3 80348

IN 03

Califonia Polytechnic State University
San Luis Obispo

Multicriteria Gain Tuning For Rotorcraft Flight Controls
(also entitled The Development of the Conduit Advanced Control System Design and Evaluation Interface with a Case Study Application Fly by Wire Helicopter Design)

Final Technical Report
10/01/95 – 09/30/97

NASA Grant Number
NAG 2-1024

Technical Monitor
Mark Tischler

Principal Investigator
Daniel Biezad

Graduate Assistant
Jason Colbourne

rec'd.
APR 16 1999
CC:202A-3

CASI.

Abstract

The Development of the CONDUIT Advanced Control System Design and Evaluation Interface with a Case Study Application to an Advanced Fly by Wire Helicopter Design

by

Jason Colbourne

Handling qualities analysis and control law design would seem to be naturally complimenting components of aircraft flight control system design, however these two closely coupled disciplines are often not well integrated in practice. Handling qualities engineers and control system engineers may work in separate groups within an aircraft company. Flight control system engineers and handling quality specialists may come from different backgrounds and schooling and are often not aware of the other group's research. Thus while the handling qualities specifications represent desired aircraft response characteristics, these are rarely incorporated directly in the control system design process. Instead modern control system design techniques are based on servo-loop robustness specifications, and simple representations of the desired control response. Comprehensive handling qualities analysis is often left until the end of the design cycle and performed as a check of the completed design for satisfactory performance. This can lead to costly redesign or less than satisfactory aircraft handling qualities when the flight testing phase is reached.

The desire to integrate the fields of handling qualities and flight control systems led to the development of the CONDUIT system. This tool facilitates control system designs that achieve desired handling quality requirements and servo-loop specifications

in a single design process. With CONDUIT, the control system engineer is now able to directly design control systems to meet the complete handling quality specifications. CONDUIT allows the designer to retain a preferred control law structure, but then tunes the system parameters to meet the handling quality requirements.

This thesis focuses on the development of CONDUIT, providing a detailed description of the system philosophy, components, flexibility, and computational power available in this tool. The philosophies on which CONDUIT is based tries to forge a tool that is powerful, and flexible in an environment that makes setup and analysis very straightforward and easy. The system has flexibility that allows the engineer to design whatever architecture they desire and still provide rapid analysis and tuning of that design. Use of the popular Simulink block diagram environment allows industries to work in a design mode that is easy and familiar. CONDUIT employs a visual environment that allows for rapid setup and provides a powerful view into the performance of the control system.

Demonstration of CONDUIT is accomplished with three UH-60A design problems that show, a design that focuses on meeting ADS-33 (Ref. 3) handling qualities in hover, another design that tuned a model following control system while maintaining as many ADS-33 requirements as the chosen model would permit, and a third design problem that implemented a wind gust rejection control law. The last example illustrates CONDUIT's ability to perform trade off studies by demonstrating the level of wind gust rejection versus actuator authority.

The power and flexibility that has been developed in CONDUIT will be used in multiple areas of control system design. By incorporating CONDUIT in the initial phases of aircraft design, manufacturers can develop improved controllers that are

designed to pilot requirements and understand the true handling of the aircraft in the earlier phases of design. This results in a more efficient, less costly design process.

Acknowledgments

I would like to dedicate this paper to the memory of Stacey Balascio, Dr. Constantinos Lyrintzis, and Dr. Preston Lowery III and the importance of their teachings and influence which gave me the skills to successfully complete this thesis and my schooling as an engineer.

Table of Contents

Section	Page Number
List of Tables	*ix*
List of Figures	*x*
Chapters	
1.0 Introduction	1
2.0 Overview of CONDUIT	3
3.0 Development of CONDUIT	11
4.0 Required Elements for problem setup	29
5.0 UH-60A Design Examples	34
6.0 Lessons learned during thesis	86
7.0 Conclusion	87
References	90

List of Tables

Table 1: The design parameters taken from Tischler (Ref. 9) paper and CONDUIT solution. 51

Table 2: Delay parameters. 57

Table 3: Initial design parameters as provided by Tischler's paper with CONDUIT optimized parameters for the model following solution. 60

Table 4: The final values of the optimization constraints. 66

List of Figures

Figure 1: Example ADS-33 bandwidth specification 8

Figure 2: Example of supporting information that is require for the ADS-33 quickness specification. 17

Figure 3: The handling quality editor provides a convenient set of push button tools that facilitate rapid problem setup. 17

Figure 4: The normalized distance criteria. 21

Figure 5: Distance algorithm calculation. 23

Figure 6: Bandwidth specification with a 0.6 design margin. 25

Figure 7: The CONDUIT environment. 28

Figure 8: Example of a Simulink block diagram 30

Figure 9: The RASCAL UH-60 helicopter. 35

Figure 10: Model following block diagram. 38

Figure 11: UH-60A model following control law. 40

Figure 12: Specs used in ADS-33 control law design. 44

Figure 13: Handling quality analysis of Boeing's control laws. 47

Figure 14: Showing the bandwidth for the yaw channel using Boeing's design parameters. 48

Figure 15: Supporting plots for roll quickness handling quality specification. 49

Figure 16: The broken loop frequency response for the roll channel for 50

the Boeing solution.

Figure 17: ADS-33 solution after tuned within the CONDUIT environment. 52

Figure 18: Yaw bandwidth supporting plots. 53

Figure 19: The supporting plots for the roll quickness specification for small input signal once it has achieved Level I. 53

Figure 20: Roll channel gain and phase margin supporting plot after tuning in CONDUIT environment. 54

Figure 21: The model following specification. 59

Figure 22: Model following specification for the Boeing design parameters. 59

Figure 23: Supporting plot for the model following spec (pitch). 61

Figure 24: Supporting plot for the model following spec (roll). 62

Figure 25: Supporting plot for the model following spec (yaw). 63

Figure 26: CONDUIT's optimized parameters for the model following solution. 65

Figure 27: Supporting plot for the model following spec with optimized solution (pitch). 67

Figure 28: Supporting plot for the model following spec with optimized solution (roll). 68

Figure 29: Supporting plot for the model following spec with optimized solution (yaw). 69

Figure 30: Position and velocity feedback loops for position hold example. 71

Figure 31: The position hold specification. 73

Figure 32: Windgust solution for minimized RMS position. 75

Figure 33: Time history plot of the helicopter's position for minimized RMS position. 76

Figure 34: Supporting plot for actuator energy (roll) with saturation in the actuator rate 77

Figure 35: Medium RMS position hold solution. 78

Figure 36: Time history plot of the helicopter's position for medium RMS position. 79

Figure 37: Supporting plot for actuator energy (roll)) with saturation in the actuator rate. 80

Figure 38: Large RMS position (minimized roll actuator energy). 82

Figure 39: Time history plot of the helicopter's position for large RMS position. 83

Figure 40: Supporting plot for actuator energy (roll)) with saturation in the actuator rate. 84

Figure 41: The trade-off between hover hold and actuator energy. 85

Chapter 1

Introduction

1.1 Introduction

Designing aircraft control systems to meet the formidable set of handling quality specifications as contained in the MIL-STD-1797A (Ref. 10) or ADS-33 (Ref. 11) can be a time consuming task. A control system that is designed to satisfy one or two key specifications may fail to meet the remaining criteria specified in the other handling quality requirements. Developing trade-offs that satisfy several requirements is a main aspect of the engineering discipline and the area of handling qualities is rooted in trade-offs however, the time required to perform a complete analysis handling qualities can be enormous, taking up to a week or more. It is impractical for an engineer to run a series of test cases to evaluate an optimal control system. The results of handling qualities often do not reveal an obvious way to improve the design. An engineer therefore uses control system design methods that involve shaping of frequency responses and time response verification while and not using handling qualities in the design phase. Only after much career experience could an engineer be expected to reflect back on the methods he has used and provide a generalized philosophy on the methods that provide an optimal control system.

There is no single standard architecture (although many have tried to force one in order to use optimal design) and each control system design team seems to have its own standard method that is developed through the projects they worked on. As such there can be variability in control system architectures from company to company. The time involved in testing several architectures prevents rigorous trade-off studies of controller architectures.

Control system engineers have generally developed good designs that satisfy the minimum requirements. They have no way of knowing if this is the best design for the given architecture. There have been several optimal design techniques that often limit the architecture of the controller and as such have limited appeal to the industry engineers. The complex nature of MIMO control laws especially in rotorcraft, which has the nature of highly coupled systems, generally prevents the engineer from seeing the trends in the control laws that would enable fine-tuning that would ensure the "best" design was found.

These aspects of control system design and handling qualities analysis have been under study for the past several years at the Rotorcraft and Powered Lift Branch (AFR) of the Ames Research Center. A key result of these recent efforts is the CONDUIT tool. CONDUIT unites the fields of handling qualities analysis and control system design into one environment. CONDUIT takes it's name from CONtrol system Design and evalUatIon inTerface. The key words from the acronym are design and evaluation interface. These words address the philosophy behind the program. Unlike other

control system theories that dictate the architecture of the controller that might conflict with experiences and knowledge, CONDUIT makes no assumptions as to the "right way" to build a controller. CONDUIT is a powerful tool that allows you to evaluate, compare and tune your control system. It is designed to evaluate the architecture that you provide to the system.

The chapters of this thesis will provide a detailed outline of the features of CONDUIT, along with designs performed on the model following control system of the UH-60A Blackhawk helicopter.

1.2 Research Objectives

The research objectives fall into two main categories: the development of a control system design tool that would allow rapid evaluation and design of control systems against handling quality criteria; use of this tool to evaluate and tune Boeing's model following control laws for the RASCAL research helicopter at NASA Ames Research Center.

