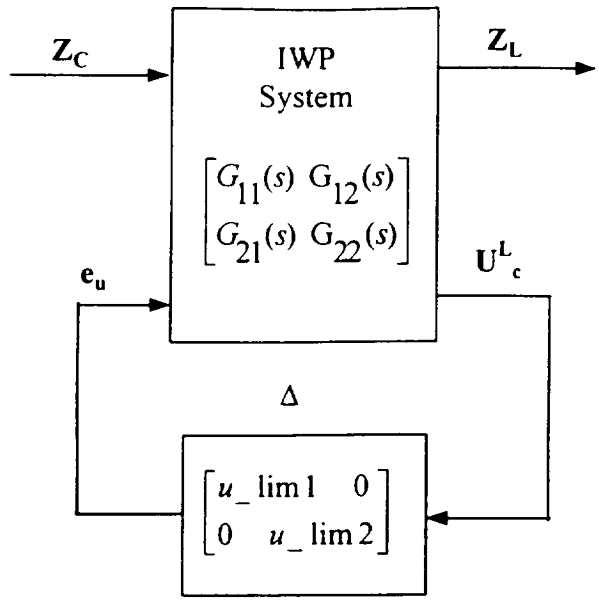

Figure 2.6 Limits as Diagonal Input Uncertainties

Structured singular value [15] can be defined as the inverse of the size of the smallest perturbation, Δ, that destabilizes the closed loop system shown in Figure 2.6 Alternatively, if we are given the perturbation Δ, we can determine whether the closed loop system is robustly stable for these perturbations from the structured singular value of the system. If the IWP system can be represented as G(s) with the partition structure as shown, then the closed loop system of Figure 2.6 is robustly stable for all perturbations within the specified bounds if and only if [15] :

$$\sup_{\omega} \mu\left[G_{22}(j\omega)\right] < 1 .$$

This states that the closed loop IWP system is robustly stable only if the maximum structured singular value of the transfer function between e_u and $U^L{}_c$,over all frequencies is less than unity. Here, U^L_c is the controller commanded control, and e_u is the quantity by which the controller commanded control exceeds the actuator limits. In this analysis, the actuator limits form the perturbation matrix Δ, as shown in the figure. This perturbation structure is diagonal and a 10% magnitude bound on these perturbations is assumed.

While including the directionality in the performance index for the CAW system, we first obtain the CAW gains that minimize the rms norm of the optimization structure presented earlier. Once we know the minimum rms norm (J_{2min}), we can minimize the structured singular value of the system while imposing a constraint on the rms norm, so that the resulting CAW gains minimize the directional sensitivity of the system, and provide a rms norm close to J_{2min}. This can be equivalently stated as :

$$\min \{ \sup_{\omega} \mu[G_{22}(j\omega)] \}$$

$$\text{subject to} \quad J_2 < kJ_{2\,min}$$

where we minimize the maximum structured singular value of the system transfer function between e_u and U^L_c subject to a rms norm constraint. The factor 'k' is a scale factor that denotes the performance degradation we are willing to tolerate in order to obtain a system less sensitive to the control direction. Hence, with this performance index definition, we satisfy both the performance requirements (requirement # 4), and the directional sensitivity requirement (requirement # 6) stated in section 2.2. It should be noted that the MAW scheme maintains the control vector direction, and hence we need not include the μ calculations for the MAW scheme. Thus the IWP schemes discussed here (CAW and MAW), together with the performance index definition satisfy all the general IWP design requirement stated in section 2.2.

We conclude this chapter by implementing the IWP schemes on two example problems presented in [6]. These examples illustrate the role played by the directional sensitivity of the limited system on the choice of IWP scheme.

Example 1:

The first example we study is a nominally stable 2 input 2 output system that is represented as :

System A :

$$P = \frac{4(0.1+s)}{s}R^{-1} \quad \text{with } R = \begin{pmatrix} 4 & 5 \\ 3 & 4 \end{pmatrix};$$

$$K = \frac{1}{4(0.1+s)}R$$

In the above representation, P and K represent the plant and the controller respectively. The schematic of the closed loop limited system is presented in Figure 2.7. In this figure, d is the desired input to the plant, y is the plant output and e is the error. The limits on the actuator are assumed to be [-1,1] as shown in the figure. The response of the nominal and limited systems for a step input of amplitude $[0.61\ \ 0.79]^T$ is presented in Figure 2.8. The overshoot and large oscillations exhibited by the limited system are the effects of windup. Here the windup effects are caused because the dynamics of the controller is relatively slower than that of the error when the operational limits are encountered.

Figure 2.9 shows the implementation of CAW scheme for this system. As seen in this figure, the actuator error is fed back to the controller through a set of CAW gains. Since we are considering the examples to illustrate the effect of directional sensitivity, we ease the IWP design requirement 5 which is concerned with actuator tracking. Also, we do not consider the signal conditioning and error weighting blocks since this is just a theoretical model and has no practical significance. The CAW gains of this system are obtained by first optimizing the J_2 norm of the limited IWP system, and then minimizing the structured singular value with a constraint on the J_2 norm. A 10% deterioration in the rms performance was allowed to reduce the directional sensitivity of the system.

The response of this CAW system to the step input $[0.61 \quad 0.79]^T$ is presented in Figure 2.11. The oscillations exhibited by CAW system are similar to that of the unprotected system, but the amplitudes of the oscillations are lesser. The CAW scheme, thus maintains the stability and provides a performance better than the unprotected system when the system encounters operational limits. This is supported by the structured singular value (μ) analysis of the limited system. This analysis assumes that the actuator limits enter the system as diagonal input uncertainties. The structured singular value is computed by assuming a 10% magnitude bound for the uncertainties in the actuator limits. The μ plot presented in Figure 2.12 shows that the maximum μ of the CAW system is less than unity for all diagonal uncertainties satisfying the above magnitude bound in the frequency range (0.001-10)Hz. Hence, the CAW system is robustly stable to direction changes of the control vector.

MAW scheme is implemented for this system in Figure 2.10. The design variable β was set at '1000' for this problem. A smaller value for β does not provide adequate IWP. It must be noted here that with a high β, the resulting system has a high bandwidth and this might cause difficulties in a practical situation. Since this is just an example, we assume that it is permissible to have a high β and that there is no limit on the actuator bandwidth. The response presented in Figure 2.11 shows that the oscillations and overshoot associated with windup are eliminated completely with the MAW scheme. Since the MAW scheme maintains the direction of control vector when the system encounters operational limits, the performance of the MAW system matches closely with the nominal system performance.

Thus we can conclude that while the limited system A is robustly stable to control vector direction changes caused by actuator limits, its performance is sensitive to control vector direction changes. Since CAW scheme does not maintain the direction of control vector, it causes performance degradation when the actuator limits are encountered. The

MAW scheme maintains the control vector direction and hence provides a performance close to nominal performance when limits are encountered.

Example 2 :

The second example is given by :

$$P = P_0 P_1$$

with $P_0 = \frac{4(0.1+s)}{s} R^{-1}$, $P_1 = \begin{pmatrix} \frac{2(10-s)}{10+s} & 0 \\ 0 & \frac{5-s}{5+s} \end{pmatrix}$;

and $K = \frac{1}{4(0.1+s)} R$

The response of the nominal and limited systems for a step input of $[0.36 \quad 0.93]^T$ are presented in Figure 2.13. As seen in the figure, the system becomes unstable upon encountering actuator limits. It was found that CAW scheme could never stabilize the limited system for any set of gains. An unstable response of the CAW scheme is presented in Figure 2.14. This is supported by the structured singular value analysis. A 10% uncertainty in the actuator limits was assumed similar to the previous example. The structured singular value plot presented in Figure 2.15 shows that the since the maximum μ is greater than unity, and hence the CAW system is not robustly stable to uncertain actuator limits.

The response of the MAW system is presented in Figure 2.13. Since the MAW scheme maintains the control direction, it provides a stable system with an acceptable performance. We can conclude the following from the above examples :

- When limited, system A is robustly stable to control vector direction changes. Hence the CAW scheme provides a stable solution when actuator limits are encountered. However, the response provided by CAW scheme is degraded while MAW scheme provides an acceptable performance. This could be due to the sensitivity of the performance to the control vector direction.
- System B when limited, is not robustly stable with control vector direction changes. Hence CAW scheme could not stabilize the limited system, while the MAW scheme provides a stable, acceptable response.
- Requirement # 6 presented in section 2.2 has to be satisfied to provide acceptable IWP characteristics. While the MAW scheme satisfies this by maintaining the control direction, with CAW scheme this is achieved by including the directional sensitivity in the performance index definition.

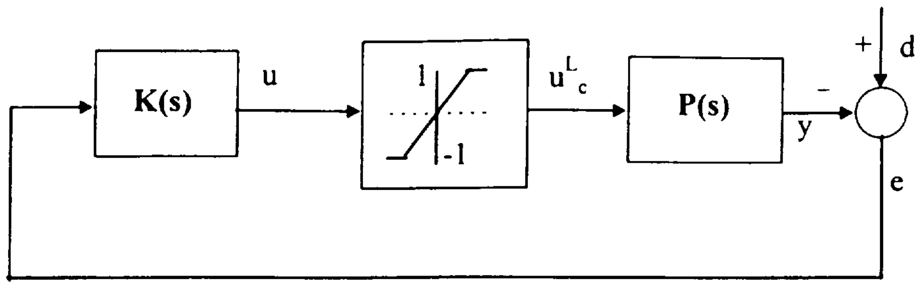

Figure 2.7 System Schematic Diagram

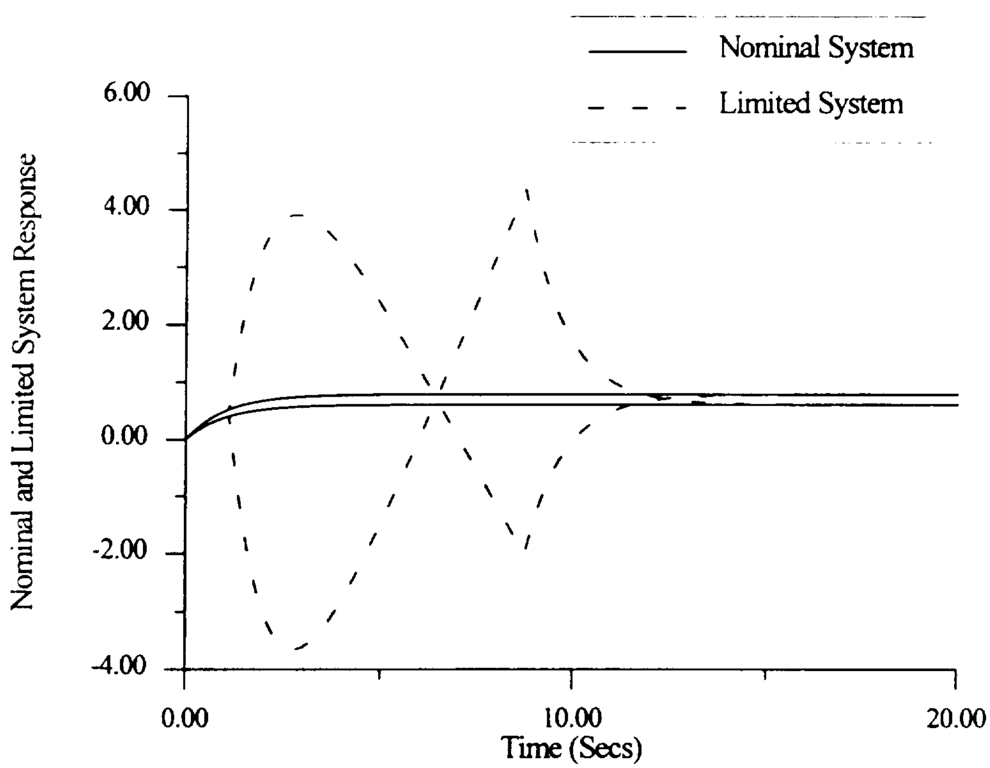

Figure 2.8 Nominal and Limited Response for System A

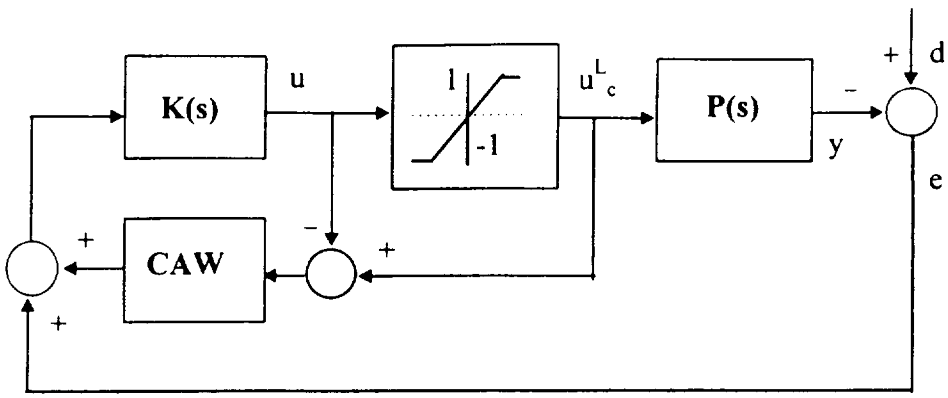

Figure 2.9 CAW Scheme Implementation

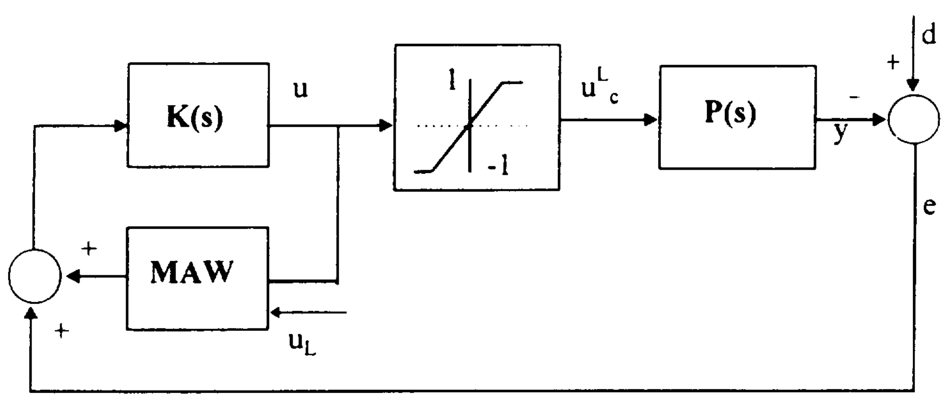

Figure 2.10 MAW Scheme Implementation

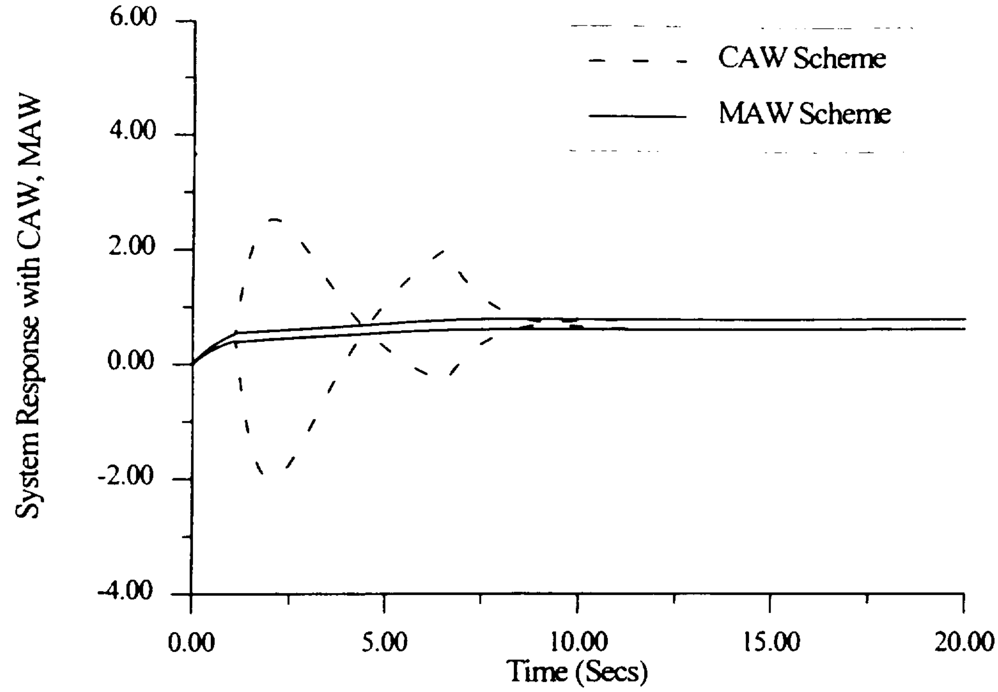

Figure 2.11 CAW, MAW Scheme Responses for System A

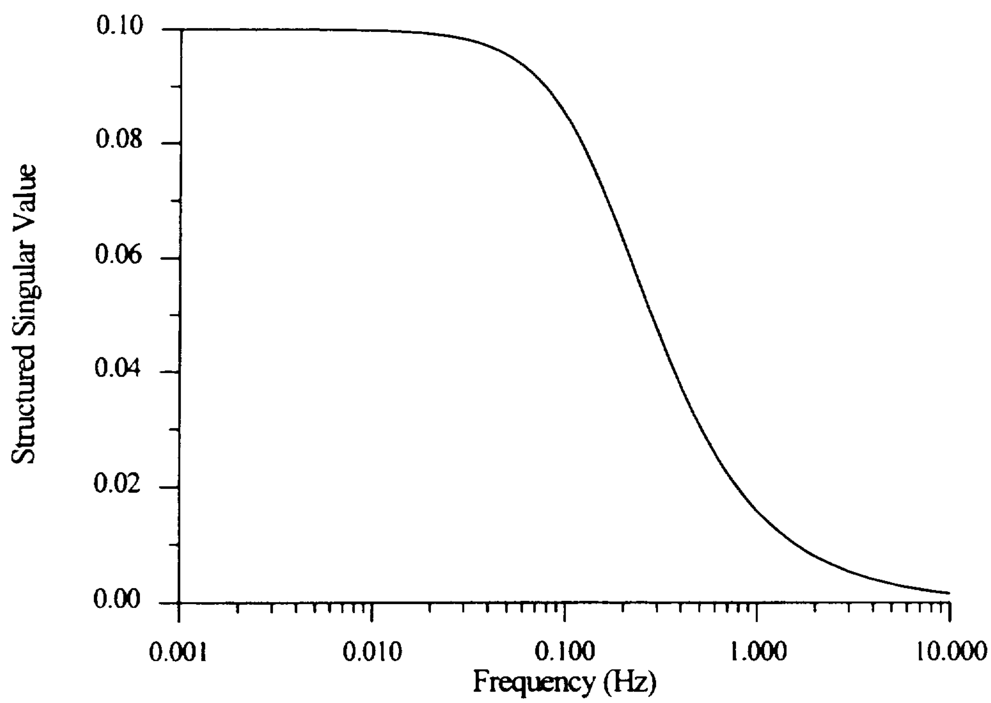

Figure 2.12 Structured Singular Value for diagonal uncertainties for System A

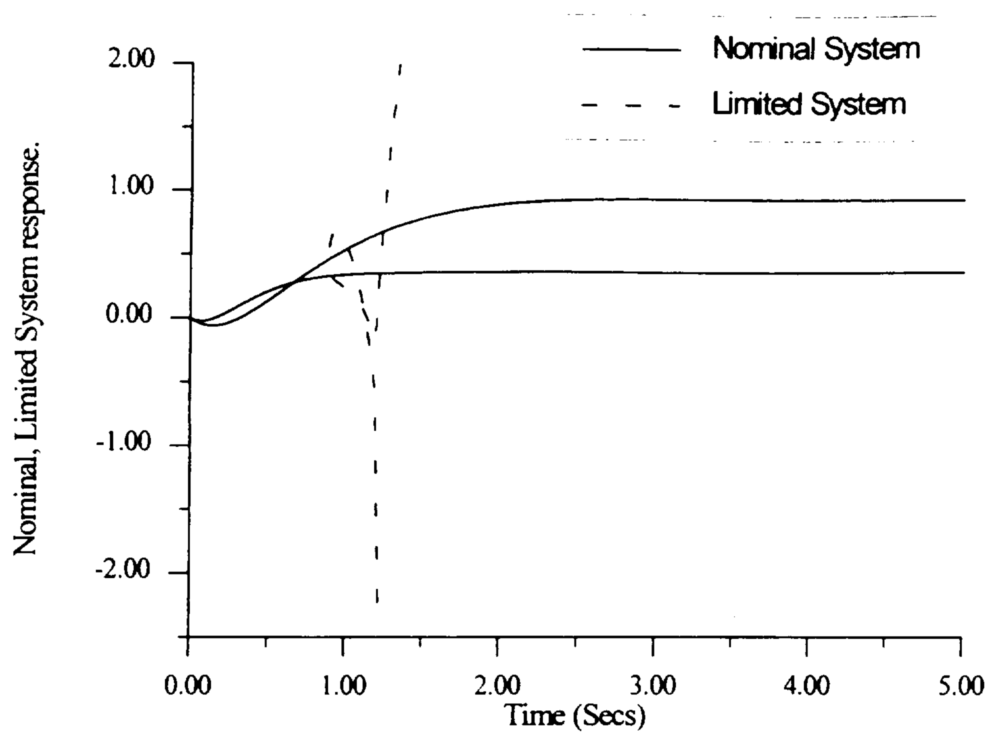

Figure 2.13 Nominal, Limited Response for System B.

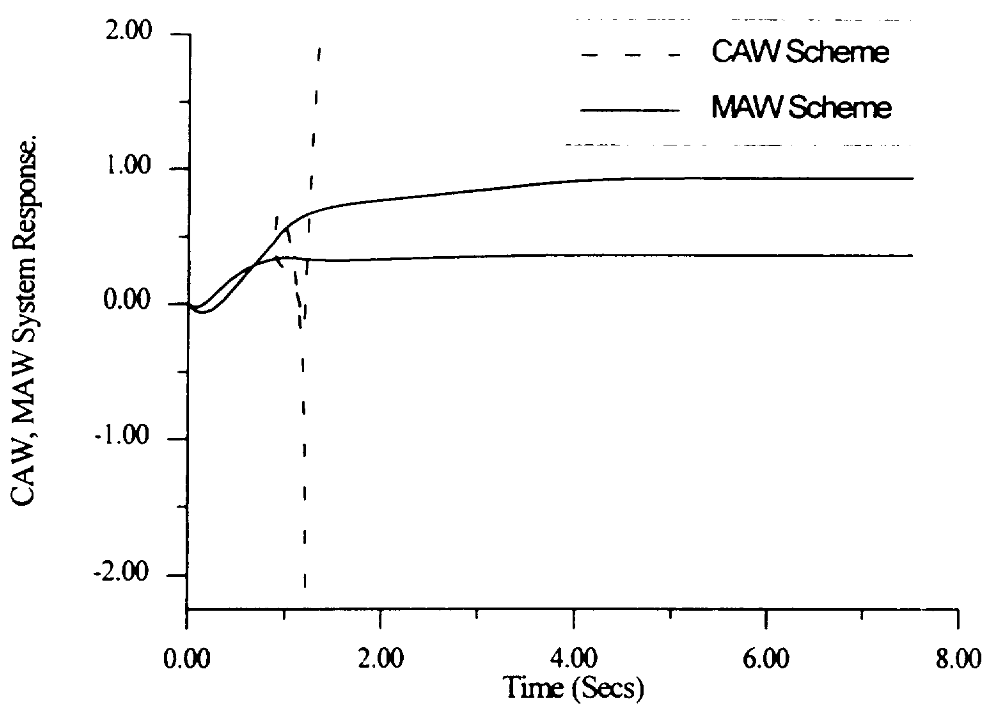

Figure 2.14 CAW, MAW Scheme Responses for System B

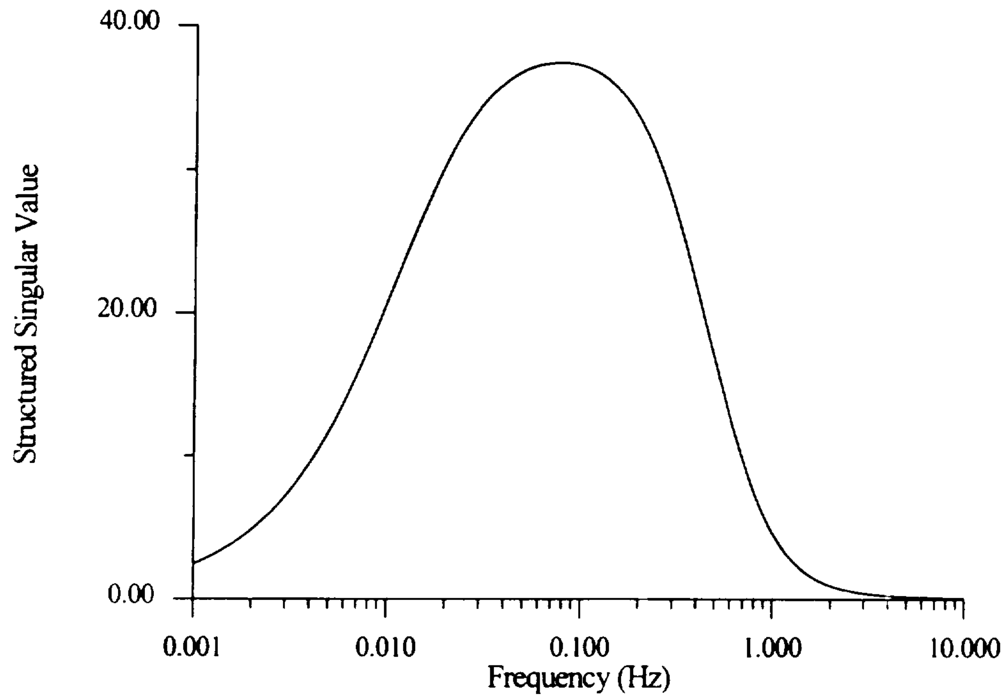

Figure 2.15 Structured Singular Value for diagonal uncertainties for System B

Chapter 3

Optimization Technique - GA with Stochastic Coding

Many of the modern day problems in engineering, medicine, economics, sociology, and other areas can be solved in different ways. The question then is to determine the strategy that solves the problem in some optimal sense. This is answered by quantifying the criteria we are trying to optimize, and then using an optimization technique to optimize that criteria. The optimization technique used to solve a non-linear optimization problem could play a crucial role in the overall solution process. In some of these problems, the optimization step takes the maximum computational time and effort expended to solve the problem. Hence, by a judicious choice of the optimization technique it might be possible to significantly reduce the overall effort required to arrive at an optimal solution. In this chapter, we examine a modified Genetic Algorithm optimization technique to obtain the optimal parameters for a test problem given in Reference [16]. The modified genetic algorithm approach uses a stochastic parameter encoding to find the optimal set of parameters [9]. This test problem has non-linear interactions between the parameters (i.e. it cannot be solved for each parameter independently), and several locally optimal solutions. The modified GA technique is shown to work effectively for this test problem.

3.1 Conventional Optimization Techniques

Traditional optimization techniques can be classified as calculus based, enumerative, or random search procedures. The calculus based approach could be either a direct or an indirect method. In the indirect method, the gradient of the cost function (with respect to the parameters being optimized) is set to zero and the resulting equations are solved to obtain the optimal set of parameters. In a direct approach, we move in the direction of the steepest gradient (of the cost function with respect to the parameters being optimized) to find a local optimum. The greatest disadvantage of either of these methods is that it is local in scope. It just gives the best solution in the neighborhood of the point being considered in the search space. Also, in cases where the gradient information is hard (or impossible) to obtain, these methods would fail.

In an enumerative method, the objective function is evaluated at a large number of points in the search space and the optimal solution is the best cost obtained over all the points. Such a technique would obviously become inefficient for large search spaces. In a random search technique, the objective function is evaluated at some random points in the search space. Though this method might provide a global solution, it requires a large number of points to be sampled before the optimal solution is found. Another drawback of the random search procedure is that it is difficult to incorporate a stopping criterion in the algorithm.

Hence, we can summarize by stating that the calculus based methods, though directed towards an optimum, suffer because of their local scope. The random search technique though has the potential to obtain the global optimum, suffers due to the lack of a directed search. A search technique referred to as Genetic Algorithms (GA), overcomes the shortcomings of the conventional techniques listed above while attempting to retain the positive aspects of both of them. In other words, GA is a search technique that provides a directed random search [17].

3.2 Genetic Algorithms (GA)

Genetic Algorithms are search techniques based on principles of natural selection and genetics. GAs combine a Darwinian survival-of-the-fittest strategy with a probabilistic information exchange between various feasible solutions. GAs are different from the conventional gradient based search techniques in the following ways [17] :

1. GAs operate on a coding of the parameter set, not on the parameters themselves.
2. GAs search for the optimum solution from a population of points, not from a single point.
3. GAs need information about the objective or cost function, but do not need any derivative or other auxiliary information.
4. GAs use probabilistic transition rules to move across the search space to obtain an optimal parameter set.

Since a GA does not require any derivative information to guide their search, it is suitable for problems where the derivative information is difficult or impossible to obtain. Other features of the GA that makes it a powerful optimization technique are its capability to incorporate apriori knowledge of the solution space, multiple objectives and multiple solutions. A typical GA search process is depicted below:

Step 1 : (Initialization) Obtain a random initial population consisting of *n* individuals representing *n* points in the search space. Each individual is a binary coded string of the parameters of the optimization problem. This binary coded string is referred to as the genotype while the actual values of the parameters are referred to as the phenotype.

Step 2 : (Fitness Evaluation) Each individual of the above population is first decoded to obtain the phenotype represented by the individual. These decoded parameter values are used to evaluate fitness of each individual using the fitness function.

Step 3 : (GA Selection) The selection procedure identifies the fit individuals of the current population that would be the parents for the next generation. Two individuals are chosen with probabilities proportional to their relative position in the current population, measured either by their contribution to the mean fitness value of current generation (proportional selection), or by their rank (linear ranking selection).

Step 4 : (GA Recombination) Two different offspring are produced due to recombination of the two parental genotypes of step 3 by means of a crossover with probability p_c. Steps 3 and 4 are repeated until we obtain n individuals for the next generation.

Step 5 : (Mutation) The offspring obtained in the above step are finally mutated with a small probability (p_m). The mutations are assumed to work on individual binary bits, either by reversing a one to a zero or otherwise.

The central processing power of the GA arises from the successful sampling and recombination of low order, highly fit, short defining length schemata into strings of better fitness. Holland's schema theorem proves that such low order, highly fit, short defining length schemata, also known as building blocks, grow exponentially in a population [17].

GAs have been shown to do well on problems with small number of parameters to optimize, and are difficult to solve using traditional optimization techniques [9]. However, for a problem with large number of parameters, large string lengths could result when the parameters are coded in a binary form. In reference [9], it has been shown that the probability of finding higher order schemata in a population of large strings approaches zero rapidly, as the string length increases. This means that the GA would attempt to solve the problem by mostly manipulating the lower order schemata, in which case, it would take a long time before the optimal solution is found. It is possible to increase the probability of the higher order schemata by having a larger population, but this calls for enormous computational resources. Hence, we need to modify the GA approach so that its effectiveness is maintained while solving problems with large parameters.