The development of the tool, CONDUIT, came partly out of the desire to facilitate the analysis of RASCAL handling qualities and incorporate this as part of the design process. The design of CONDUIT focused on the philosophies of providing a visual environment that provided powerful and complete analysis combined, with an easy-to-use interface that allows for push button setup and evaluation. The second set of

key philosophies was a flexible setup area that allows the user to create any architecture for their controller with a minimum of restrictions.

Three RASCAL design examples were used to demonstrate CONDUIT capabilities, to find weaknesses in CONDUIT, and to develop strategies to best interact with the CONDUIT system. These designs were used throughout the development of CONDUIT as test cases to verify the algorithms. Also the use of a system that allowed the design to satisfy handling quality requirements found that several specifications would hinder the tuning of a design. Information concerning the proper use of specifications in these designs provided a strategy that allows us to rapidly tune and analyze control system design.

Chapter 2

Overview of CONDUIT

2.1 Statement of Purpose

This thesis addresses the concerns of integrating the areas of handling qualities with control system design. Providing a powerful and flexible environment was a major component of this thesis work along with developing tools that provide detailed analysis of the control system. CONDUIT's ability will be featured in three design cases of tuning the control laws for a model following RASCAL control system. Through these design cases, strategies and techniques will be developed for operating CONDUIT in the most effective way.

2.2 The Philosophy of CONDUIT

CONDUIT was developed to bring handling quality requirements into the control system design process. The method of building the system is based on a few key philosophies which provides the foundation for making a useful tool for control system design.

CONDUIT has been laid out with three principle philosophies: the first principle was that CONDUIT should be powerful by automating all handling quality calculations.

The time spent calculating a set of handling qualities specifications can be enormous and is one of the key reasons that companies don't use the specs for design.

The second principle stresses that CONDUIT remain flexible. CONDUIT won't dictate the design of the control system. CONDUIT is required to be flexible so that it doesn't discourage engineers by limiting their control system architectures. The key is that CONDUIT will evaluate the design of the control system and help tune the control system. CONDUIT will not constrain the design of the controller. The CONDUIT program provides the flexibility that allows the engineer to build to any control system architecture desired. The tuning features will work with the architectures given that the parameters selected to tune the system indeed produce an effect on the system.

The third principle is to make the program easy to use. CONDUIT requires a push button environment that allows for easy setup and analysis. The program needs to have the ability to communicate a large amount of information with only a few key strokes. The idea is to provide information to users so that they can understand the system and trust the results that the computer provides.

2.3 The Features of CONDUIT

The entire control problem setup is done in a Graphical User Interface (GUI) environment that allows the control system engineer to perform rapid set up and analysis. Providing the user with an in-depth understanding of their control laws is crucial, so

CONDUIT provides supporting figures for all calculations that go into handling qualities analysis.

The major problem with the previous prototypes of CONDUIT was the amount of knowledge and effort required to setup a specification. Hundreds of lines of code in various languages were required for each problem and then it was tied only to that control system architecture and only for that setup.

The CONDUIT environment uses MATLAB (Ref. 6) as the arithmetic engine of the program to manage all the control system functions. All control laws are entered in MATLAB's Simulink environment. This provides a great deal of compatibility with companies that have developed control systems with MATLAB. All graphs and analysis plots are performed in color with use of MATLAB.

The block diagram is the symbolic representation of the mathematical dynamics of the aircraft and control systems. The block diagram is generated in the Simulink environment. CONDUIT provides the user access to the Simulink environment as well as importing Simulink files generated in MATLAB.

The handling qualities specifications are the key parameters that drive the whole system. The specifications provide both an evaluation of the control system as well as criteria that drives the optimization routines for tuning the system. Handling quality specifications such as ADS-33 for Military Rotorcraft and MIL-STD 1797A for fixed wing are charts that qualify aircraft performance into 3 regions: Level I - implies that desired performance is achievable with no more than minimal pilot workload, Level II -

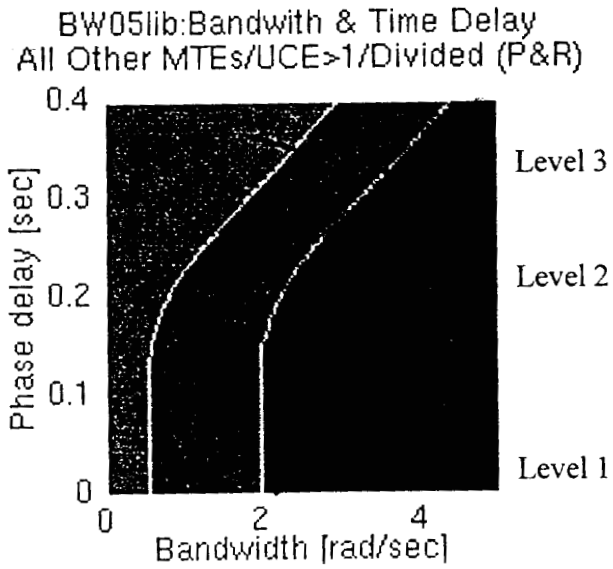


Figure 1

Example ADS-33 bandwidth specification.

allows the pilot the mission to be completed with an increased workload and/or some degradation in performance, and Level III - implies the aircraft can be controlled but that the mission cannot be successfully completed (Ref. 7). An example of ADS-33 handling quality specifications is provided in figure 1.

The specification scripts are structured in modules that are self contained. This approach allows for the simple addition and modification of the specifications by the user.

Specs are easily manipulated in the CONDUIT GUI environment. Specification setup is a rapid process with little knowledge of computer code required to implement the specs. A master editor for the handling qualities makes setup easily performed. The axis and the boundaries can changed with the mouse. This gives the user the advantage of customizing specs and running "what if" scenarios to comprehend elements of the control law. CONDUIT gives the user the ability to relax specification boundaries, that can't be achieved with the current aircraft.

Most engineers approach any new tool with a healthy eye of skepticism. Many programs that promise the answers to complex engineering problems go unused unless the engineer has confidence in the answers. To gain confidence engineers require proof, a way to validate the calculations, and the approach to calculations. They won't use an answer they can't justify. That is why CONDUIT features supporting plots that are contained in the specifications modules. These plot are produced with the click of the mouse and provide a series of plots that tell the user where the data point on the specification came from. The user then has confidence in the answers that CONDUIT produces.

After the user analyzes the performance of the aircraft at one set of design parameters it comes time to improve the control system design. However it isn't always clear how to proceed. To help the engineer tune his given control system a state of the art

optimization engine is integrated into CONDUIT. The CONSOL-OPTCAD optimization software is a FSQP (Frequency Sequential Quadratic Program) algorithm that provides for multi-criterion optimization. The power of multi-criterion optimization allows for the tuning of several design parameters while optimizing to several design criteria simultaneously.

Chapter 3

Development of CONDUIT

3.1 Previous Work Done

The early development stages of CONDUIT began in a joint program between Ames Research Center and the University of Maryland and evolved through many stages that took key contributions from several people during its over five year development. A primary contribution came from professor André Tits of the University of Maryland in the form of CONSOL-OPTCAD (Ref. 2). This program although separate from CONDUIT is used as the optimization engine that provides the ability to tune complex control systems.

The mating of MATLAB with CONSOL-OPTCAD to form the first generation of control system optimization developed by Michael Fan and André Tits. Then a graphical display of the handling qualities was added by Gil Yudilevitch. This later design tool although somewhat difficult to setup contained the first concepts later evolved in the CONDUIT program. The work done in this environment used only the UH-60A model following control system and had no possibility to change to different problem without months of work. The problem was cumbersome since the use of multiple copies of the block diagram needed to be used. Failing to keep all files updated presented the risk of different models being used in different types of specs without the user realizing this.

However the main purpose of CONDUIT was born out of this work: The combining of the handling quality specifications into the control system design phase.

The next generation of this code was shaped by Patrick Potter, Chujen Lin, and Mark Lehene. Their work placed a GUI interface , named GIFCORCODE, over the work done by Gil Yudlevitch. Although this allowed for the running of the CONSOL-OPTCAD optimization engine in a GUI environment as opposed to the text commands used in the work done by Gil, it didn't allow for any increased level of ease or flexibility in the problem setup. That was still in the form of lengthy code that would need to be read, understood, and altered by any user. Implementing a new problem in this program could take weeks or months.

Using the system Gil Yudilevitch had set up, Patrick Potter was able run different design studies on the UH-60A problem. By swapping in the state space model for various flight conditions a preliminary use of CONDUIT was being developed.

3.2 Key contributions of this thesis

The work performed in developing CONDUIT over the past year is the focus of this thesis. The elements that have taken work previously completed on this project and transformed it into a releasable project will be highlighted in the following sections. The incorporation of the philosophies stated at the beginning of this paper shaped the design and layout of the system that made it powerful, flexible and easy to use. The major contributions of this thesis come in the areas of: continuous to digital block diagram conversion, easily altered modular specification, rapid setup and design, a normalization algorithm, robustness in the design process, and a new program environment.

3.3 Continuous to digital block diagram conversions -- "digcon"

The aircraft and control systems for use in CONDUIT are modeled in Simulink. It is this environment that the architecture of the controller is laid out, although a separate program, CONDUIT incorporates the Simulink environment within it.

In keeping with the philosophies of providing an environment that makes for an easy problem setup and rapid analysis of the handling quality specifications, it is necessary to perform the time responses in the digital domain. Typically performing time response calculation with Simulink block diagrams in the continuous domain takes several times longer than the same response done in the digital domain would require. The effect of this on an optimization process, that requires hundreds of time domain simulations, would be to bring it to a halt. Prototypes for CONDUIT used two copies of the block diagram, one in the continuous domain and a duplicate in the digital domain. This allowed for rapid evaluation but risked the chance that discrepancies might be made when altering one and not the other. The time involved in having the user work with several block diagrams and maintaining their consistencies made the setup process more taxing and left an opportunity for errors.

To remedy this, a function (C program) written for use in MATLAB converts all the continuous linear blocks in the architecture to discrete linear blocks. A separate block diagram with the digital components is generated and used for all time response based specifications. The conversion method decided on was Tustin's method. Sample time is

set and the conversion and simulations are run at this sample time. The program can be used on any Simulink block diagram and can be called as a function from within the MATLAB environment. The function is titled "digcon" (digital conversion). Calling this function generates a Simulink discrete block diagram in the form of the continuous diagram for the sample time used. This provides a significant increase in speed for all time response functions over continuous diagrams. The power of this function is that it leaves the nonlinear elements in the diagram so that saturation and other nonlinear effects can be seen in the discrete diagram. A conversion of the end-to-end response to the discrete domain would eliminate the nonlinear effects.

3.4 Specifications

Handling quality specifications are the driving force behind CONDUIT. They provide the unique feature that allows for the control system to be designed to pilot ratings. Control system design is usually done with methods that have to do with frequency response shaping. With CONDUIT the engineer can actually use handling qualities as the design goals. The implementation of specifications in the CONDUIT environment is the core of the system. CONDUIT supports the direct incorporation and tuning of handling qualities in the design process.

The handling qualities specifications are mainly generated from standard publications such as ADS-33 or MIL-STD-1797A . Additional specifications can be created by the users to provide design criteria to investigate issues specific to the project

or just general criteria that enforce common requirements; crossover frequency and actuator energy for example.

The specification is a key element of CONDUIT. All designs made are intended to meet the specifications, so handling quality specifications are crucial to developing the correct and optimum design. The two key elements that they need to possess to keep the spirit of the design are the ability to implement them easily and, more importantly, to see what they are calculating and how they influence the design. The key elements of the specifications are discussed in the five following sections.

CONDUIT has the specs setup as a series of modules (provided or user defined) that define the geometry of the chart, perform the calculations required for the specifications and finally generates supporting plots and data that justify the results. The module is chosen when you wish to analyze that specification. The setup is easily performed in a GUI environment..

The borders or splines define the boundary between the sections of the Level I, two, and three regions. A new feature added during this thesis was the ability to move the splines so that the requirements used in the optimization could be altered quickly and visually. The mouse can move boundaries to drive certain specs harder and relax other specifications that may be limiting the design process.

The first thing that needed to be done was to turn the specs into modules that could be added and removed according to the designer's desires. Having this done effectively required an area to choose the specifications from. This area is called the library. Here pictures of every spec are stored in sections that separate the various types of specifications. The user can scroll through and click with the mouse on the

specifications that the user requires for the design. The new spec will be added to the handling quality window. The handling quality window is the primary work region that chosen specs are displayed in. All setup, interaction, and information occur in this window.

Providing the user with detailed information of how the control system is performing in an easily read presentation is part of the strength of CONDUIT. An important feature to the CONDUIT program is the supporting plots contained in each of the specifications. This allows for intensive analysis of the calculation of the spec. There is no blind trust that is required for the results. Each of the results is justified with a series of plots and data that reinforce the calculations involved in the specification. Supporting plots display the process involved in generating the data for these specifications. This feature is a powerful and required feature that allows the engineer to trust and understand how the results are generated. It removes the tedious repetition of calculations that would be required to double check the results with each answer CONDUIT provided. The supporting plots are activated by clicking on a spec after results have appeared on the specifications. A supporting plot is a window which shows the graphs and useful information that went into the calculation of the spec. Figure 2 shows an example of the graphs used in generating the data for the quickness handling quality specification.

3.5 Handling Quality Editor

One of main appeals of CONDUIT is the ease in which the problems can be setup in the GUI based environment. A key tool that facilitates problem setup is the handling

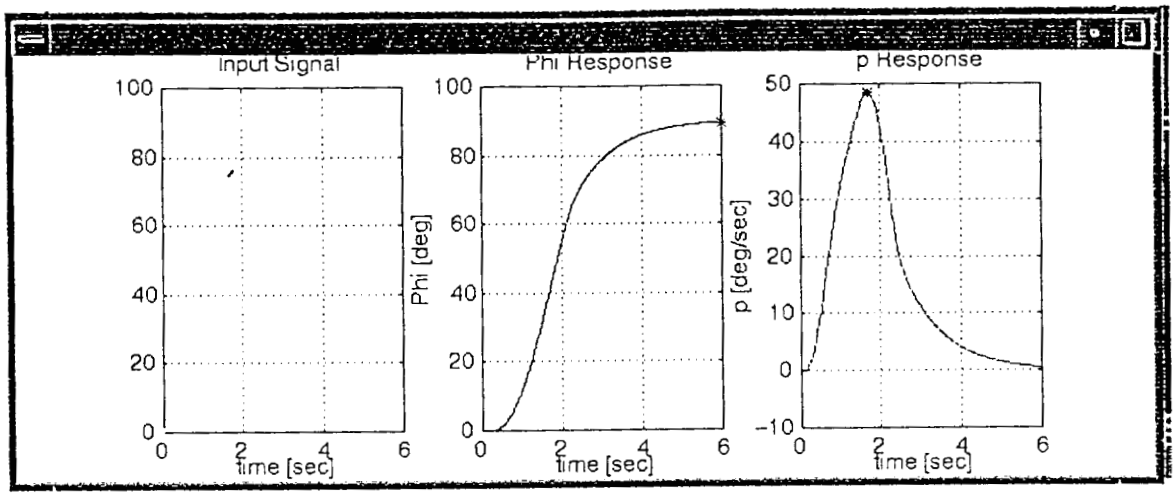


Figure 2

Example of supporting information that is require for the ADS-33 quickness specification.

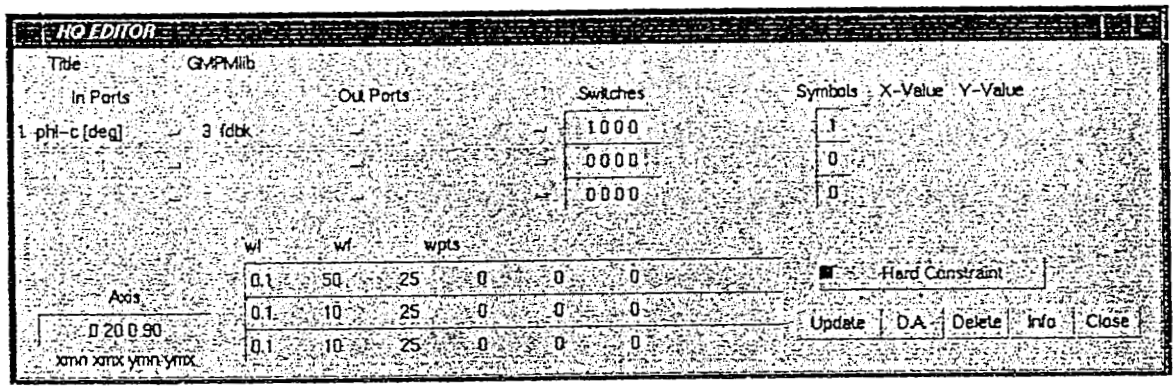


Figure 3

The handling quality editor provides a convenient set of push button tools that facilitate rapid problem setup.

quality editor. This window (Figure 3) allows the user to connect the spec to the block diagram. The first column in the editor provides a list of input ports that are taken from the block diagram. The inport indicates the starting control path for the specification. This either represents where an input signal enters the block diagram for a time simulation based spec or it represents the input for the frequency response calculation. The three rows allow for up to three channels or cases to be run on a single spec, typically pitch, roll, and yaw or three different signals into the same channel on a time specification. The second and third columns allow for either one or two out ports. These out ports indicate the other end of the control path from the in port. Frequently specs will require two outports, i.e., rate and position from a single input signal. For ease in setup of the in and out ports the editor provides a pull down list of both the port numbers and names as labeled on the block diagram.

The fourth column allows for switches that alter the control paths in any fashion that the designer can invent. Up to four switches can be thrown at once, with the understanding that a single switch can be used to throw multiple switches. With this ability the designer can produce very sophisticated block diagrams.

The next column is used only in time domain specifications. This column allows the user to designate the input signal vector defined in the initialization file. This vector represents a time history input such as pilot commands or wind gust perturbations.

The last user defined column allows the user to specify the symbol being used in the diagram. This feature allows the user to maintain a consistent notation for similar control paths. To the right of the last column is an area that indicates the horizontal and

vertical values of the that channel for point data only. The user when needing to know the exact x and y position of a data on a handling quality chart can refer to these columns. Line data such as time histories will not provide a value here. The bottom of the spec builder provides information such as the axis values that can be edited as well as a list of options that are specific to the individual spec. The lower right contains a series of push buttons. The long button allows the user to set the optimization to one of four options: hard constraint, soft constraint, objective function, or check only. This is used to specify the priority during the optimization. The next row of buttons provide access to series of features.

The update button sets the entered parameters to the spec once they have been entered by the user.

The info button opens a window containing information specific to the spec. The information contains a detailed history of where the spec comes from as well as the purpose of the specification. Also contained in the information box are details on how to setup the spec ensuring units and other details are correctly set by the user.