Reference [8] examines the techniques to modify the regular GA to maintain its effectiveness while solving large parameter problems. In the sensitivity based method, the problem of large string lengths is avoided by choosing a few sensitive parameters and optimizing the performance index with respect to these sensitive parameters. However, this approach is highly problem dependent, and for large parameter problems, even the task of determining the sensitivity of each parameter could result in a significant computational overhead. Also, if most of the parameters of the optimization problem are highly correlated, this approach would fail.

In GA-local search hybrid technique, the GA searches the solution space at a global level while the local/direction search algorithm searches locally around the solutions provided by the GA. This enables us to have a coarser encoding structure for the GA, thereby reducing the string length. This hybrid technique however results in a computational overhead if the local search does not improve the solution provided by GA.

3.3 Stochastic Genetic Algorithms

The previous sections described the problems faced when we use the GA and the modified GA techniques for solving large parameter problems. In this section, we detail a GA with stochastic encoding of the parameters (referred to as a Stochastic GA) that overcomes these problems and converges quickly to an optimal solution.

Stochastic Genetic Algorithm is first presented in [9] as an approach to effectively solve problems with large number of parameters. Some of the features of the Stochastic GA as given in [9] are :

1. Each discrete possibility as decoded from the binary string, represents a search region and not a single value.
2. The regions defined above are dynamic, (i.e.) the same genotype could represent a different region at a different time.

3. The search regions are altered based on the GA evolution.
4. The search region is not explicitly constrained.
5. No region is completely discarded.

These features can be implemented in the GA by encoding the search region as a binary string. The search region is represented by a multivariate Gaussian distribution with a mean vector(μ) and the variance matrix(Σ). The mean vector gives the expected values of the parameters in the search region, and the variance matrix gives the probability of finding an optimal parameter set in a particular area of the search region. The stochastic children are obtained by sampling this multivariate Gaussian distribution. In this study, two variations of the stochastic GA were implemented for a test problem. The two approaches differ in the way the stochastic children are obtained from the parent's phenotype. In the first approach (Approach A), the stochastic child is obtained by varying all the parameters in the parent's phenotype using the multivariate Gaussian distribution (Reference [9]). In this study, the variance matrix used in the multivariate Gaussian distribution of Reference [9] is adapted continuously as the GA population evolves. This adaptation of the variance matrix helps to exploit the most promising regions as the GA explores the search region. In the second approach (Approach B), only one parameter is stochastically varied in each string of the GA population. The difference between the two approaches can be intuitively pictured as follows. For the stochastic child in the approach A, all the features of the child are slightly different from the parent, while for the approach B, the stochastic child retains all but one feature of the parent.

Thus, in approach A, change in the fitness due to the overall change in features is detected, where as in approach B; the contribution of each of the parent's feature to the fitness is determined. Hence, with approach B it is possible to fine tune each feature independently to obtain an improved fitness. The algorithm that implements the above details is presented below.

Step 1: (Initialization) An initial population of *n* individuals, characterized by its genotype is randomly generated. Each individual's genotype is a binary string representing a search region of the parameters. The search region is encoded in the binary string by means of a multivariate Gaussian distribution with mean vector μ and a variance matrix Σ.

Step 2 : (Stochastic Phenotype Variation)

Approach A : Each of the *n* individuals produce *m* offspring, so that a total of *mn* new individuals are available. The search regions represented by these offspring's are obtained by displacing the parent's mean vector(μ), with the variance matrix(Σ), resulting in a new mean vector for the offspring.

Approach B : In this approach, only one parameter in the parent's mean vector (μ) is displaced by the variance of that parameter choice. Thus, in both the above approaches, the phenotype of the descendent is thus slightly different from that of the parent, while the genotype is the same.

Step 3 : (Filtering) Out of *mn* individuals in step 2, only *n* individuals become parents. The phenotypes of the chosen individuals are used to redo the coding, resulting in a modified mean vector. In approach A, the variance matrix of the offspring is altered based on $(1/5)^{th}$ success rule. According to this rule, the variance of the Gaussian distribution is decreased if atleast one out of five phenotype variations (of the same genotype) in step 2 results in an improvement of the performance index. Otherwise, the variance is increased. The algorithm that implements this is given in Figure 3.3. The variance matrix is not altered in approach B.

Step 4 : (GA Selection) Two parents are chosen with probabilities proportional to their relative position in the current population, measured either by their contribution to the mean performance of the current generation (proportional selection) or by their rank (linear ranking selection). In a tournament selection procedure, two best individuals from a random number of individuals are chosen for the next generation.

Step 5 : (GA Recombination) The two parents selected in step 4 are recombined with a probability of crossover p_c, giving rise to two new parental genotypes. Steps 4 and 5 are repeated until we have *n* new individuals representing the next generation.

Step 6 : (GA Mutation) The new generation obtained above undergoes a mutation operation, where the individual bits of each offspring are mutated (reversed from a one to a zero, or vice versa) with a small probability (p_m).

Initially when the search is started, a symmetric multivariate Gaussian distribution is assumed over the search region. The initial Gaussian distribution for a two parameter optimization problem is illustrated in Figure 3.1(a). A two dimensional planform view of the Gaussian distribution and the parameter choices are shown in Figure 3.2(a). In approach A, as the GA population evolves, the distribution shifts towards more promising regions and the variance matrix is altered to exploit the promising regions. This is shown in Figure 3.1(b) for the two parameter case, where the Gaussian distribution is shown after *N* generations. Figure 3.2(b) shows the planform view of the Gaussian distribution after *N* generations.

3.4 Application of Stochastic GA for a Test problem

The stochastic GA methods were applied to a test problem given in [16]. The function $f(x_1,x_2)$ given below has strong nonlinear interactions between the variables x_1 and x_2 i.e. the optimal value for one parameter cannot be determined independent of the other. Such problems are not easily solved by the random hill climbing technique. The nonlinear interaction between the variables of the function can further be increased by constructing a composite function of more parameters from this primitive function. The function $F(x_1,x_2....x_n)$, can in general be a function of any number of variables and is constructed from the primitive function of two variables $f(x_i,x_{i+1})$, as shown.

$$f(x_1,x_2) = (x^2{}_1 + x^2{}_2)^{0.25}[\sin^2(50(x^2{}_1 + x^2{}_2)^{0.1}) + 1.0]$$

$$F(x_1,x_2,...x_n) = f(x_1,x_n) + \sum_{i=1}^{n-1} f(x_i,x_{i+1}) \qquad x_i \in [-100,100]$$

In this study, the function $F(x_1,x_2,\ldots.x_n)$ is optimized for 51 parameters using the GA techniques described before. The optimization was carried out with both the regular GA and the modified Stochastic GA approaches. For the regular GA approach, a population size of 101 was considered. The parameters were assumed to lie in the range [-100,100], 6 bits were used to encode each parameter, a probability of crossover of 0.77 and a probability of mutation of 0.0077 were used.

For the stochastic GA, 3 offspring were assumed to be produced at the step 2 of the algorithm. An initial range of [-100,100] is assumed, and in the approach A, the standard deviation in step 3 is decremented by 5%. Also the increment step for the standard deviation was set at 2%. (Refer to algorithm in Figure 3.3 for details). The population size was set at 51. The probability of crossover and mutation for the stochastic GA were set at same levels used for the simple GA, and tournament selection procedure was used for both simple GA and stochastic GA.

For the stochastic GA in approach B, first parameter was fine tuned in the first individual of the population, second parameter in the second individual, and so on. Hence for this approach, we must have as many individuals in the GA population as the number of parameters. For problems with large number of parameters, we might have to carry out a sensitivity analysis, and then use this technique to determine the optimal values for the few sensitive parameters.

3.5 Results of Stochastic GA search

The results obtained from the application of stochastic GA and the regular GA are presented in Figure 3.4. As seen in the figure, the stochastic GA in approach B converges quicker than the regular GA and the stochastic GA in approach A. The stochastic GA in approach B also converges to a local optimum closer to the global optimum than the regular GA and the stochastic GA in approach B. The drawback of approach B, however is that we must have a population size equal to the number of parameters of the optimization problem. Hence approach B becomes computationally expensive for problems with large number of parameters. The stochastic GA approach A when implemented for such problems would converge to the optimum quicker than the regular GA method.

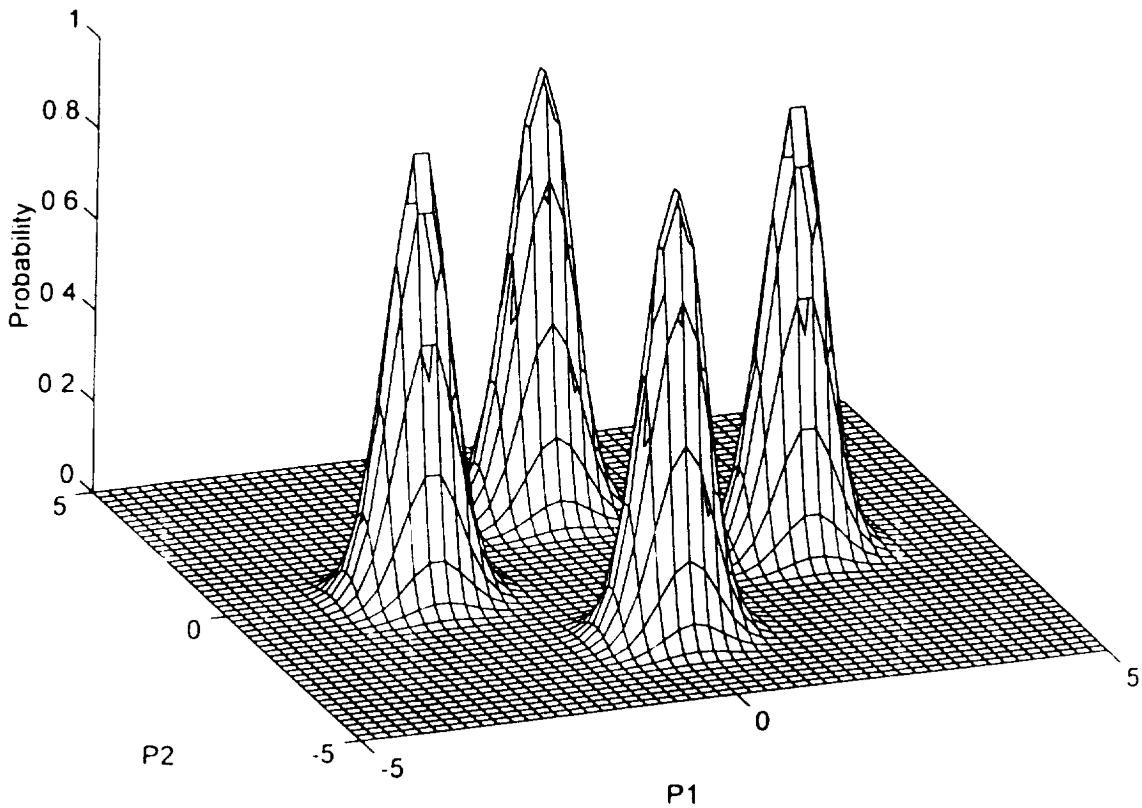

Figure 3.1(a) Initial symmetric Gaussian distribution for a two parameter case.

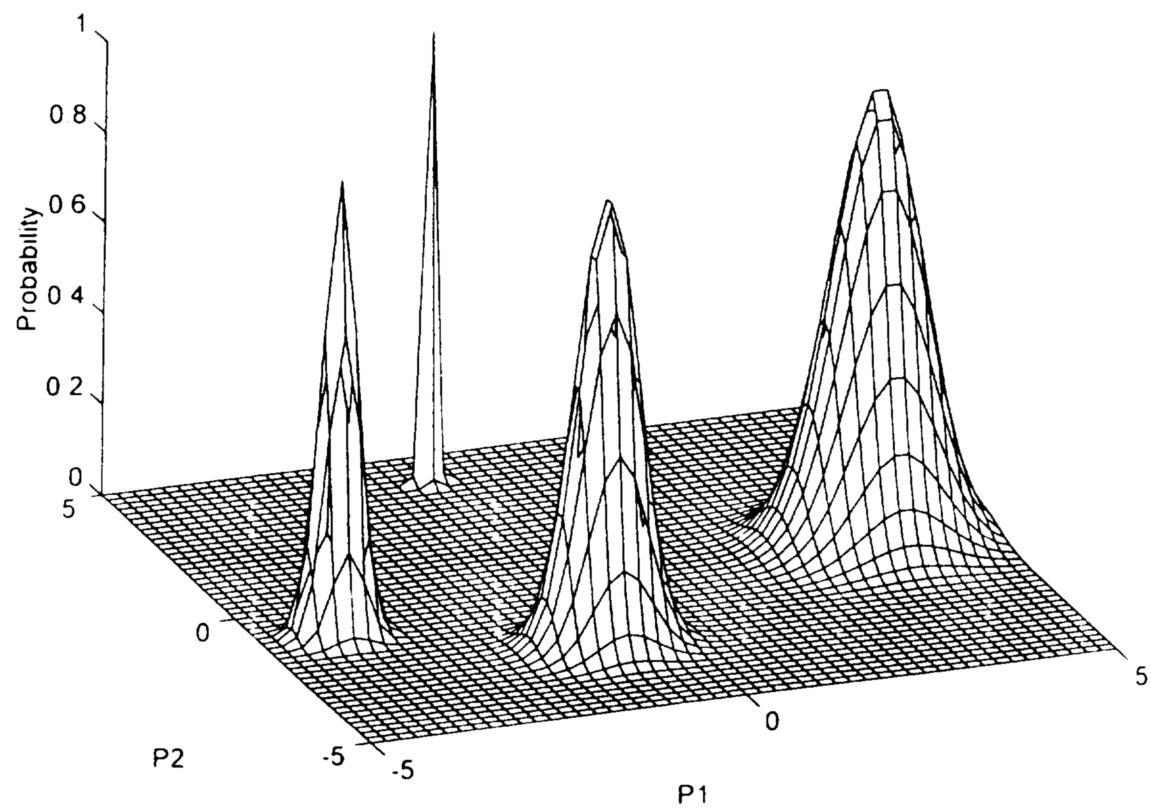

Figure 3.1(b) Gaussian Distribution for a two parameter case after GA evolution.

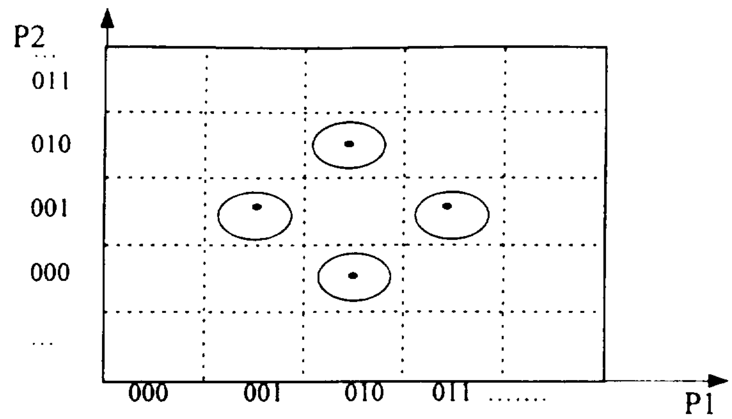

Figure 3.2(a) Planform view of the Initial Gaussian distribution and binary coded parameter choices for a two parameter case.