The button marked D.A. allows the user to map points on the spec and check the calculated normalized distance calculated by the distance algorithm. This allows the user to verify that the distance algorithm is converting the proper normalized value being compared to the competing specs.

The close button simply closes the handling quality editor. Failure to update new entries before pressing this button will result in the loss of new data.

3.6 Distance Algorithm

The distance algorithm is the key element that allows for the tuning of the system. The optimization engine performs the comparisons in-between the handling qualities specification. In order to perform the comparison a standard scale had to be used in which all of the specs could be rated. The boundaries are obviously the starting point for measuring the rating of the spec. It seemed obvious that specs that had results resting on the boundary of Level I and Level II regions would all be equal, likewise all data located on the Level II / III boundaries would be weighted equally between all of the specifications.

With these two starting points as a scale for comparing the specs the following rules were developed: The value given to these regions will be a 1 for data on the Level I / II boundary. The value of 2 is given to data with the values lying on the border of the Level II and III regions.

The distance algorithm makes comparisons during the optimization of the specifications in a single problem. Normalizing the value of specs against other specs is crucial in evaluating the overall performance of the controller and aircraft. Looking at the bandwidth and quickness spec in Figures 4 (a) & (b) the properties used in determining the algorithm in the distance algorithm can be determined.

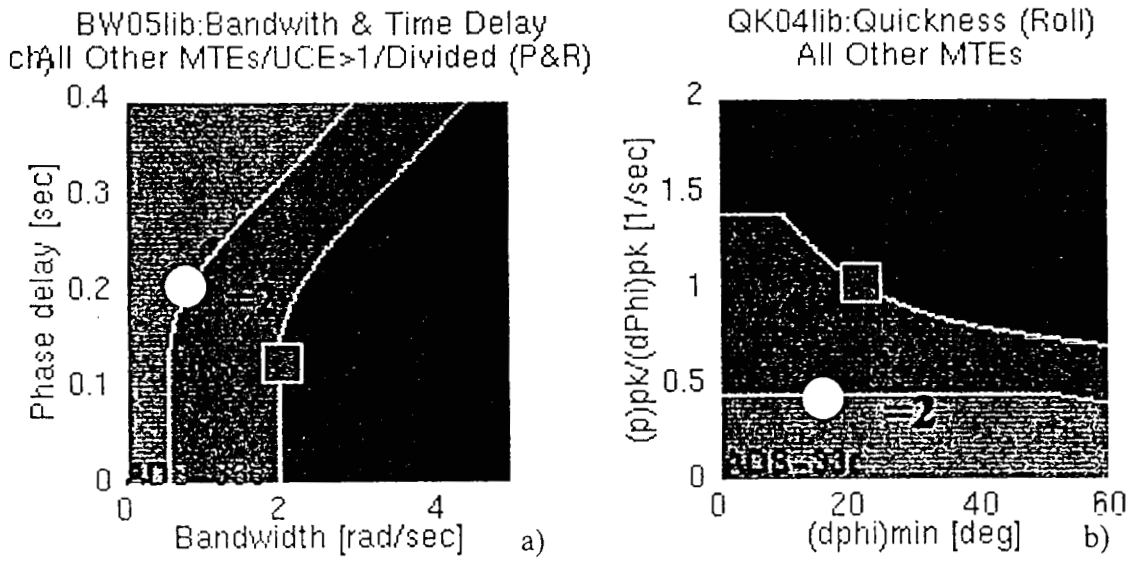


Figure 4

The normalized distance criteria.

a) All specifications that have data located on the level one, level two boundary are rated equal in performance to all other specifications that have the data on this boundary. (b) Similarly the data located on the level two level three border will be rated equally across all specs.

Examining the bandwidth spec there are two borders that separate the different regions on the plot. A value of one will be given to any data lying on the border that separates the Level I and Level II boundaries. Likewise a value of two will be given for data lying on the border separating the Level II / III areas on the plot. Using these two marks the rest of the algorithm was based. This basic rule implicates that the drop from the Level I / II border to a Level II / III border is an equal reduction in performance or handling qualities for all the specs. This holds true for the development of specifications that the change of spec from one border to the next is equal to the change from a border to the other in all other specs. With this normalization all specs can be measured in the distance separating the borders and scaled to the value of one and two. The complexity of the algorithm goes into determining how to measure the borders to the data point. Having the distance algorithm determine this distance for a wide variety of spec shapes, the scale is performed to the following series of rules and calculations. The charts in figure 5 are used to depict the processes involved in generating the scaled values. The distance algorithm calculates the distance of the closest point on the spline to the data point (d1). From this line between the data point and the spline the closest distance is searched for within 30 degrees of this spline. This was done so that line distances were taken from the same direction and to avoid problems that arise in specs that have "U" shaped splines or 90 degree bends in the spline. This second is labeled d2 in figure 5. The width of the Level II region is d3. This is found according to equation 1,2 and 3.

$d3=d2-d1$: for $d1 < d2$ (Level I)

$d3=d2+d1$: for $d1, d2 < d3$ (Level II) Eqns. 1, 2 and 3

$d3=d1-d2$: for $d1 > d2$ (Level III)

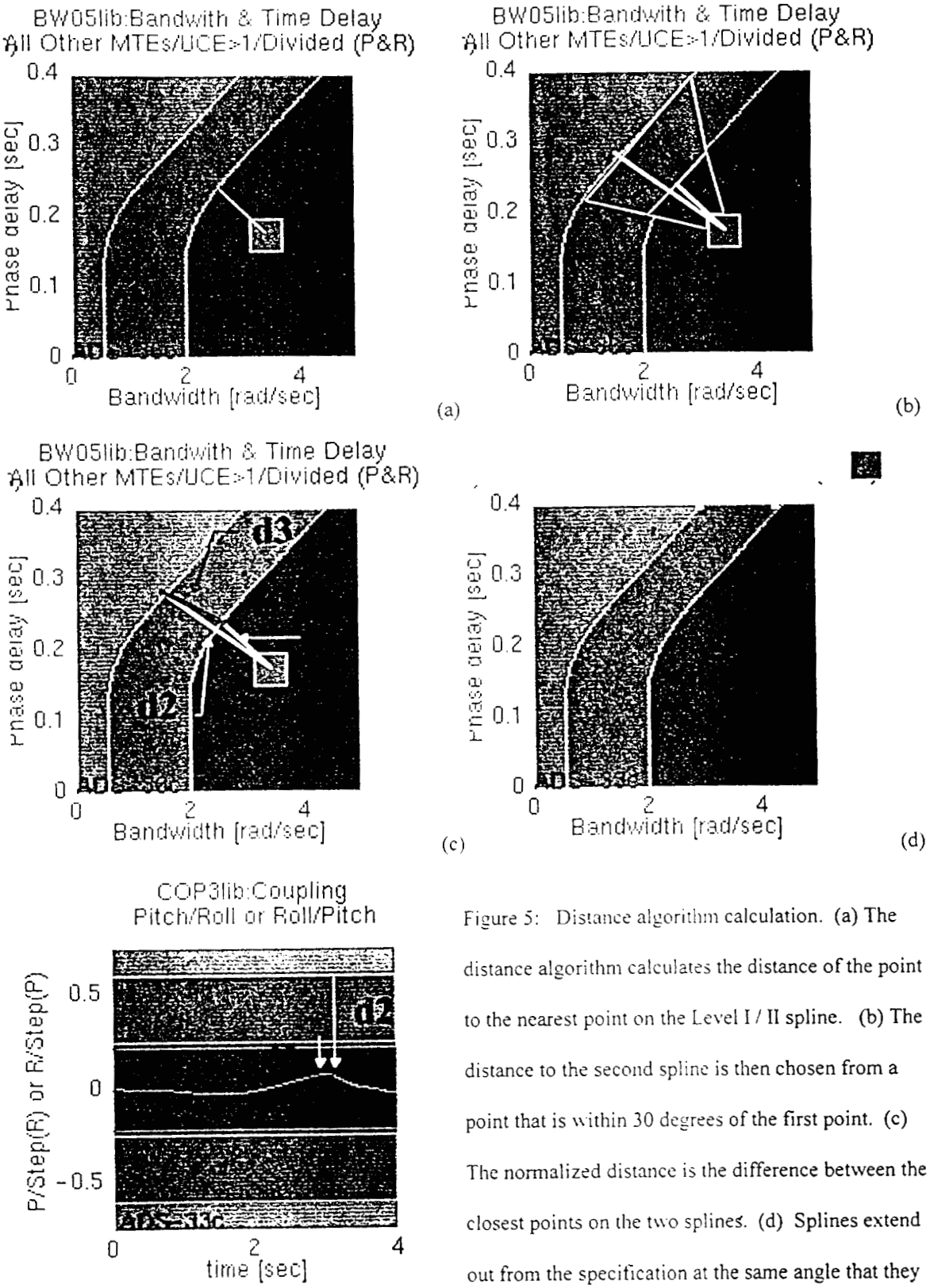


Figure 5: Distance algorithm calculation. (a) The distance algorithm calculates the distance of the point to the nearest point on the Level I / II spline. (b) The distance to the second spline is then chosen from a point that is within 30 degrees of the first point. (c) The normalized distance is the difference between the closest points on the two splines. (d) Splines extend out from the specification at the same angle that they approach the edge. (e) Specifications that use vector data are determined by the "worst point" of the vector.