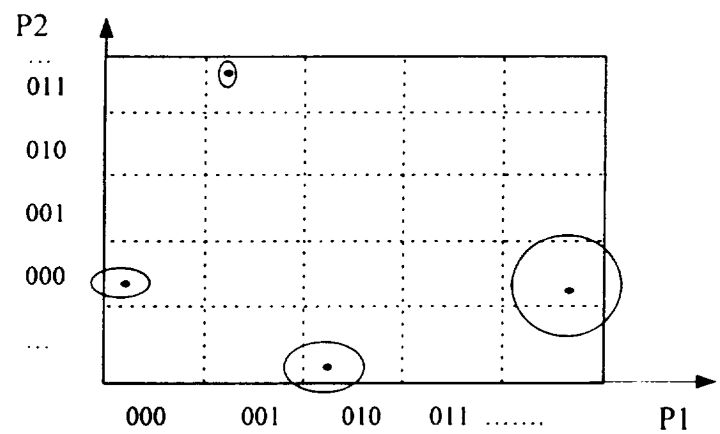

Figure 3.2(b) Planform view of the Gaussian distribution and the binary coded parameter choices after GA evolution (*N* generations)

PI_parent ← **fitness**(*Parent)*

PI_bestchild ←**fitness**(*bestchild*)

If (*PI_bestchild* better than *PI_parent)*

Phenotype(*Parent*) ← **Phenotype**(*bestchild)*

end

If (Stochastic Phenotype Variation in step 2 improves fitness atleast once in 5 variations)

decrement **Variance**(*Parent_Genotype*) by 5%

else

increment **Variance**(*Parent_Genotype*) by 2%

end

Figure 3.3 Stochastic GA in approach A. Algorithm that adapts the variance of the Multivariate Gaussian distribution that encodes the parameters as binary strings. "fitness" returns the fitness of the individual passed to function, Parent and bestchild refer to the parent and the best stochastic child obtained at step 2 of the algorithm.

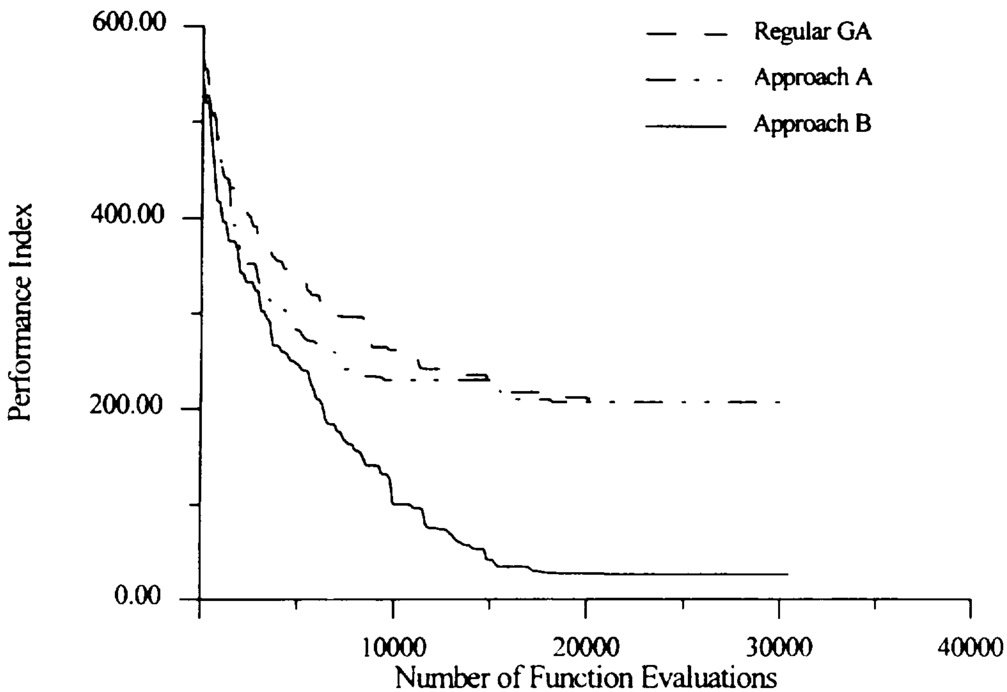

Figure 3.4 Evolution using stochastic GA (approaches A and B), and regular GA

Chapter 4

Application to STOVL Aircraft Engine Controller Problem

The advanced maneuvering capabilities of modern aircraft require the propulsion system to play a vital part in the aircraft control. Separately designed flight and propulsion control systems are however, inadequate for efficient operation of the aircraft with reasonable pilot workload [18]. Hence, an Integrated Flight Propulsion Control (IFPC) system that accounts for all the subsystem interactions is necessary to obtain an optimal system performance with minimal pilot workload. Such a centralized design results in a high-order IFPC system that poses implementation problems. The high-order centralized controller is therefore partitioned into lower order airframe and propulsion controllers with a specified interconnection structure. The partitioning is such that the performance and the robustness characteristics of the assembled partitioned controllers match that of the centralized controller [18]. Such a partitioning also makes it easier to perform independent testing of the airframe and propulsion subcontrollers. Also, the partitioning allows the designer to address the airframe and propulsion system non-linearities separately, and to design the integrator windup protection independently for each subcontroller [19].

Preliminary evaluation of the NASA Lewis linear integrated flight/propulsion control system has indicated the need for integrator windup protection on the propulsion and the airframe subsystems for large command inputs. In this chapter, the CAW and MAW techniques will be applied to provide IWP for the linear model of the engine subcontroller that forms the part of IFPC system of a STOVL aircraft.

4.1 The STOVL Aircraft Engine Control System Model

The aircraft considered for study is a model of a supersonic STOVL aircraft powered by a high performance turbofan engine. A schematic diagram of the aircraft with various controls is shown in Figure 4.1(a) [20]. The engine control system of the STOVL aircraft is equipped with the following controls :

- ejectors to provide propulsive lift at low speeds and hover.
- a vectoring ventral nozzle for pitch control and lift augmentation at low speeds.
- a two-dimensional Convergent-Divergent (2D-CD) vectoring aft nozzle with after burner for supersonic flight.
- compressor fan speed.

The control variables that control the above parameters are respectively :

- ejector butterfly angle (ETA, degrees)
- ventral nozzle area (A78, $inch^2$)
- aft nozzle area (A8, $inch^2$)
- fuel flow rate to engine (WF36, lbm/hr)

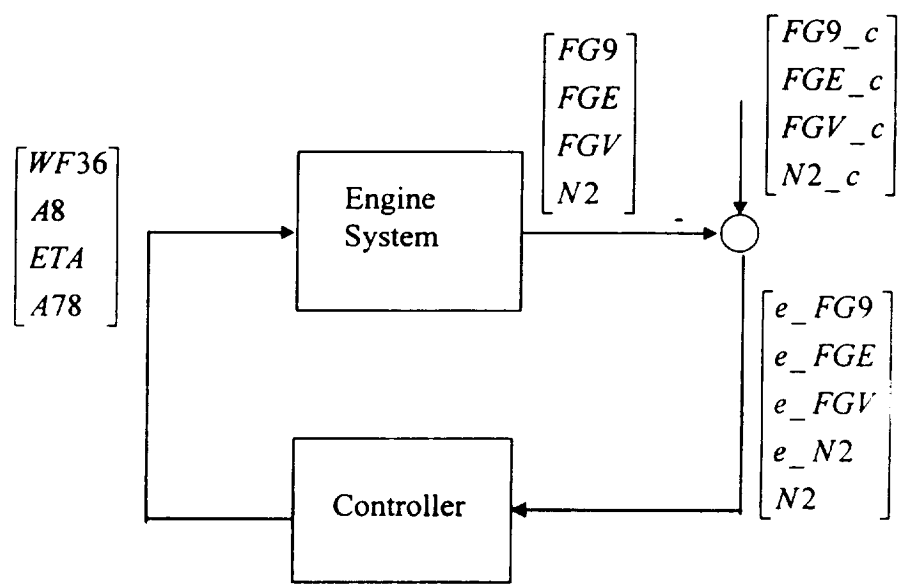

Figure 4.1(b) Schematic of the Engine Control System

The engine state space model can be represented as (Figure 4.1(b)):

$$\dot{X}_e = A_e X_e + B_e U_e$$
$$Y_e = C_e X_e + D_e U_e$$

Where A_e ,B_e ,C_e ,D_e are the system matrices of the engine (given in the appendix)

X_e is the engine state vector,

$$U_e = \begin{bmatrix} \mathrm{WF36} \\ \mathrm{A8} \\ \mathrm{ETA} \\ \mathrm{A78} \end{bmatrix} \; ; \; Y_e = \begin{bmatrix} \mathrm{FG9} \\ \mathrm{FGE} \\ \mathrm{FGV} \\ \mathrm{N2} \end{bmatrix}$$

FG9 - aft nozzle thrust (lbf)

FGE - ejector thrust (lbf)

FGV - ventral nozzle thrust (lbf)

N2 - compressor fan speed (rpm)

The state space model of the controller can be written as (Figure 4.1(b)):

$$\dot{X}_c = A_c \; X_c + B_c \begin{bmatrix} \mathrm{e\_zeng} \\ \mathrm{N2} \end{bmatrix}$$

$$Y_c = C_c \; X_c + D_c \begin{bmatrix} \mathrm{e\_zeng} \\ \mathrm{N2} \end{bmatrix}$$

where A_c, B_c, C_c, D_c are the system matrices for the controller (given in the appendix),

X_c is the controller state vector,

The vector of performance errors : $\mathrm{e\_zeng} = \begin{bmatrix} \mathrm{FG9\_c - FG9} \\ \mathrm{FGE\_c - FGE} \\ \mathrm{FGV\_c - FGV} \\ \mathrm{N2\_c - N2} \end{bmatrix}$;

The inputs to the controller are: $u_c = \begin{bmatrix} \mathrm{e\_zeng} \\ \mathrm{N2} \end{bmatrix}$

The outputs of the controller are: $Y_c = \begin{bmatrix} \text{WF36} \\ \text{A8} \\ \text{ETA} \\ \text{A78} \end{bmatrix}$

4.2 Selection of limited Actuators

The engine model considered above has four inputs, four control variables, and four actuators. The four actuators control the fuel flow to the engine, ejector butterfly valve angle, ventral nozzle area, and the aft nozzle area respectively. It would be ideal if we designed the IWP for all possible actuator limit combinations. However, observations at NASA Lewis Research Center have shown that in practice that situation never occurs. Hence, to simplify the analysis and to study a more realistic situation, we consider a case where two of the four actuators are limited. It was expected that the fuel flow to the engine would be strongly coupled with the ejector thrust generated by the engine. Since it was desired to study the effectiveness of the IWP schemes for systems with strong coupling, the fuel flow to engine (WF36) and ejector thrust (FGE) were assumed to be limited. This choice of actuator limits is shown in the engine closed loop block diagram in Figure 4.2. Here the ejector butterfly valve angle and the fuel flow rate commanded by the controller are limited by the hard limits '*u_lim1*' and '*u_lim2*' (blocks 31 and 41). Other actuators, the aft-nozzle area and the ventral nozzle area are assumed to be unlimited.

4.3 IWP Optimization Structure

For the selected actuators, the IWP performance definition structure was setup as shown in the second chapter. The performance definition structure for the engine controller problem is given in Figure 4.3. Blocks 1-4, 11-13 are the command loop shaping blocks that scale the input magnitude and the frequency spectra. More specifically, block 11 is the scaling block while the blocks 1-4, 12, 13 are the frequency shaping blocks with a first order lag. Blocks 5-10 and 15 weight the performance errors and actuator position errors as shown. Block 15 is the scaling factor that is same as the inverse of block 11, blocks 6-9 weight the performance errors between the IWP system and the nominal system, and blocks 5, 10 weight the actuator position errors. The errors between the nominal and the IWP systems are obtained in block 14, "Error with IWP", the details of which are shown in Figure 4.4.

In Figure 4.4, block 7 is the nominal engine control system without limits, while block 23 is the limited engine system for which the IWP has to be designed. The performance errors and the actuator position errors are shown in this figure. The details of block 6 are shown in Figure 4.5, where block 13 is the engine system, block 6 is the controller, and block 21 is the scheduling gain to extend the 80 knot nominal design controller for the 100 knot engine. The IWP is built inside the control system of the limited engine model, i.e. within block 6 of Figure 4.5.

4.4 CAW Scheme for Engine Control System

The CAW scheme is implemented in Figure 4.6. The terms of the gain block '*ceng_iwp*' are the gains of the CAW scheme and are to be obtained by optimizing the performance index. The actuator error is fed to the '*ceng_iwp*' block as shown and the outputs from the '*ceng_iwp*' block are added to the state derivative calculations. With CAW implemented, equations for the controller can be modified as :

$$\dot{X}_c = A_c \; X_c + B_c \begin{bmatrix} \text{e\_zeng} \\ \text{N2} \\ \text{e\_ueng} \end{bmatrix}$$

$$Y_c = C_c \; X_c + D_c \begin{bmatrix} \text{e\_zeng} \\ \text{N2} \\ \text{e\_ueng} \end{bmatrix}$$

where $\text{e\_ueng} = \begin{bmatrix} \text{e\_ETA} \\ \text{e\_WF36} \end{bmatrix}$ is the vector of actuator errors.

With zero actuator error, the IWP gains do not contribute to the state derivative calculations and hence the CAW scheme is memoryless.

4.5 MAW Scheme for Engine Control System Model

The implementation of the MAW scheme in the engine controller is shown in Figure 4.7, where the blocks 3 and 6 implement the MAW scheme in the nominal control system as shown. During the optimization stage, a value for α (windup factor) is assumed and the design variable β is obtained for the best system performance. With the MAW scheme implemented, the controller equations can be given as:

$$\dot{X}_C = \left(A_C + \beta\ (\alpha - 1)I\right)X_C + B_C\begin{bmatrix} e\_zeng \\ N2 \end{bmatrix}$$

$$Y_C = C_C X_C + \alpha D_C\begin{bmatrix} e\_zeng \\ N2 \end{bmatrix}$$

where e_zeng is the vector of performance errors. When the actuators are not limited, $\alpha=1$, and we get back the nominal control system. Thus, the MAW scheme is also memoryless. The system matrices of the controller, [A_c, B_c, C_c, D_c] that appear in Figure 4.7 are listed in the appendix.

4.6 Optimization of the Performance Index

The algorithm to optimize the 2-norm (rms norm) error of the IWP optimization structure is presented in Figure 4.8. First, we ensure that the chosen set of gains stabilize the controller and the closed loop IWP system. If the IWP gains form a stable system, the 2-norm of the system is obtained from "*rms*" function available in MatrixX software. The IWP gains that yield the minimum 2-norm are obtained using the stochastic GA optimization technique described in chapter 3.

The algorithm to optimize the performance index of the CAW system is given in Figure 4.9, where the structured singular value (μ) of the IWP system is minimized with a constraint on the 2-norm. The structured singular value (ssv) of the IWP system is obtained from "*ssv*" function available in the MatrixX software. A 10% bound for the diagonal uncertainties and a frequency range (0.001-10) Hz are assumed for the μ analysis. 'k' is the factor (>1) that determines the 2-norm performance degradation we are willing to tolerate in order to improve the structured singular value of the system. In this study, 'k' is assumed to be 1.1, which means that we allow a 10% degradation in the 2-norm performance to minimize directional sensitivity of the IWP system. First few steps of this algorithm ensure that the IWP gains form a stable controller and a stable closed loop IWP system. If a set of gains do not yield a stable solution, or if they do not satisfy the 2-norm constraint, a high penalty is imposed on them. Thus, the parameters that satisfy the rms norm constraint and minimize the structured singular values of the system are obtained as the gains for the CAW scheme. In this algorithm, J_{2min} is the minimum 2-norm of the IWP optimization structure given in Figure 4.3.

The algorithm to compute the performance index for the MAW scheme is similar to that shown in Figure 4.8. In the MAW approach, we need not include the structured singular value computations. This is because, the MAW method preserves the direction of the control vector and hence there is no directional uncertainty. Thus the performance measure for the MAW scheme is just the 2-norm of the optimization structure presented in Figure 4.3. Also in the MAW approach, we have just one design variable (β) to be determined for optimum performance. The rms performance measure was found to monotonically decrease with β, indicating that a high value of β is ideal for good performance. However, with high beta, the system bandwidth increases and also, it may not be practically feasible to implement a high β. Hence, from these considerations the value for β was chosen to be 25 for this study.