The value for d3 is approximate since the d1 and d2 lines don't always lie on top of each other. The normalized value is calculated according to equations. 4, 5, and 6, for the Level I, II, and III regions respectively.

$$nd = 1 + \left(\frac{-d1}{d2 - d1}\right)$$

$$or$$

$$nd = 1 + \left(\frac{d1}{d2 + d1}\right) \qquad \text{Eqns 4, 5, and 6}$$

$$or$$

$$nd = 1 + \left(\frac{d1}{d1 - d2}\right)$$

The specifications that use vector data instead of point data, the algorithm takes the point on the vector data that has the worst value and uses that as the data point.

3.7 Design Margin

With all math models being an approximation of some degree of the actual system or aircraft it is always better to incorporate a level of robustness so that any inaccuracies between model and aircraft don't reduce the aircraft to a Level II from a Level I. To keep CONDUIT from producing a design that rides on the border between Level 1 and Level 2 a design margin can be incorporated into the handling quality specifications.

As explained earlier the weighting of the specifications against each other is based on the distance between the Level I and Level III areas (the width of the Level II region). This weighting distance normalizes the specs between each other. This normalizing length is also used in the calculating the design margin. The design margin represents a region that extends into the Level I region by factor of the normalized distance. This is represented on the normalized scale that possess a value of one on the Level I, Level II border and a

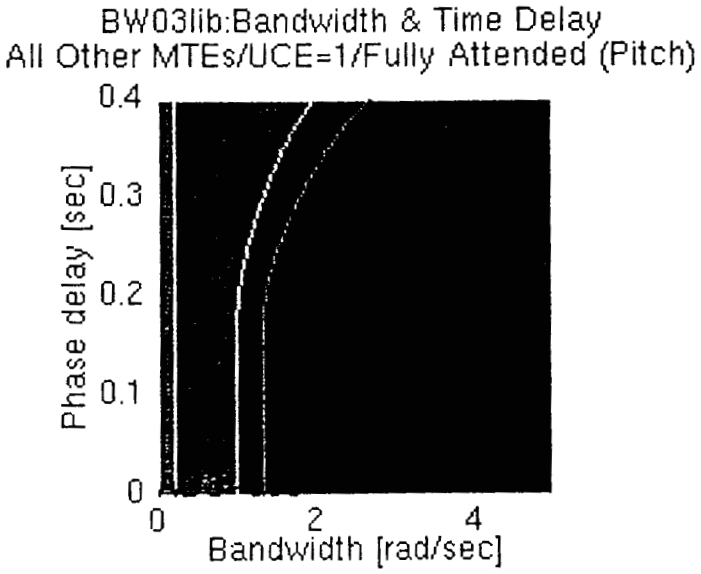


Figure 6

Bandwidth specification with a 0.6 design margin.

value of two on the Level II, Level III border. The value of the design margin represents this value on this normalized scale. For example a design margin value of 0.9 produces a region of ten percent of the normalized distance, for that spec, extending into the Level I region. Figure 6 demonstrates the a 0.6 Design Margin on a Bandwidth and phase delay spec. The varying width of the Level II region is reflected in the distance that the design margin (green line) extends into the Level I region.

3.8 Environment

A considerable change in CONDUIT over the previous prototypes is the centralized integrated environment in which CONDUIT has incorporated all the function and features into. CONDUIT uses a main window with pull down menu and iconified buttons that allow the user to trigger the most frequently used features of the system. The desire in CONDUIT was to make easy-to-use tool that didn't require a significant knowledge of programming to setup and analyze problems. A key advantage over previous versions was that any modification required the user to read through pages of code within several different files, written in different languages and then enter a significant number of lines of new code. Even to change the order or the number of specs it would be a tedious undertaking. CONDUIT allows the specs to be manipulated via the mouse. The addition, modification, and removal of the specifications is easily performed by the "click" of a mouse. Setup is performed with pull down menus that allows the user to install specs with easy push button features. This was done without compromising the flexibility of the system and is what allows it to be so powerful. With a setup and evaluation of an existing block diagram only taking a few hours. CONDUIT is also a very visual environment that displays the information in plots and graphs that aid in the rapid understanding of the control system. It gives the user the ability to see what is happening

with the controller. Figure 7 shows a collage of the CONDUIT interface. Multiple investigation into varied configurations becomes possible within a significantly shortened time period over existing conventional methods.

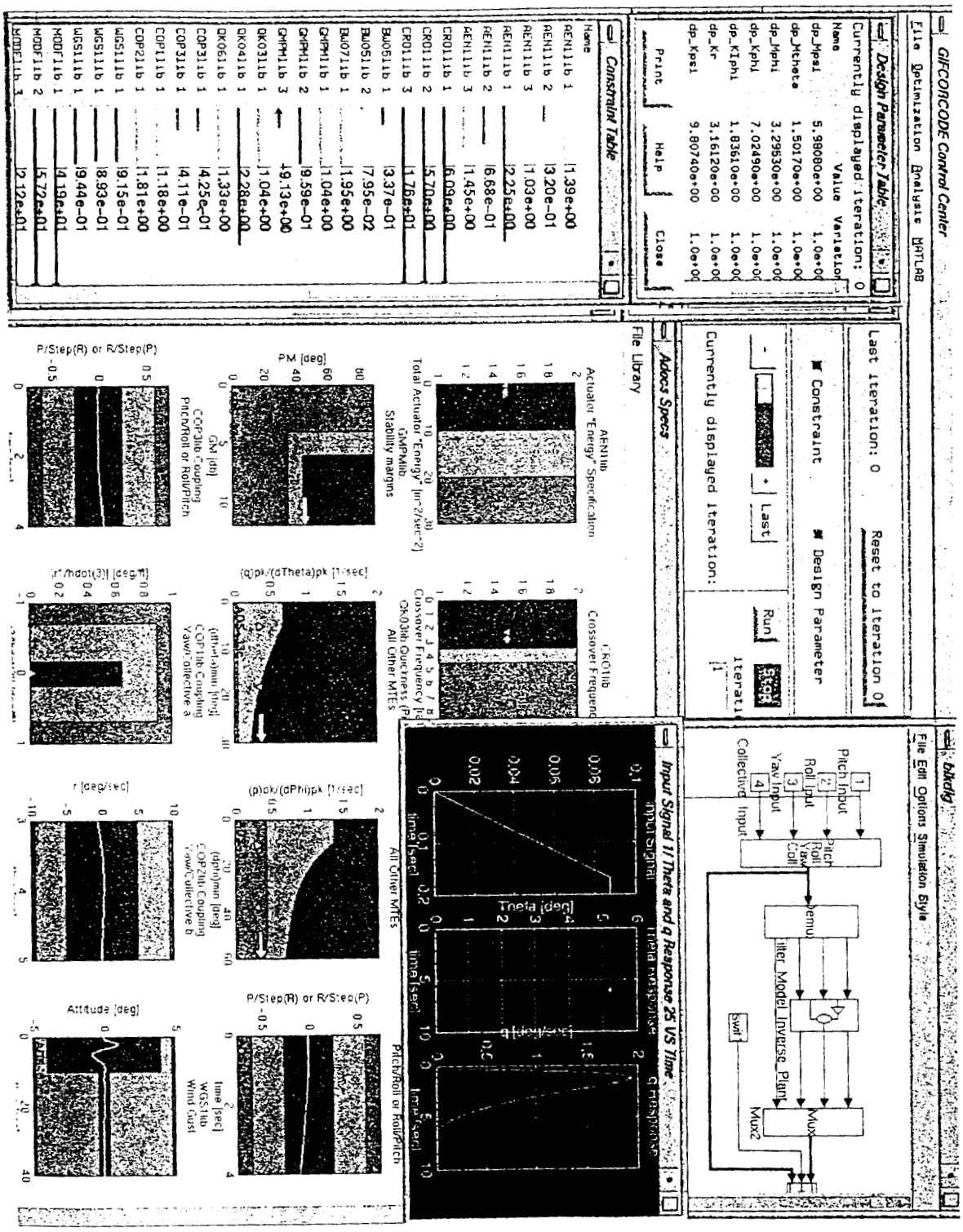


Figure 7

The CONDUIT environment.

Chapter 4

Required Elements for a Design Problem

4.1 Aircraft Model

The aircraft model is the central part of the problem in CONDUIT. The model and control system dynamics are developed in the Simulink environment. The complexity of the model is left to the prerogative of the engineer, there is an obvious trade-off between accuracy and computational time. The Simulink environment supports sophisticated nonlinear designs that can be expressed in a graphical environment that is used in CONDUIT. Figure 8 is an example of a simple block diagram used for an XV-15 demonstration problem that is released with CONDUIT. There are a few minor rules that must be followed to ensure a formatting that will be accepted by CONDUIT. The rules pertain to the selection of mnemonics for the design parameters, switches, and input signals.

An initialization file must be created to support the block diagram to store information that couldn't be included in the model. This file contains the definitions for all the key variables, functions and defining mnemonics. The initialization file is comprised

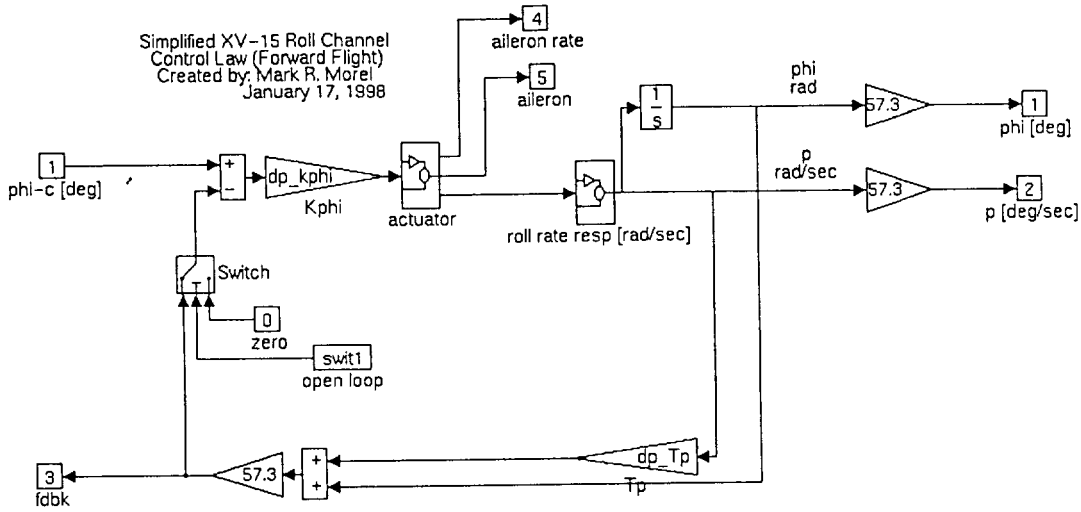


Figure 8

Example of a Simulink block diagram.

of: default parameters that define the input signals that will be required for time domain simulations; modeling data; and the initial value of the switches.

4.2 Handling Quality Specifications

Specifications are key to finding a solution. Once the user has entered the model into CONDUIT they need to select specifications that will allow the user to analyze and improve the controller. The user needs to pick specifications that will provide a complete picture to determine if the controller and aircraft perform as a Level I aircraft. Choosing these specifications is left to the user depending on what aspect of the system is of interest. Once the user has decided which specifications are required, specs can be chosen from the pull down library window with the mouse and be added to the handling qualities window. Once here the specs can be setup and modified by clicking on the individual specification and opening the handling quality editor. The specifications work between ports in the block diagram. Menus provide the information required to 'wire' the spec to the block diagram. The editor allows the selection of the in and out ports, any breaks in the channels that would be necessary, and signals that need to be passed into the specification.

The splines can be moved to suit the desired design by the user. Once the specs have been chosen a robust margin can be selected to prevent the optimization engine from designing to the border.

4.3 System Evaluation

Once the user has set up the aircraft model with controller and chosen and 'wired' the specifications it is possible to proceed to the evaluation mode of CONDUIT. In the initial phase of the evaluation mode all the design parameters are set to the value 1. These

are arbitrary values and can be replaced by other values that might be known for a rough or baseline design. Evaluate the design with either default or chosen parameters. Within a few seconds to a few minutes, depending on the complexity of the model and the number of specs the handling quality window will display the calculated value of the specifications. Typically the default values will give poor results. By selecting the spec a supporting plot can be shown that illustrates why the value is calculated to be that way. Since the optimization can't always find solutions when starting at really bad results it is typical to use classical techniques and intuition in conjunction with the results supplied in the supporting plot to quickly determine a set of design parameters that will produce a solution that might be in the ball park of a reasonable solution. CONDUIT can be used in this fashion quite productively by allowing the user to tune the parameters by hand until a solution is found. Used in this way the tool represents quite a bit of power when other methods require a week to run a new set design parameters through the system. The user can run 'what if' scenarios by changing different aspects of the control system and almost instantly seeing the effect those changes have on a set handling quality specifications.

4.4 Starting the Optimization

The CONSOL-OPTCAD optimization engine allows the user to find the best design in a region. Although the system can tune be tuned manually with the use of classical control techniques and the supporting plots, this is not always easy. In problems that have coupling or multiple feedback gains with forty to fifty specifications and 12 design parameters it becomes difficult to determine which design parameters and in which combination of values will produce an over all improved design. This is when it becomes necessary to use the optimization engine.

The optimization process optimizes according to constraint type chosen in the handling quality editor. The constraints can be divide into three priorities, these are hard constraints, soft constraints and objective functions. The hard constraints must be satisfied for there to be a solution. The soft constraints are used to find a solution by giving the program targets to be met. The objective functions are used to help pick out the optimal solution by minimizing that specification. The program strives to satisfy these specifications in three different phases. Optimization conducted in three phases. In Phase 1 the program moves toward satisfying the hard constraints. CONSOL-OPTCAD adjusts the design parameters to achieve Level I for all the specifications that were set to hard constraints. Typically hard constraints are chosen to be the specs that guarantee stability or those that are of particular interest to the user for this problem. These are specifications that need to be met with no exceptions. In phase two CONSOL-OPTCAD attempts to satisfy the specs, that are set to be soft constraint or objective functions into the Level I region. In this phase these two types of specs behave in a similar matter. These specs are ones that you would like to meet Level I but are possibly willing to compromise if there is no solution. These types of decisions are why this is an interface tool for the engineer and not a "turn the crank" all purpose solution. If all the hard constraints, soft constraints and objective function specifications meet the Level I area then the optimization engine reaches the third phase. In the third phase the objective functions drive the data points for the specification further into the Level I region until a optimal design has been reached. This design is normally characterized by a local minimum.

Chapter 5.0

UH-60A Design Examples

5.1 Introduction to Design Problems

The following designs demonstrate the use of CONDUIT with the Sikorsky UH-60A Blackhawk helicopter. The Ames Research Center operates a Blackhawk research helicopter under the RASCAL (Rotorcraft Aircrew Systems Concepts Airborne Laboratory) program (Ref. 1) (Figure 9). Boeing Helicopters has designed a model following control law using conventional design techniques. In an effort to simultaneously validate the control laws designed by the engineers at Boeing and the algorithms in CONDUIT the Boeing ADOCS control laws for the RASCAL helicopter (29-69, Gulsman) was run through a design session with CONDUIT. Designs are performed by developing a traditional architecture and implementing design parameters; variables that can be used to tune the control laws. Design parameters can be used as gains, pole or zero locations, transfer function coefficients, or any number that represents a value in the block diagram. The first two problems use 12 design parameters, 9 feedback gains, and 3 model parameters for the pitch, roll, and yaw loops.

Figure 9

The RASCAL UH-60 helicopter.

The first problem shows how the RASCAL architecture can be tuned to an optimal design that satisfies the ADS-33 handling qualities requirements. The second design problem focuses on the model following architecture and produces a solution that allows the helicopter's end-to-end response to track a second order transfer function model. The third design problem demonstrates CONDUIT's ability to perform trade-off analysis between competing specifications.

5.2 Blackhawk Design for ADS-33 Requirements

The first design problem addressed in this paper is concerned with establishing a baseline set of design parameters that will allow the helicopter to function in accordance with ADS- 33 control laws (Ref. 11). This set of design parameters will establish a confirmed standard that can be referred to in future designs and specialized optimizations. The Blackhawk helicopter requires a baseline control law that will satisfy ADS-33 handling quality specifications. In this design problem, the model used in the UH-60 model following control system architecture is allowed to change rather than remain fixed. This is done because this problem focuses on the end-to-end response and the resulting handling qualities as the important elements.

The architecture consists of a model following design. The model following architecture will be briefly reviewed here with more detail described in section 5.2. The simplified block diagram (figure 10) will serve to illustrate the concept behind this model following design. Equations in section 5.2 prove the block diagram reduces to the model

M if P^{-1} is truly the inverse of the plant. The model following control system uses an approximation of the inverse plant to cancel out the plant dynamics and uses feedback gain to reduce the mismatch between the model and the plant. Assuming a good model is used, a detailed analysis of how the aircraft will perform to all the specifications is not clear to the designer without calculating each one in CONDUIT or by hand. The model employed by the engineers that developed this system is a second order transfer function that exhibited good characteristics and tried to alter the aircraft's end-to-end response dynamics to meet this model.

The first order inverse plant approximation (P^{-1}) was created first by matching a zero over first order curve fit of the complete frequency response of the pitch, roll, and yaw channels of the aircraft. The transfer function was then inverted and placed into the P^{-1} block in figure 10.

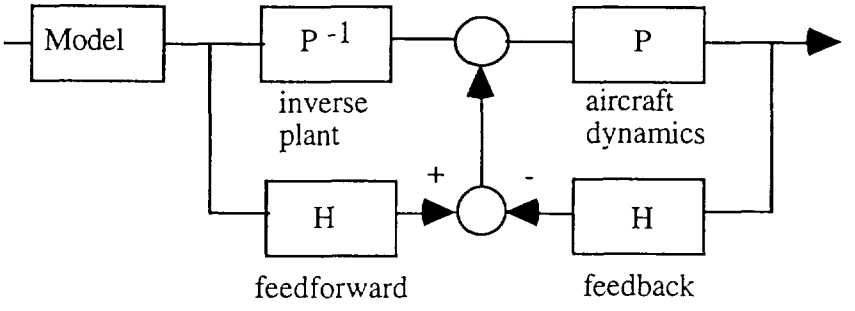


Figure 10

Model following block diagram.

This first design problem was to tune this architecture (including model) so that the design meets the ADS-33 specifications. Figure 11 shows the actual block diagram used in this model. The block diagram consists of a 14 degree of freedom state-space representation of the UH-60A helicopter dynamics (Ref. 3). The inverse plant dynamics (P^{-1}) are the inverse low order transfer function fits of the higher order plant. Feedforward and feedback loops are summed through the feedback compensation block (H). The feedback compensation consists of a PID controller using the 9 feedback gains (3 three for each of the pitch, roll, and yaw channels) as 9 of the twelve design parameters. The remaining three design parameters make up the second order model transfer functions.