4.7 IWP implementation Results for the Engine Controller Problem

In this section, we present the results of the application of IWP techniques for the engine controller of a STOVL aircraft and compare the system performance obtained using the CAW and the MAW schemes.

4.7.1 Nominal and limited system response

During simulation of the engine control system, a FGE step input command of 1000 lbf and a N2 step input command of 100 rpm were assumed. The nominal system is the linear control system without any actuator limits. For the limited control system, fuel flow to the engine and the ejector butterfly valve angle were assumed to be limited. The limits on ETA and WF36 were set at 7.0 degrees and 450 lbm/hr respectively.

The FGE command tracking response of the nominal and the limited system (without IWP) is presented in Figure 4.10. The effects of integrator windup of the limited system are clearly seen in this figure. The limited system exhibits oscillations, high steady state error, and a degraded performance when the step command is terminated after 5 seconds. The degraded performance following the termination of the step command is because, the actuators of the limited system do not get off the limits until the integrators unwind. The ejector angle commanded by the controller to track the FGE step command is given in Figure 4.11. As seen from this figure, the nominal control system requires an ejector angle of about 9 degrees to track the FGE step command. When the ejector angle is limited at 7 degrees, the controller's integrators windup resulting in poor tracking of the limited actuator.

The N2 command tracking response of the nominal and the limited system (without IWP) is presented in Figure 4.12. Again, the limited system exhibits increased oscillations and sluggish behavior (increased rise and settling times). The fuel flow commanded by the nominal and the limited controllers to track the N2 command is presented in Figure 4.13. The nominal system requires about 570 lbm/hr of fuel to track the N2 command. When the fuel flow is limited at 450 lbm/hr, the windup causes poor actuator tracking as shown.

4.7.2 CAW scheme for Engine Control System - Results

The CAW gains were obtained by first minimizing the 2-norm of the optimization structure (J_{2min}), and then minimizing the structured singular value of the IWP system with a constraint on the 2-norm in terms of J_{2min}. Structured singular value analysis for the CAW gains is presented in Figure 4.14. Since the maximum μ obtained in Figure 4.14 is less than unity, the CAW system is robustly stable for changes in control vector direction.

When the CAW scheme is implemented to provide IWP, the FGE command tracking response of the CAW system matches closely with that of the nominal system as shown in Figure 4.10. The CAW scheme eliminates the oscillations and the high steady state error exhibited by the unprotected system. When the step command terminates, the system with IWP gets off the limits quickly, and tracks the FGE almost exactly as the nominal system. Also, when IWP is implemented, an accurate tracking of the limited actuator is achieved. This is shown in Figure 4.11, where the ETA commanded by the controller is at the limit (7 degrees).

The command tracking response of the CAW system for the N2 command is shown in Figure 4.12. The CAW system exhibits reduced oscillations, a smaller steady state error and is less sluggish than the unprotected system. The IWP controller tracks limited WF36 accurately as seen in Figure 4.13. We can also observe that the IWP controller demands a higher fuel flow than the nominal system during the time period (2-5) seconds. This is due to the limit on the ejector angle, ETA. The control variables of this system are all coupled and hence, when one actuator is limited, the controller attempts to track the commands with an increased value for other variables. Hence, when the ejector angle is limited, the control system attempts to track the FGE command with an increased fuel flow to the engine.

4.7.3 MAW scheme for Engine Control System - Results

The results of the implementation of the MAW scheme for the engine controller are presented in Figures 4.10 and 4.12. Figure 4.10 shows the FGE command tracking response of the MAW system. Figure 4.12 shows the N2 command tracking response of the IWP system with MAW scheme. We can observe from these figures that with MAW scheme, the system gets off the limits quickly and provides a response close to the nominal system when the step command is terminated. However, MAW scheme increases the rise time and results in a high steady state error for this system as seen in Figures 4.10 and 4.12. Figures 4.11 and 4.13 indicate that with MAW scheme, an accurate actuator tracking is attained when the system encounters the operational limits.

4.7.4 Comparison between CAW and MAW schemes

In Figures 4.15 and 4.16, the steady state error variations on FGE and N2 commands with ETA and WF36 limits are presented. Figure 4.15 shows the variation of the steady state error on the commanded FGE as the limit on ejector angle is increased from 1 degree to 6 degrees. For this case, input to the system is a FGE step command of 1000 lbf, and a limit of 450 lbm/hr is set for fuel flow. The CAW system results in a high steady state error when stringent limits are imposed on ETA. As the limit on ETA is relaxed, the error reduces initially but stays constant for ETA greater than 4 degrees. The error does not decrease further due to the limit on fuel flow. For a given fuel flow rate to the engine, there is a maximum ejector thrust that could be generated. As the ejector valve angle is increased from zero, the ejector thrust increases until the maximum value is attained. Once the maximum is attained, FGE can be increased further only by increasing the fuel flow to the engine. Without the limits on fuel flow, the control system tends to track the FGE command accurately for any limited ETA (>0) by appropriately increasing the fuel flow to the engine. Figure 4.15 shows that the steady state error is higher with the MAW scheme for any ETA limit.

Figure 4.16 shows the variation of the steady state error on N2 command as the limit on WF36 is varied between 200 lbm/hr to 600 lbm/hr. For this case, a N2 step command of 100 rpm is the input and ETA limit is set at 7.0 degrees. As seen in the figure, MAW system results in a higher steady state error than the CAW system. With

CAW scheme, steady state error approaches zero as the limit on fuel flow is relaxed. However, with MAW scheme the errors attain a constant value and do not approach zero even if the limits on fuel flow are removed. We can conclude from these results that for the engine controller problem, with the fuel flow and the ejector valve being limited, the CAW scheme provides better anti-windup protection than the MAW scheme.

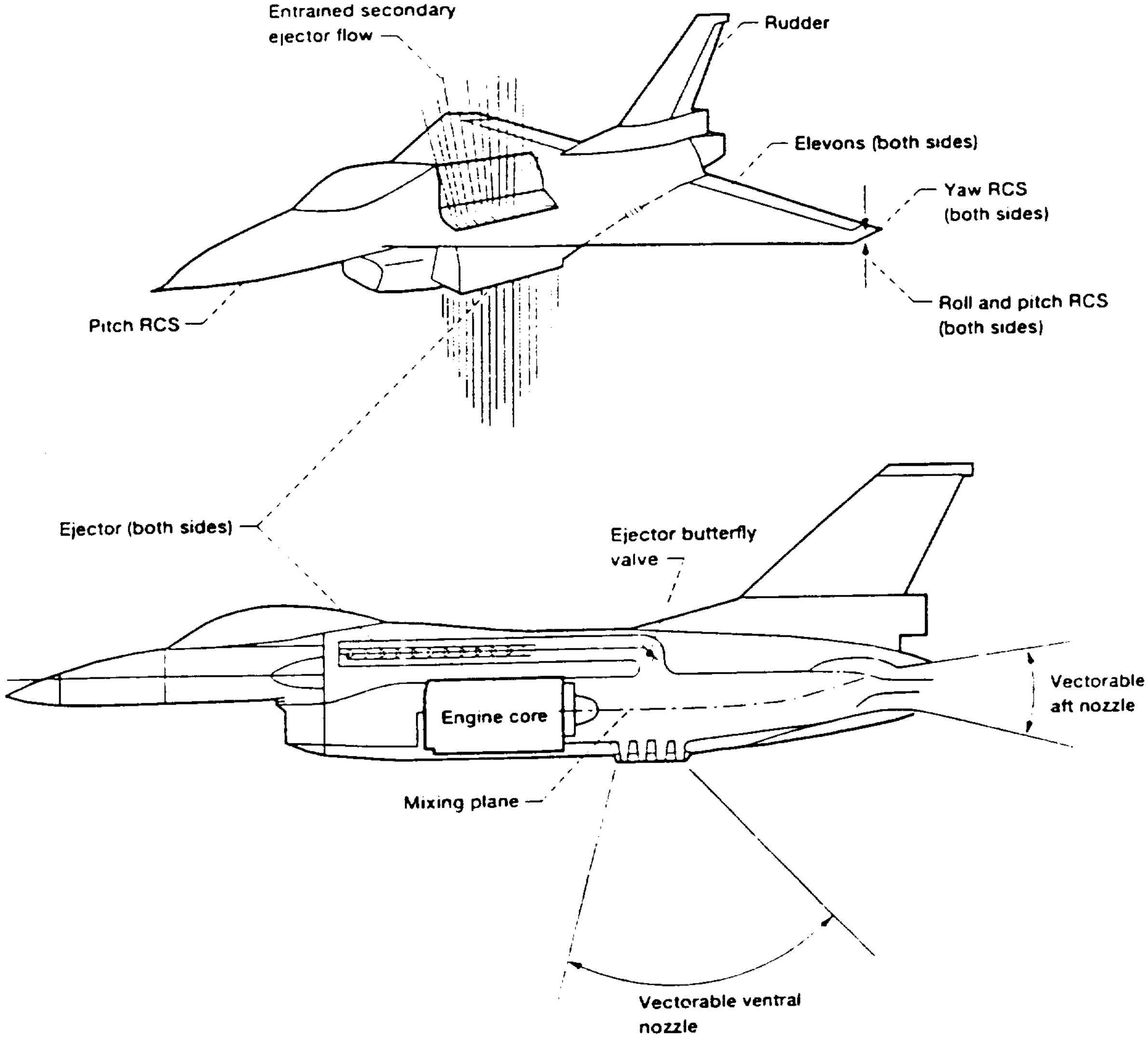

Figure 4.1 Control Actuators of a STOVL aircraft.

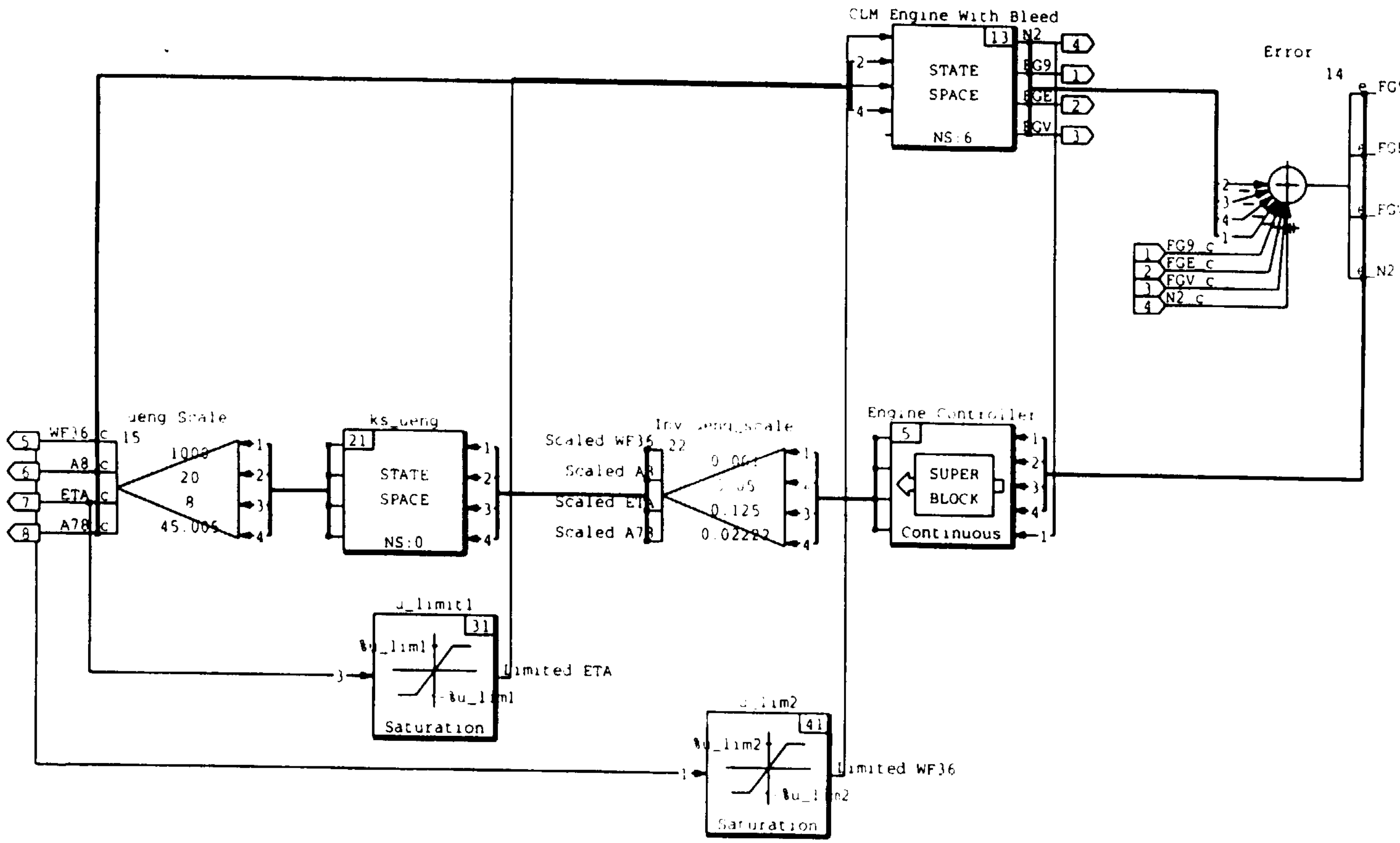

Figure 4.2 Limited Engine System.

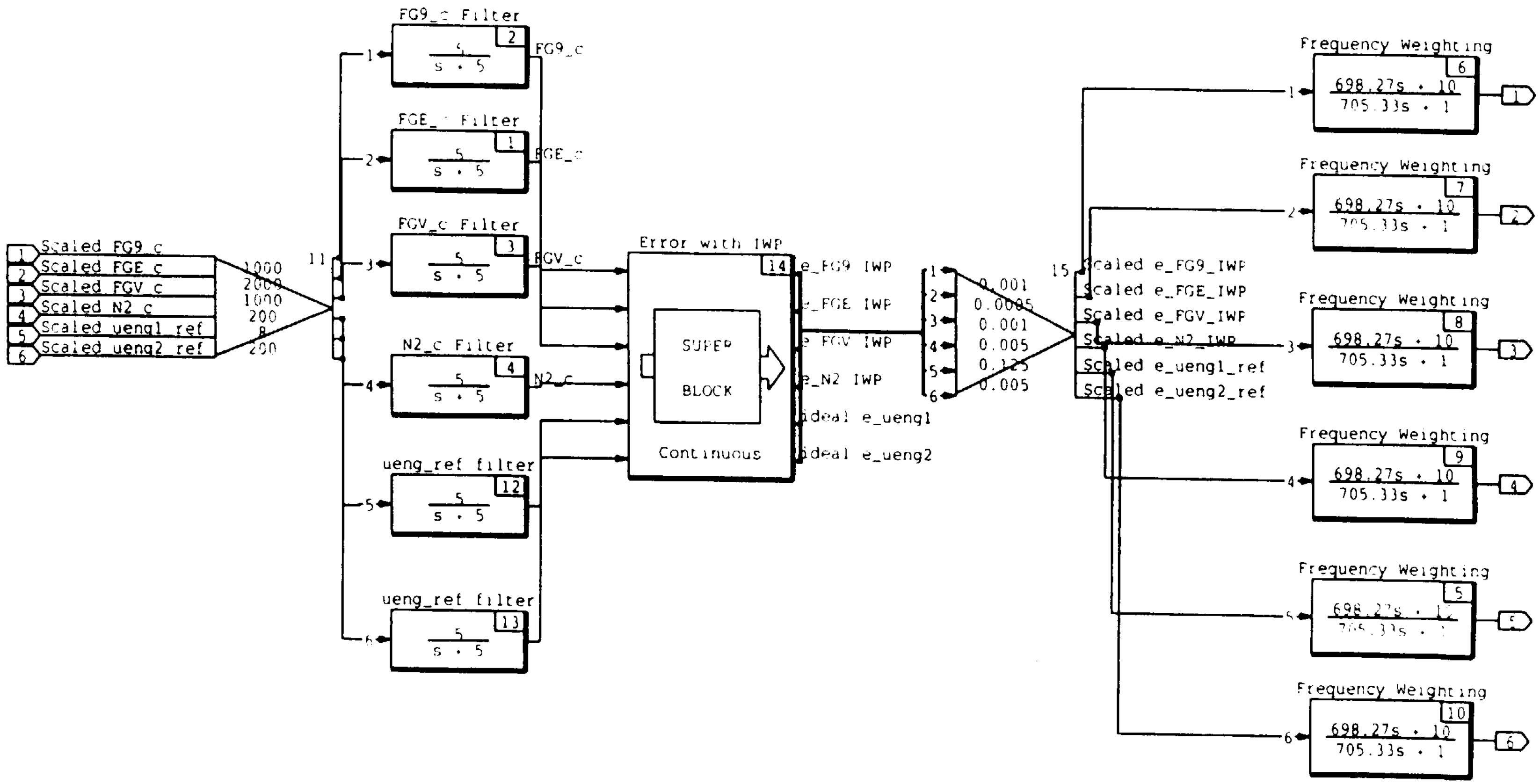

Figure 4.3 IWP Optimization Structure.

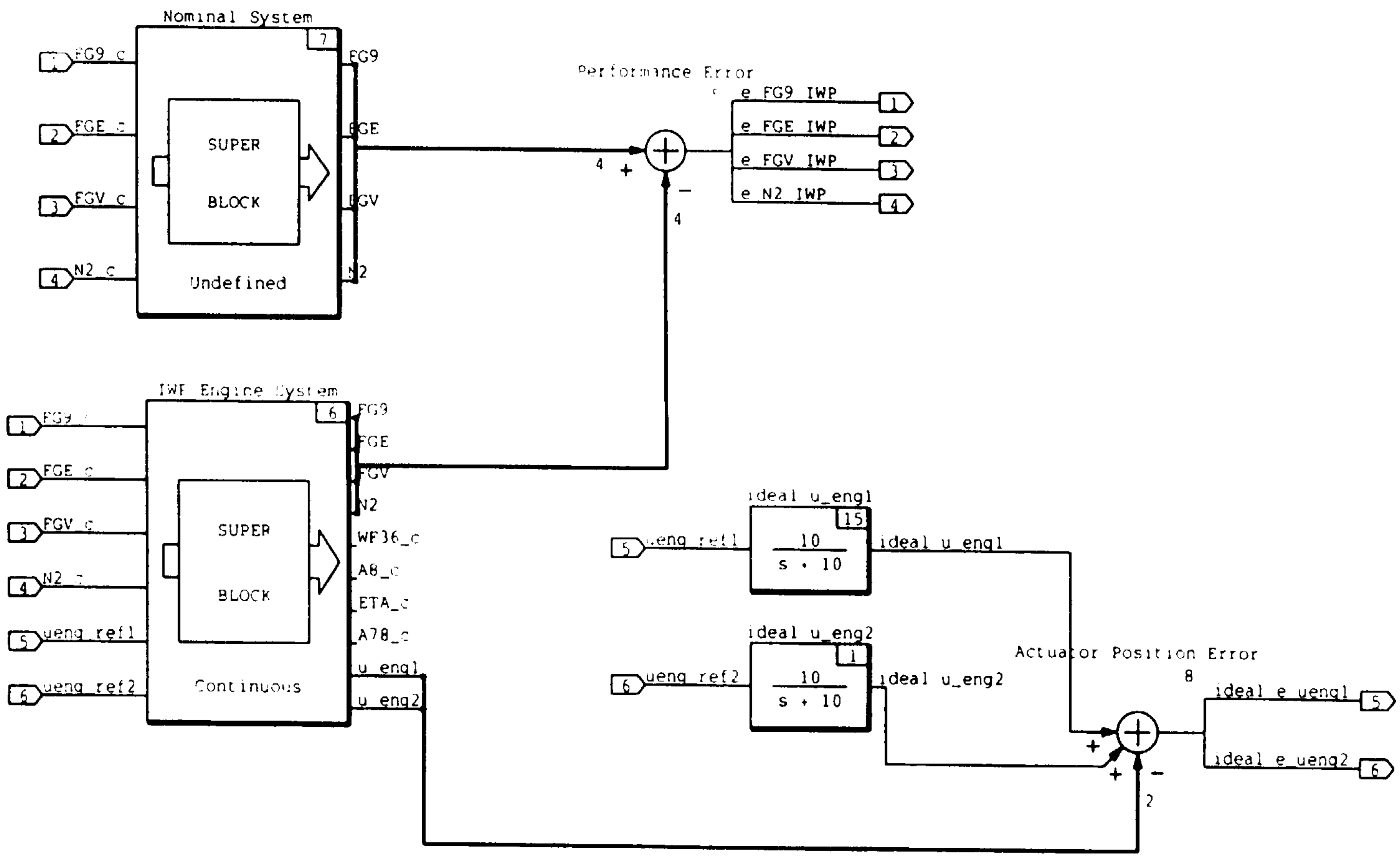

Figure 4.4 Error with IWP System.

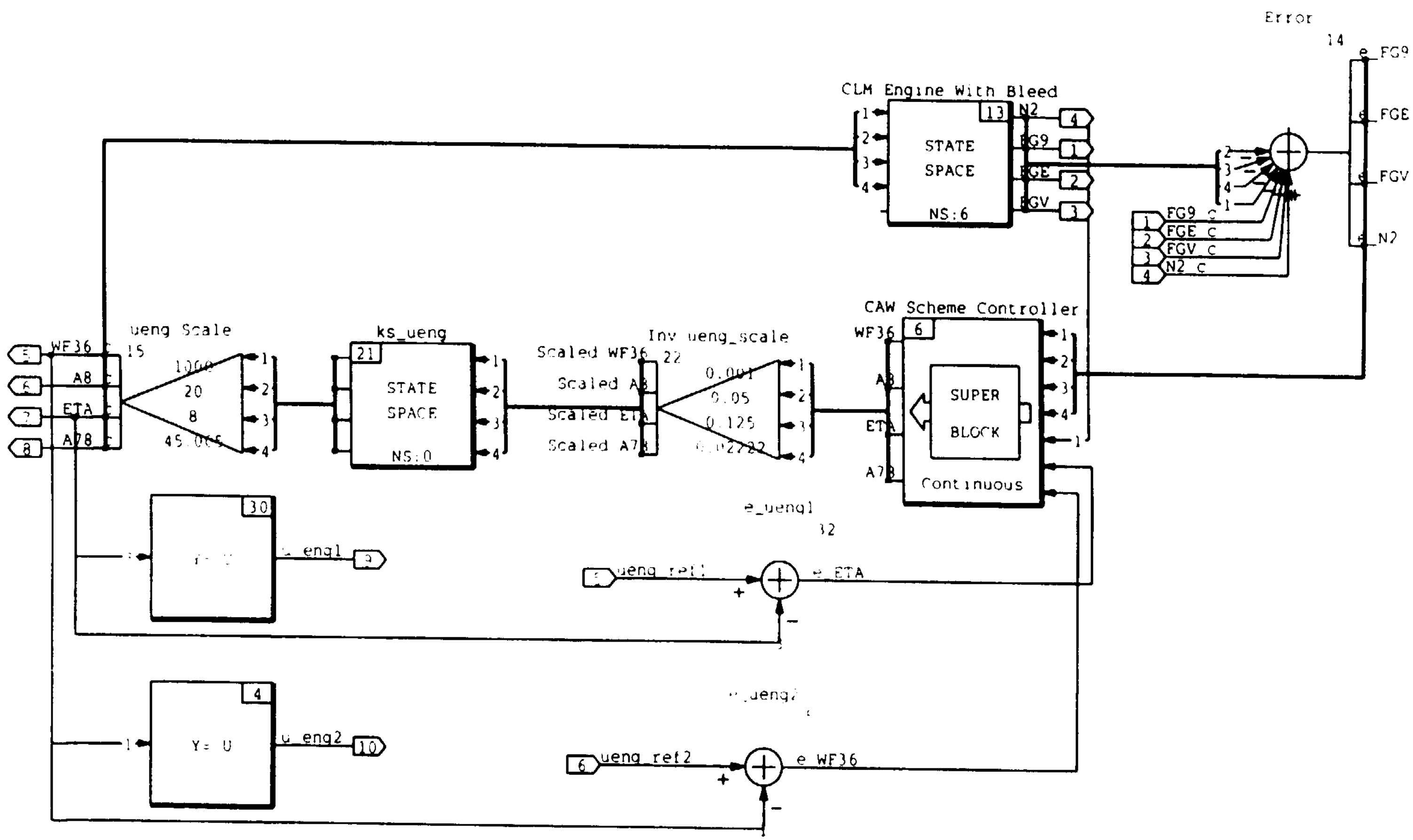

Figure 4.5 IWP Closed Loop System.

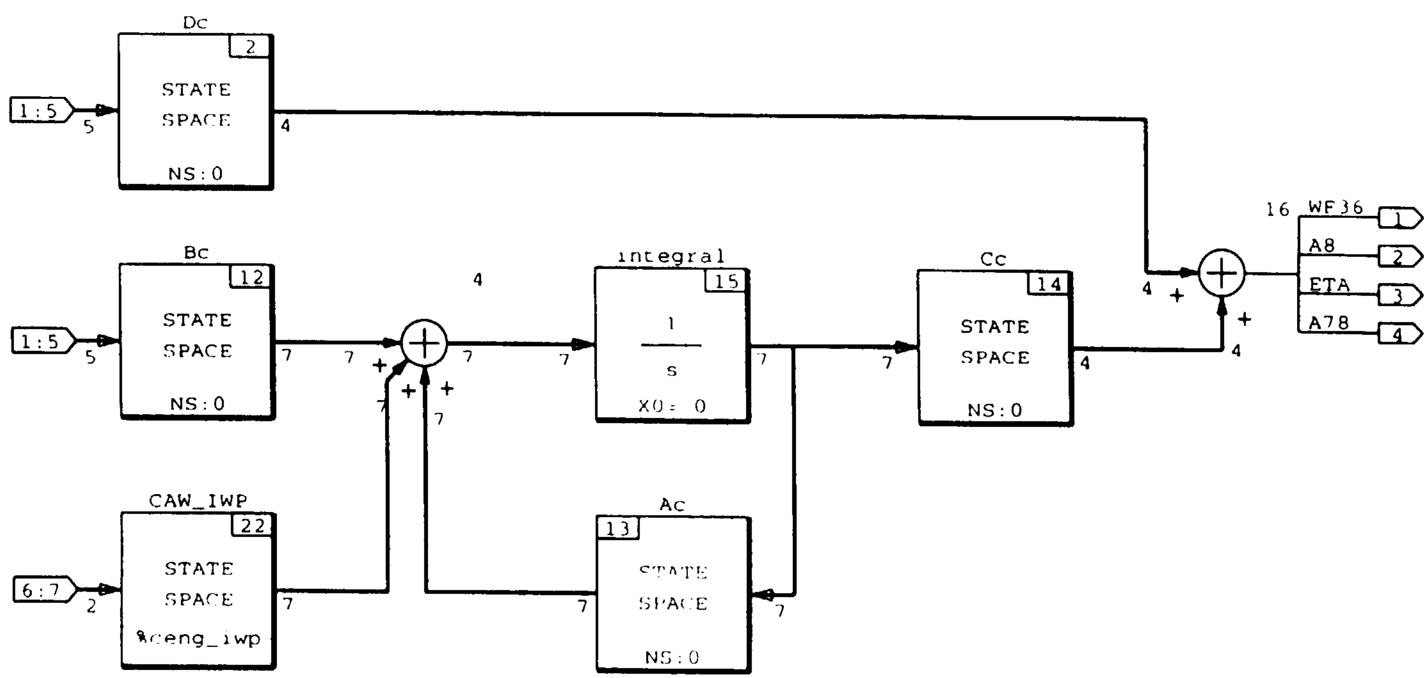

Figure 4.6 CAW Scheme Controller.

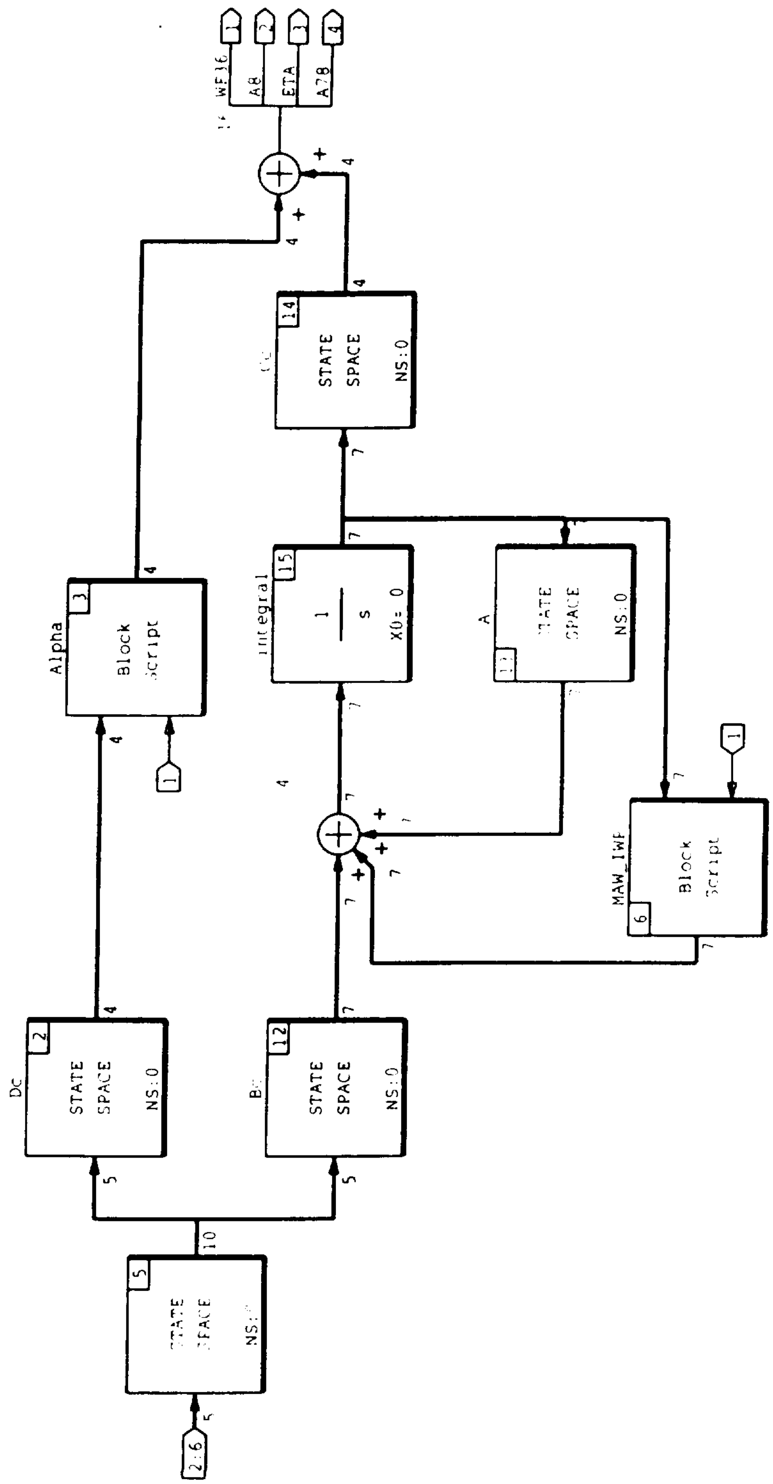

Figure 4.7 MAW Scheme Controller.