The second order inverse of the plant dynamics was obtained by taking frequency data from the plant and crossfeeds together. These data were fed into Navfit, a curve fitting utility of CIFER that produced the transfer functions in equations 7 through 10.

The transfer function fit for the pitch channel is given as:

$$\frac{q}{\delta} = \frac{3.20647}{(s+0.58468)(s+11.770)} \qquad \text{Eqn. 7}$$

The transfer function fit for the roll channel is given as :

$$\frac{p}{\delta} = \frac{16.6734}{(s+6.1178 \pm 3.39789i)} \qquad \text{Eqn. 8}$$

The transfer function fit for the yaw channel is given as :

$$\frac{r}{\delta} = \frac{397.616}{(s+592.64)(s+0.79536)} \qquad \text{Eqn. 9}$$

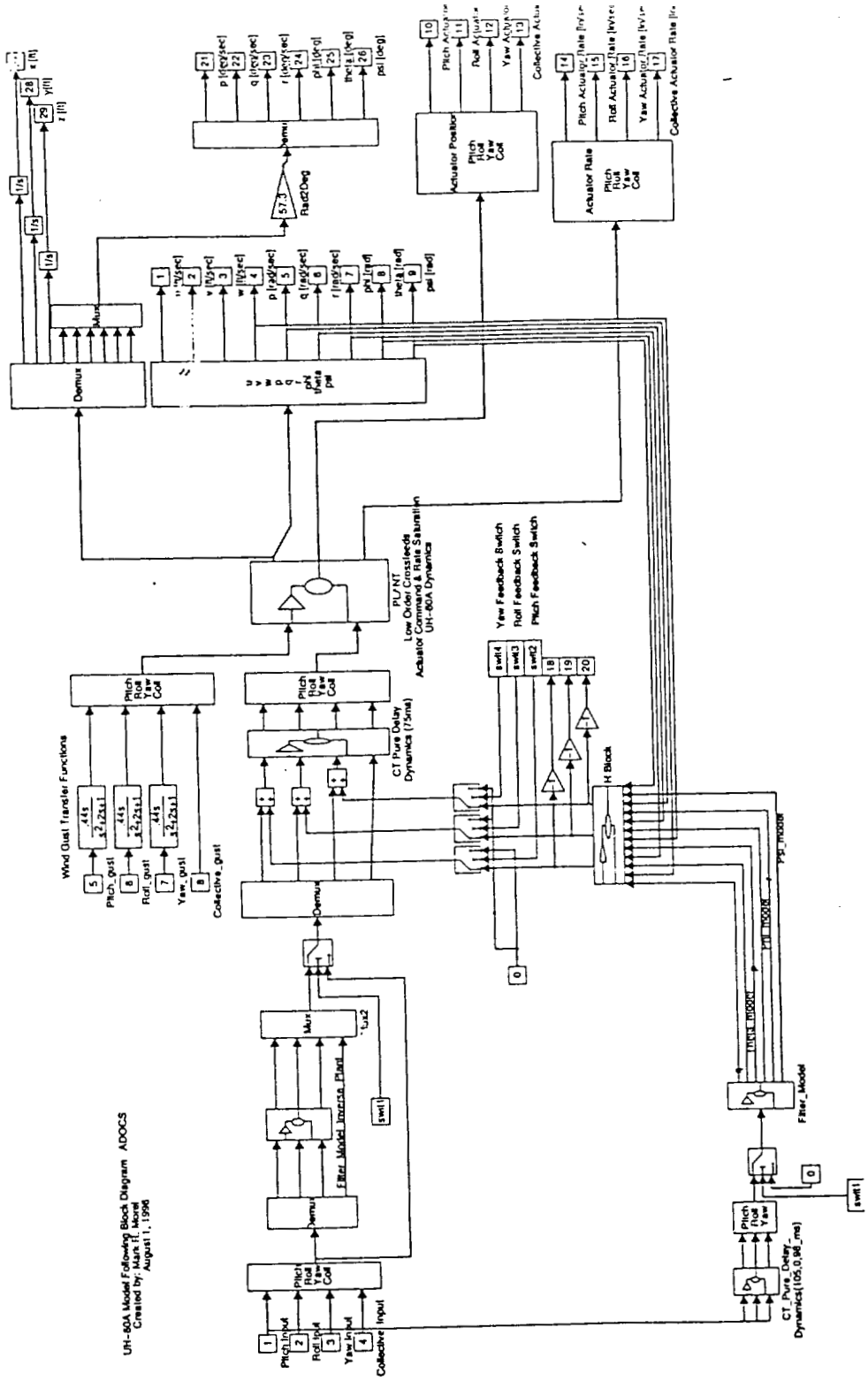


Figure 11

UH-60A model following control law.

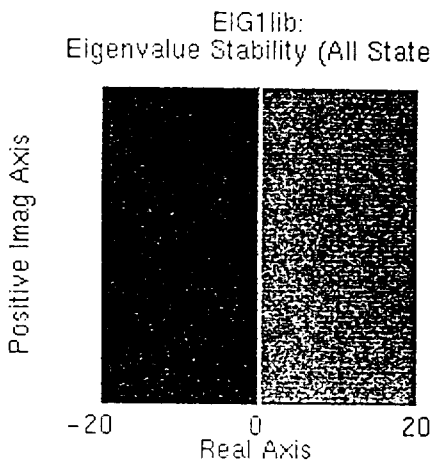
It is obvious that the yaw channel has an unusually high pole and gain. This indicates that the first order inverse will be adequate for the plant inverse. Previous papers (Ref. 8) indicate that the first order is adequate for the Yaw channel. The first order expression is given as:

$$\frac{r}{\delta} = \frac{.6707}{(s + 0.784849)} \qquad \text{Eqn. 10}$$

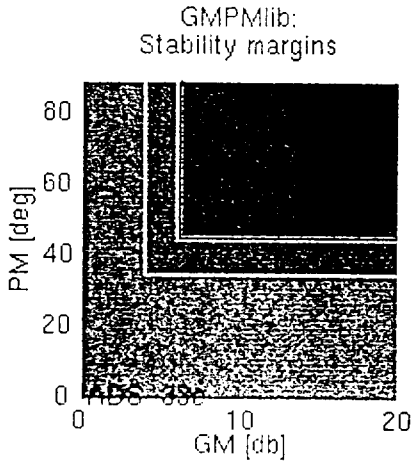
The integral of these three transfer functions (converting q/δ, p/δ, and r/δ to θ/δ, ϕ/δ, and ψ/δ respectively) are implemented into the P^{-1} block for the block diagram to be used in CONDUIT.

The specifications used in this design example come from both ADS-33 (Ref. 11) and those developed at the Ames Research Center based on classical control theory. The specs chosen were: crossover frequency, actuator energy, eigenvalue stability, gain and phase margins, bandwidth, quickness, coupling, and wind gust. Figure 12 shows the specs that will be used in this and the next example design problem. The specs are going to be ranked by their optimization type (hard constraints, soft constraints, and optimization function). The eigenvalue stability specification (12a) uses the real part of the roots of all the poles for system. It drives them into the left hand plane to ensure over all stability of the plant and controller. The Level II region on this spec was made very small to enforce the idea that either the roots are stable (Level 1) or not (Level 2).

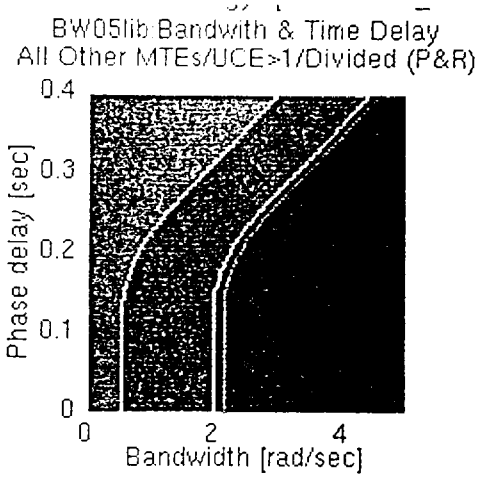
The gain and phase margin spec (12b) also enforces a stability margin using 6 dB of gain margin and 45 degrees of phase margin. This spec is useful in comparison to other similar specifications that measure stability (i.e. damping ratios) since this is smooth and continuous during the optimization phase. Both the eigenvalue and gain and phase margin

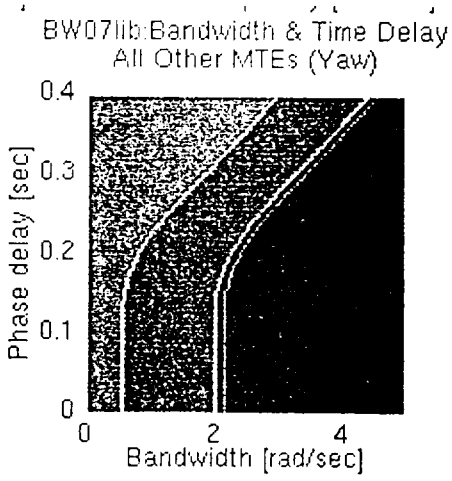

EIG1lib:
Eigenvalue Stability (All States)
Positive Imag Axis
-20
0
20
Real Axis

(a)

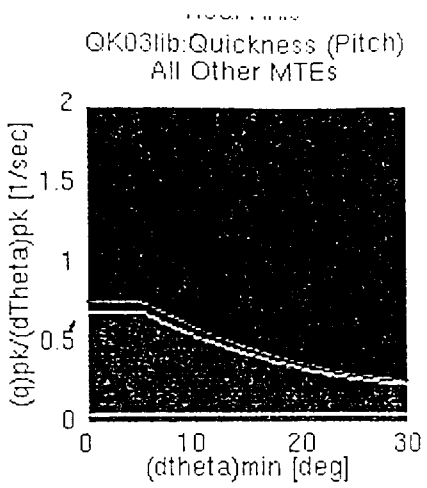

GMPMlib:
Stability margins
PM [deg]
80
60
40
20
0
0
10
20
GM [db]