While (Solution not converged)

caw_iwp ← IWP gains

If (*engine controller* and *closed loop system*) are stable

J_2 ← **rms**("IWP Optimization Structure")

else

J_2 ← high penalty

end

endwhile

Figure 4.8 Algorithm for RMS norm optimization. "rms" is the function that returns the 2-norm of the system.

J_{2min} ← minimum rms. norm of "IWP optimization structure".

While (Solution not converged)

caw_iwp ← IWP gains chosen

If (*engine controller* and *closed loop system*) are stable

J_2 ← **rms**("IWP Optimization Structure")

if ($J_2 < k^*J_{2min}$)

μ ← ssv("IWP Engine System")

else

μ ← high penalty

end

else

μ ← high penalty

end

endwhile

Figure 4.9 Algorithm for CAW gain optimization. "rms" returns the 2-norm of the system passed to the function, and "ssv" returns the maximum structured singular value of the system passed to the function.

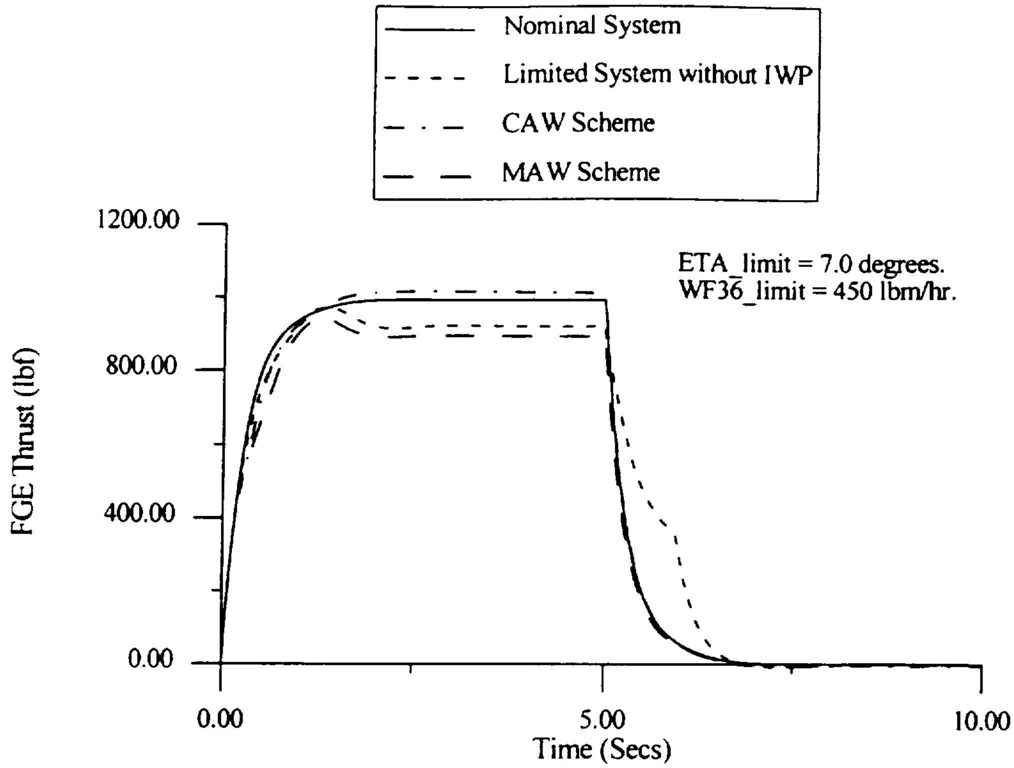

Figure 4.10 FGE command tracking response for nominal, limited (without IWP), CAW and MAW Control Systems.

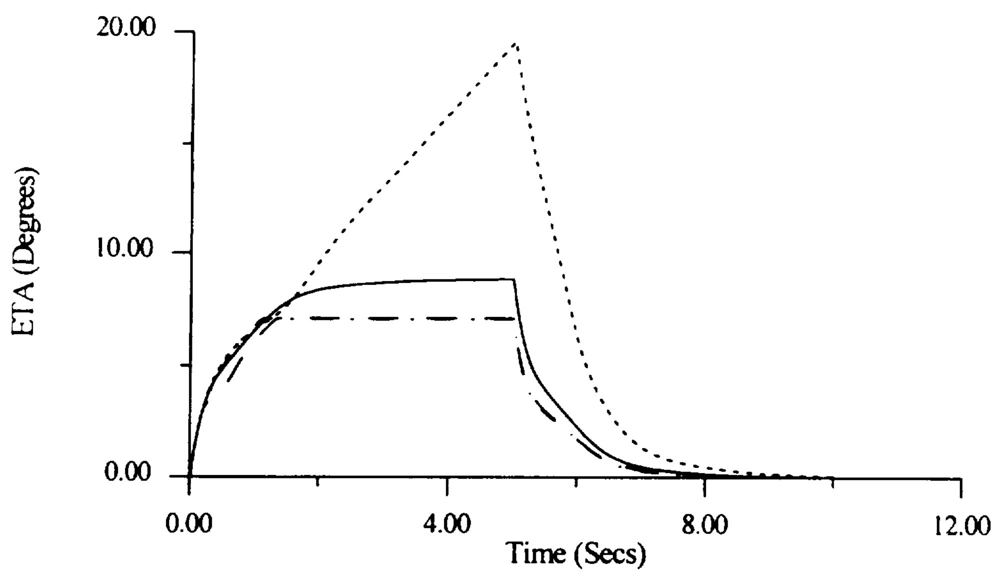

Figure 4.11 Ejector Angle (ETA) commanded by the nominal, limited, CAW and MAW controllers. The limit on ETA is 7 degrees.

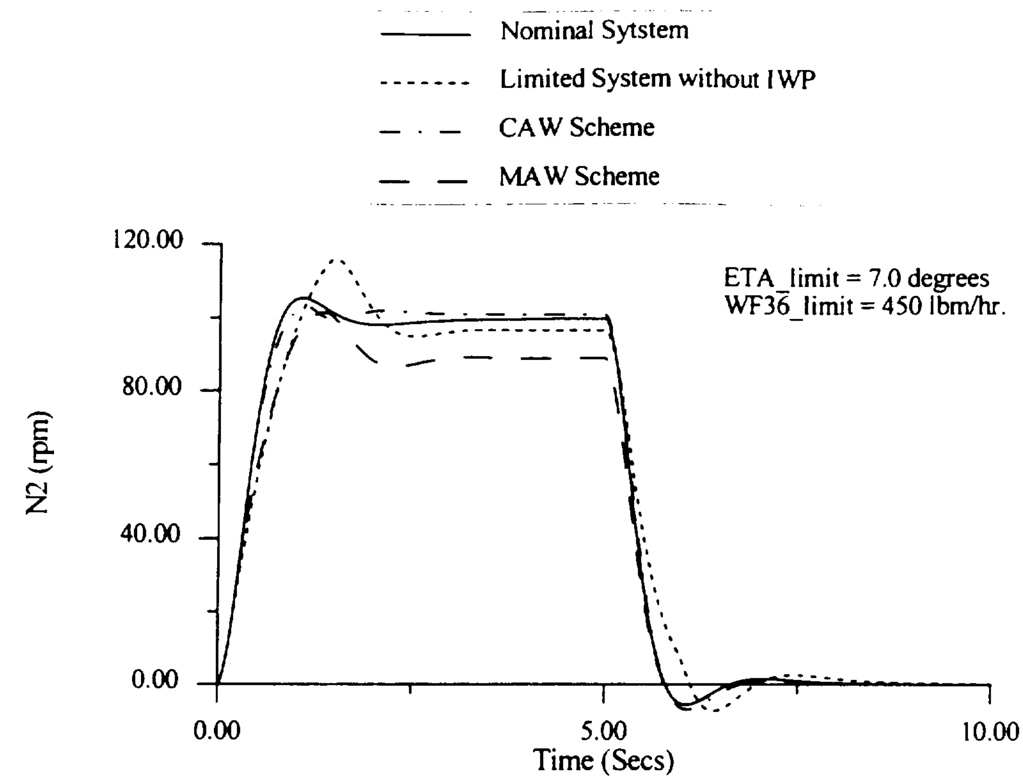

Figure 4.12 Nominal, limited system (without IWP), CAW and MAW Control System responses for a N2 step command of 100 rpm.

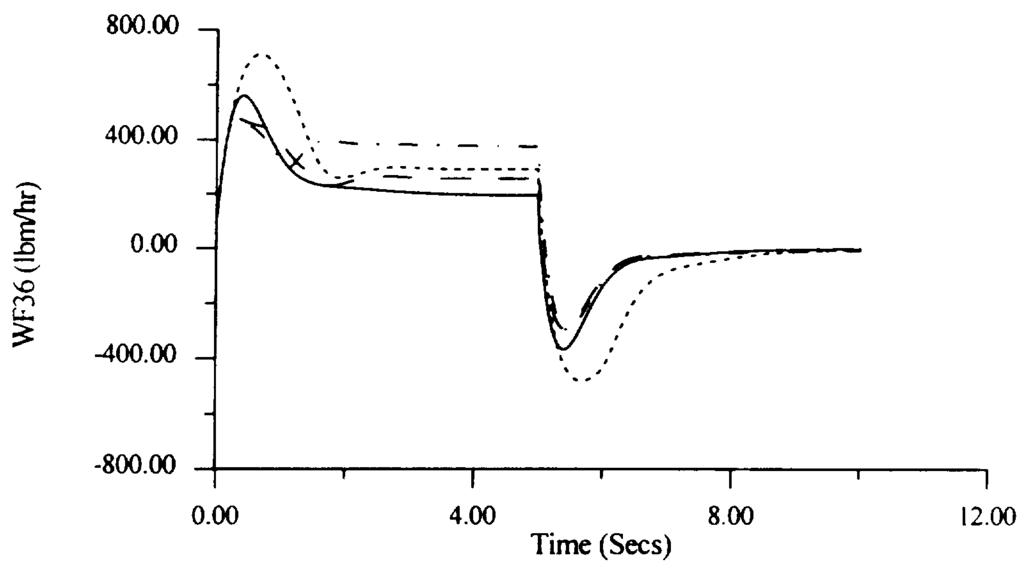

Figure 4.13 Fuel flow (WF36) commanded by the nominal, limited, CAW and MAW controllers. The limit on WF36 is 450 lbm/hr.

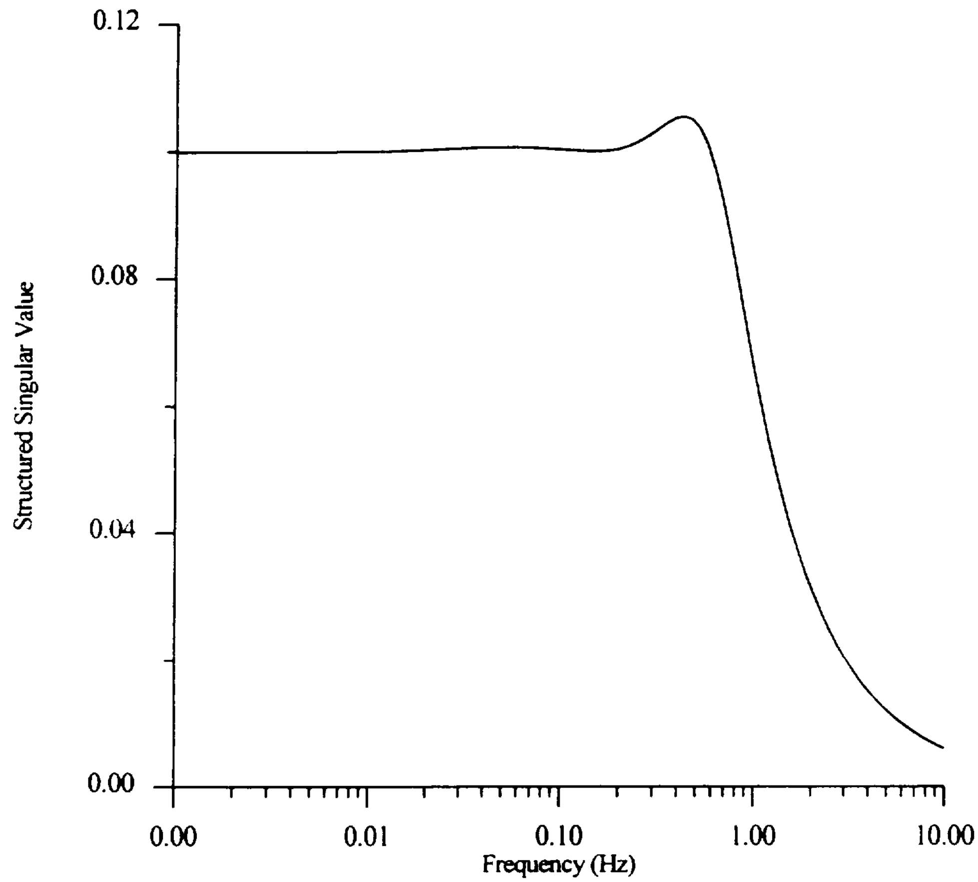

Figure 4.14 Structured Singular Value of the limited IWP system for uncertain actuator limits.

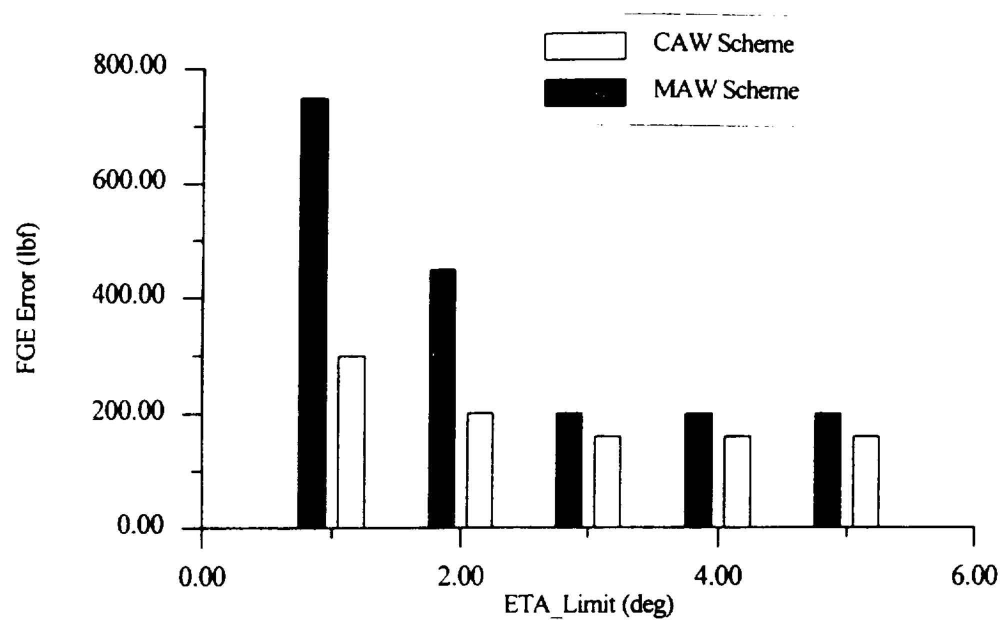

Figure 4.15 Steady state error on FGE step command of 1000 lbf with varying ETA limits. Fuel flow (WF36) is limited at 450 lbm/hr.

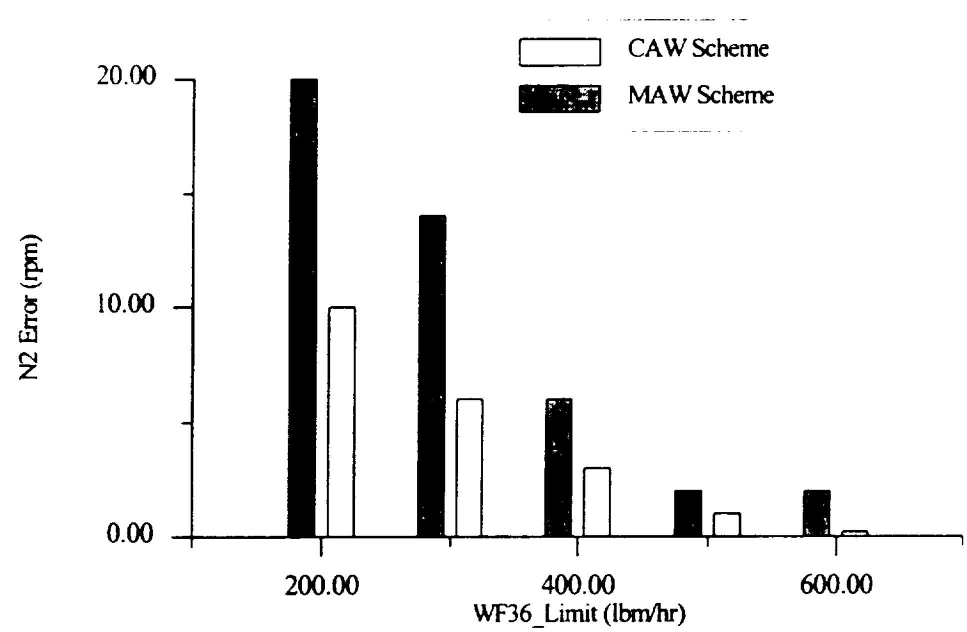

Figure 4.16 Steady state error on N2 step command of 100 rpm with varying limits on fuel flow. ETA limit is set at 7.0 degrees.

Chapter 5

Conclusions and Recommendations

Two anti-windup compensation techniques based on the back-calculation principle, namely the Conventional Anti-Windup (CAW) and the Modified Anti-Windup (MAW) techniques were described and implemented for three problems. An unified performance index has been defined for the CAW scheme such that the directional sensitivity of the CAW system is minimized while providing an acceptable performance. Since the MAW scheme maintains the control direction, we just consider the 2-norm performance measure to obtain an optimal performance for the MAW system. The results obtained from the IWP implementations for these problems are summarized below.

In the first illustrative example (System A) considered in chapter 2, the limited system was robustly stable to changes in the control direction. Hence the CAW scheme could provide a stable IWP controller, but the performance of the CAW system was deteriorated. This could be due to the sensitivity of system performance to the control direction. Since the MAW approach maintains the direction of the control vector, a stable system with an acceptable performance was obtained with the MAW system.

In the second example (System B) of chapter 2, the limited system was not robustly stable for changes in control vector direction caused due to actuator limits. Hence it was not possible to obtain a CAW controller that stabilized the limited system. The MAW scheme provided a stable controller with an acceptable performance.

The limited engine controller system considered in chapter 4 was robustly stable for uncertain actuator limits, and the simulation results indicate that CAW system provides a performance better than the MAW scheme. Since in the MAW scheme, the design variable β is the only choice to obtain an optimal performance, it is not always possible to tune the system to a desirable performance. Based on these results, we can draw the following conclusions :

- The directional sensitivity of the IWP system must be included in the performance index to minimize the system sensitivity to control direction changes caused when actuators encounter operational limits.
- If the limited system is robustly stable to diagonal uncertainties, then CAW approach would provide a stable IWP system. The CAW scheme would perform better than the MAW scheme if the performance of the system is not very sensitive to the control direction. However, if the performance is sensitive to the control direction, then MAW scheme would perform better.

If the CAW system is not robustly stable to control direction changes, the MAW scheme could provide adequate windup protection.

The following recommendations are made based on the results obtained in this study. Given a linear control system for which a IWP technique is to be implemented to satisfy the general IWP requirements, we first attempt to implement the CAW technique for the system. If it is possible to obtain a stable closed loop IWP system with the CAW scheme, we optimize the 2-norm performance of the CAW system after including appropriate loop shaping and frequency weighting blocks in the performance definition structure. Once the set of CAW gains that minimize the 2-norm of the IWP system is obtained, we can minimize its directional sensitivity by minimizing the structured singular value of the IWP system with a constraint on the 2-norm. The CAW gains thus obtained will minimize the directional sensitivity of the closed loop system while ensuring an acceptable performance.

If a set of CAW gains that stabilize the limited system cannot be obtained, the limited system is not robustly stable to control vector direction changes. Hence we must resort to the MAW scheme to maintain the control direction when the actuator encounters the operational limits.

As a general conclusion, if the CAW scheme optimized with the unified performance measure shows robust stability, it is preferred over the MAW scheme.

References

[1] Mattern Duane. "A Comparison of two multi-variable Integrator Windup Protection schemes", *AIAA Guidance, Navigation and Control Conference Proceedings*, Monterey, CA, August 1993.

[2] Watts, S.R., Garg, Sanjay. "An Optimized Integrator Windup Protection Technique Applied to A Turbofan Engine", *AIAA Guidance, Navigation and Control Conference Proceedings*, AIAA-96-3814, San Diego, CA, July 1996.

[3] Astrom, Karl., Rundqwist, Lars. "Integrator Windup and How to Avoid it", *Proceedings of the 1989 American Control Conference*, Pittsburgh, PA, pp.1693-1698, 1989.

[4] Campo, P.J., Morari, M., Nett, C. N. "Multivariable Anti-Windup and Bumpless Transfer : A General Theory", *Proceedings of the 1989 American Control Conference*, Pittsburgh, PA, TP9-5.00, pp.1706-1711, 1989.

[5] Peng, Youbin., Vrancic, Damir., Hanus, Raymond. "Anti-Windup, Bumpless, and Conditioned Transfer Techniques for PID Controllers", *IEEE Control Systems Magazine*, pp.48-56, August, 1996.

[6] Doyle, John. C., Smith, Roy. S., Enns, Dale. F. "Control of Plants with Input Saturation Nonlinearities", *Proceedings of the 1987 American Control Conference*, Minneapolis, MN, TA8-10:30, pp.1034-1039, 1987.

[7] KrishnaKumar, K., Goldberg, D.E. "Genetic Algorithms in Control System Optimization", *AIAA Journal of Guidance, Control and Dynamics*, 15(3), May-June 1992, pp.735-740.

[8] Swaminathan, R. "Controller Partitioning Optimization for a STOVL Aircraft using Genetic Search Methods", Masters Thesis, The University of Alabama.

[9] KrishnaKumar, K., Narayanaswamy, S., Garg, Sanjay. "Solving Large Parameter Optimization Problems Using A Genetic Algorithm with Stochastic Coding", *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons, pp.287-301, 1995.

[10]MatrixX, "Xmath Optimization Module", Integrated Systems, Inc., Santa Clara, CA, June 1994.

[11]Wittenmark, Bjorn. "Integrators, Nonlinearities, and Anti-reset Windup for Different Control Structures", *Proceedings of the 1989 American Control Conference*, Pittsburgh, PA, pp.1679-1683, 1989.

[12]Raymond, Hanus., Kinnaert, Michael. "Control of Constrained Multivariable Systems using the Conditioning Technique", *Proceedings of the American Control Conference*, Pittsburgh, PA, TP9-5:30,pp.1712-1718, 1989.

[13]Yang, Sangsik., Leu, Ming. C., "Stability and Performance of a Control System with an Intelligent Limiter", *Proceedings of the 1989 American Control Conference*, Pittsburgh, PA, TP9-4:30, pp.1699-1705, 1989.

[14]Radhakisan, Sohanlal, Baheti. "Simple Anti-Windup Controllers", *Proceedings of the 1989 American Control Conference*, Pittsburgh, PA, pp.1684-1686, 1989.

[15]Doyle, John., Glover, Keith., Smith, Roy. "Xmath X_μ module", Integrated Systems, Inc., Santa Clara, CA, June 1994.

[16]Whitley, D., Beveridge, R., Graves, C., Mathias, K., "Test Driving Three 1995 Genetic Algorithms: New Test Functions and Geometric Matching", unpublished paper, Colorado State University, Co, 1996.

[17]Goldberg, D.E., "Genetic Algorithms in Search Optimization and Machine Learning", *Addison-Wesley*, Reading, MA, 1989.

[18]Garg, Sanjay. "Partitioning of Centralized Integrated Flight/Propulsion Control Design for Decentralized Implementation", *IEEE Transactions of Control Systems Technology*, Vol. 1, No. 2, June 1993.

[19]Garg, Sanjay., Ouzts, Peter. J., Lorenzo, Carl. F. "IMPAC- An Integrated Methodology for Propulsion and Airframe Control", NASA TM 103805.

[20]Mattern, Duane., Garg, Sanjay. "Propulsion System Performance Resulting From an Integrated Flight/Propulsion Control Design", NASA TM 105874.

Appendix

The system matrices for the Engine are given by :

$$A_e = \begin{bmatrix} -4.700010e+00 & 3.018767e+00 & 4.967368e-01 & 1.602869e-01 & \cdots \\ 4.065682e-01 & -3.244523e+00 & 6.536386e-01 & 2.265022e-01 & \cdots \\ 1.175501e-02 & 7.358288e-03 & -1.958345e-01 & 3.970851e-04 & \cdots \\ 5.816997e-04 & -5.115731e-03 & 1.058630e-02 & -6.892528e-02 & \cdots \\ 1.873175e-04 & -3.088027e-02 & 2.000850e-02 & 6.505967e-03 & \cdots \\ 2.103332e-03 & -1.236427e-02 & 6.840893e-03 & 1.851961e-03 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} 2.827156e-01 & -1.578444e-01 \\ -6.877085e-02 & 1.892776e-03 \\ 2.165923e-04 & 3.248888e-04 \\ 1.098345e-04 & 9.293700e-05 \\ -1.482563e-01 & 3.452896e-04 \\ 4.816788e-03 & -6.845138e-02 \end{bmatrix}$$

$$B_e = \begin{bmatrix} 4.116053e-01 & 1.589558e+01 & 2.808532e+01 & 1.517464e+01 & -8.390485e+01 \\ 7.904529e-01 & -1.158799e+00 & -2.386159e+00 & -1.104540e+00 & -7.945873e+01 \\ 1.278608e-03 & -4.016956e-02 & -7.278302e-02 & -3.834760e-02 & -4.053042e-01 \\ 7.185900e-03 & -1.079511e-02 & -2.109056e-02 & -1.032735e-02 & 2.895667e-01 \\ 2.420218e-02 & -3.411469e-02 & -6.670258e-02 & -3.263581e-02 & 1.256648e+00 \\ 7.719461e-03 & -3.368659e-02 & -6.136922e-02 & -3.219078e-02 & 3.720236e-01 \end{bmatrix}$$

$$C_e = \begin{bmatrix} 1.000000e+00 & -1.425019e-17 & -2.003546e-22 & -1.122547e-22 \ldots \\ 1.642651e+00 & 4.554199e-01 & -4.125977e-02 & -2.856445e-02 \ldots \\ 1.519971e+00 & 4.229004e-01 & -3.417969e-02 & -2.563477e-02 \ldots \\ 5.802856e-01 & 1.606079e-01 & -1.379395e-02 & -9.948730e-03 \ldots \end{bmatrix}$$

$$\begin{bmatrix} -8.348264e-24 & 1.385719e-23 \\ 1.586914e-02 & 8.911133e-02 \\ 1.757813e-02 & 8.666992e-02 \\ 6.225586e-03 & 3.192139e-02 \end{bmatrix}$$

$$D_e = \begin{bmatrix} -8.938996e-19 & 3.256075e-20 & 1.049588e-18 & 7.984864e-21 & 0.000000e+00 \\ 1.199692e-01 & 9.158529e+00 & -2.332959e+01 & -1.277783e+01 & -2.366211e+01 \\ 1.140380e-01 & -1.245736e+01 & 8.738379e+01 & -1.189714e+01 & -2.202637e+01 \\ 4.239934e-02 & -4.720947e+00 & -8.239990e+00 & 1.392403e+01 & -8.377686e+00 \end{bmatrix}$$

The system matrices for the Engine Controller are given by :

$$A_c = \begin{bmatrix} -1.086374e-02 & 4.105142e-05 & -6.839382e-06 & 0.000000e+00 \ldots \\ 4.886350e-05 & -1.073942e-02 & 7.689710e-06 & 0.000000e+00 \ldots \\ -1.365098e-05 & 1.021952e-05 & -1.071602e-02 & 0.000000e+00 \ldots \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -7.251576e-03 \ldots \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -3.671925e-03 \ldots \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -6.453421e-03 \ldots \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 6.830768e-03 \ldots \end{bmatrix}$$

$$\begin{bmatrix} 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 0.000000e+00 & 0.000000e+00 & 0.000000e+00 \\ 3.413143e-04 & 1.879533e-03 & 3.298592e-04 \\ -1.451197e-02 & -3.628540e-02 & -6.237118e-02 \\ -3.853371e-02 & -1.295285e-01 & -3.114797e-01 \\ 5.258483e-02 & 2.589395e-01 & -2.926340e+00 \end{bmatrix}$$

$$
B_c = \begin{bmatrix}
4.930231e-04 & 7.559844e-04 & 2.145138e-03 & 0.000000e+00 & 0.000000e+00 \\
-2.493989e-03 & 3.370512e-04 & 8.464059e-05 & 0.000000e+00 & 0.000000e+00 \\
4.544243e-04 & 9.250075e-04 & -1.405475e-03 & 0.000000e+00 & 0.000000e+00 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -1.014878e-02 & 4.633683e-06 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -2.561923e-03 & -2.081017e-04 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -4.501312e-03 & -7.602718e-04 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 4.778705e-03 & -2.599835e-03
\end{bmatrix}
$$

$$
C_c = \begin{bmatrix}
1.912883e+03 & -6.844580e+02 & 1.336771e+02 & -1.668799e+03 \ldots \\
-2.965411e+01 & -3.751226e+01 & 1.278786e+01 & -6.990420e+00 \ldots \\
-2.331950e+00 & 1.018938e+01 & 1.564451e+01 & -5.892188e+00 \ldots \\
-4.905758e+01 & 4.661629e+01 & -5.235811e+01 & -3.684842e+01 \ldots
\end{bmatrix}
$$

$$
\begin{matrix}
1.606993e+02 & 4.951522e+02 & 2.779248e+02 \\
6.471627e+00 & -9.979779e+00 & -8.766341e+00 \\
1.673976e+00 & 2.719421e+00 & 4.858876e+00 \\
-1.350391e+01 & -2.139078e+01 & -3.642707e+00
\end{matrix} \Bigg]
$$

$$
D_c = \begin{bmatrix}
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 1.242878e+00 & -3.416672e+00 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & 3.504186e-05 & -3.268677e-02 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -9.807313e-04 & -2.097606e-02 \\
0.000000e+00 & 0.000000e+00 & 0.000000e+00 & -4.619351e-03 & -9.870098e-02
\end{bmatrix}
$$

The scheduling gains for the 100 knot engine model to extend the 80 knot nominal design controller is given by the matrix :

$$
K = \begin{bmatrix}
9.118556e-01 & -1.847999e-01 & 6.108760e-02 & 1.466123e-01 \\
-4.999717e-01 & 2.278030e+00 & 4.999329e-01 & 4.999664e-01 \\
-1.461968e-01 & 2.113056e-01 & 1.250743e+00 & -1.722879e-03 \\
1.740255e-01 & -1.604945e-01 & -1.629085e-01 & 6.807980e-01
\end{bmatrix}
$$