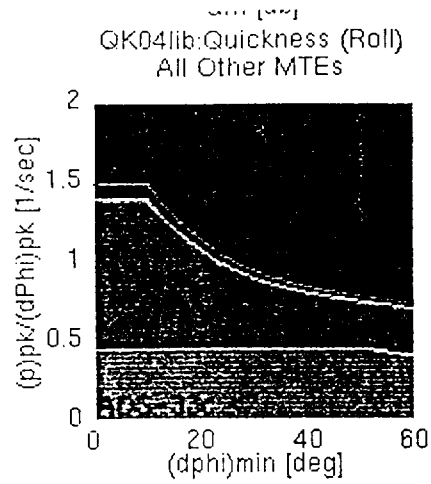
(b)

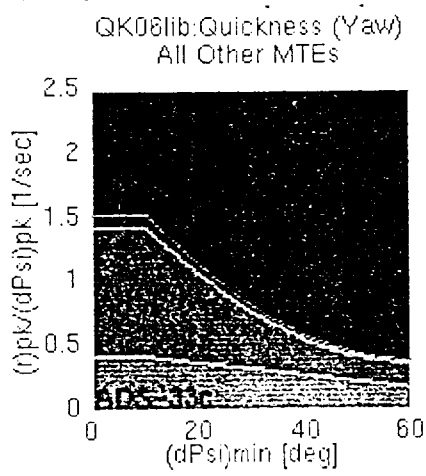

BW05lib:Bandwith & Time Delay
All Other MTEs/UCE>1/Divided (P&R)
Phase delay [sec]
0.4
0.3
0.2
0.1
0
0
2
4
Bandwidth [rad/sec]


BW07lib:Bandwidth & Time Delay
All Other MTEs (Yaw)
Phase delay [sec]
0.4
0.3
0.2
0.1
0
0
2
4
Bandwidth [rad/sec]

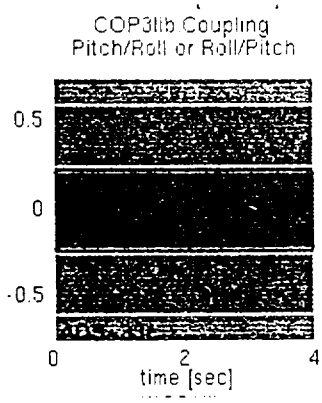
(c)

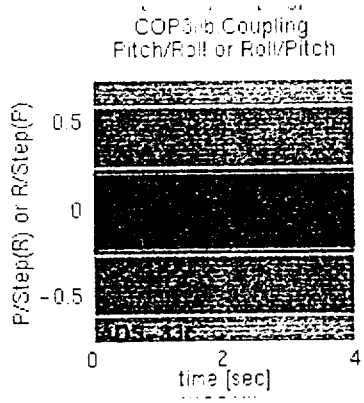

QK03lib:Quickness (Pitch)
All Other MTEs
(q)pk/(dTheta)pk [1/sec]
(dtheta)min [deg]

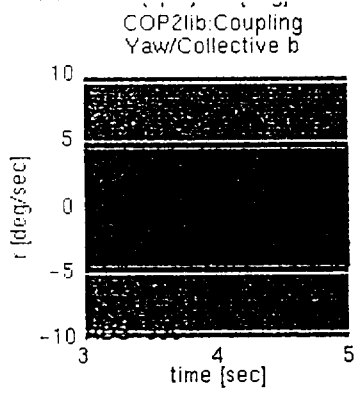

QK04lib:Quickness (Roll)
All Other MTEs
(p)pk/(dPhi)pk [1/sec]
(dphi)min [deg]


QK06lib:Quickness (Yaw)
All Other MTEs
(r)pk/(dPsi)pk [1/sec]
(dPsi)min [deg]

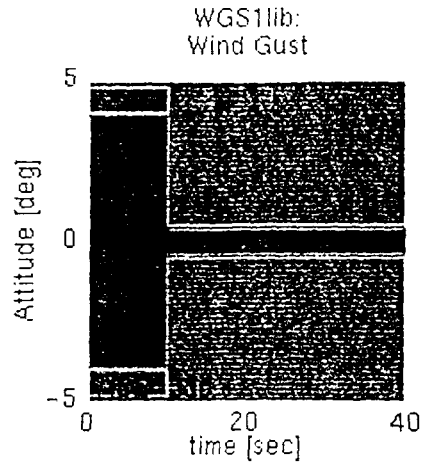
(d)


COP3lib:Coupling
Pitch/Roll or Roll/Pitch
time [sec]

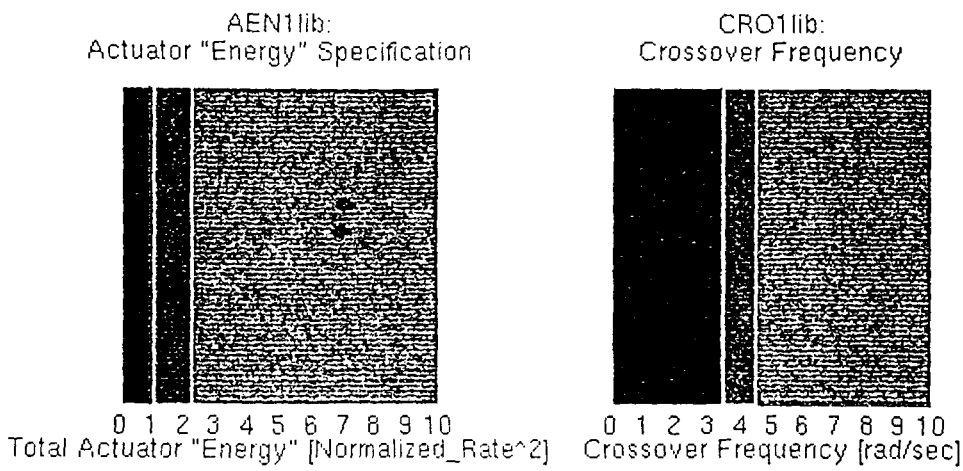

Pitch/Roll or Roll/Pitch
P/Step(R) or R/Step(P)
time [sec]


COP2lib:Coupling
Yaw/Collective b
r [deg/sec]
time [sec]

(e)



(f)



(g)

Figure 12

Specs used in ADS-33 control law design.

(a) eigenvalue stability, (b) gain and phase margin, (c) bandwidth, (d) quickness specifications, (e) coupling specifications, (f) wind gust specifications, and (g) cross-over frequency and actuator energy specifications were chosen to be hard constraint in the handling quality editor. These specifications guarantee stability and that no design would be accepted that had compromised stability.

The bandwidth and phase delay spec (12c) ensures that the end-to-end control system is responsive to a certain frequency. Bandwidth is a measure of responsiveness, too little, and the aircraft feels sluggish. The higher the delay in the system the more bandwidth is required.

The quickness specification (12d) is a measure of agility of the aircraft. The requirement ensures that the aircraft is more agile for small attitude changes and will allow for less responsiveness for larger attitude changes.

Helicopters typically have a large degree of cross-coupling between the channels. It is usual for there to be a significant amount of motion in other directions when a command is placed into single direction. The coupling specification (12e) help minimize the amount of coupling in the off-axes.

The wind gust specifications (12f) enforce a level of stability and damping in the feedback loop. A simulated wind gust should damp out according to the time response envelope of the specification.

The bandwidth, quickness, coupling, and windgust specifications were set as soft constraints since these requirements are not as crucial as stability. It is possible that a Level II solution in some of these specification might be accepted if limited by the aircraft.

The two specifications for optimization (12g) are crossover frequency and actuator energy. These two specifications are complimentary. A high crossover frequency usually allows more noise through the system and forces the actuator to work

harder. Driving down the crossover frequency drives down the actuator energy. These specifications were set to objective functions in an effort to minimize the values and reduce any extra actuator energy not required for a Level I design.

A design margin of 0.9 was selected to ensure a minimum level of robustness. The 0.9 design margin extends the Level II region ten percent more into the Level I region so that solution won't be a borderline Level I / II, solution.

After the specifications were setup, CONDUIT was placed into the run (analysis, optimization) mode. The design parameters were initially set to the Boeing default parameters and the analysis was run (Figure 13). Unless specified otherwise, the data in these figures use the yellow triangle for pitch, the green inverted triangle for roll and the black diamond for yaw. The time response vectors use the same color code for pitch, roll, and yaw as the point data and are always in the order of pitch, roll, and yaw as the specs appear from left to right in the following figures. The example of this order is shown in the wind gust specification. When the Boeing ADOCS control laws are tested against the ADS-33 specifications it is clear that the aircraft quickness is not sufficient in the yaw and roll channels and barely so in pitch. The yaw bandwidth is also solid Level II as seen in figure 14, the supporting plot of the yaw bandwidth for the yaw bandwidth calculation. The bandwidth corresponds to the frequency of the model transfer function in yaw. It obvious that the design parameters dp_Mpsi of the yaw model will need to increase. To examine why the quickness specification didn't meet the requirements for the small yaw inputs; figure 15 contains the supporting plot for the small input yaw channel.

Next CONDUIT was exercised to optimize the design parameters so the controller would meet all the specifications and drive down the crossover frequency and

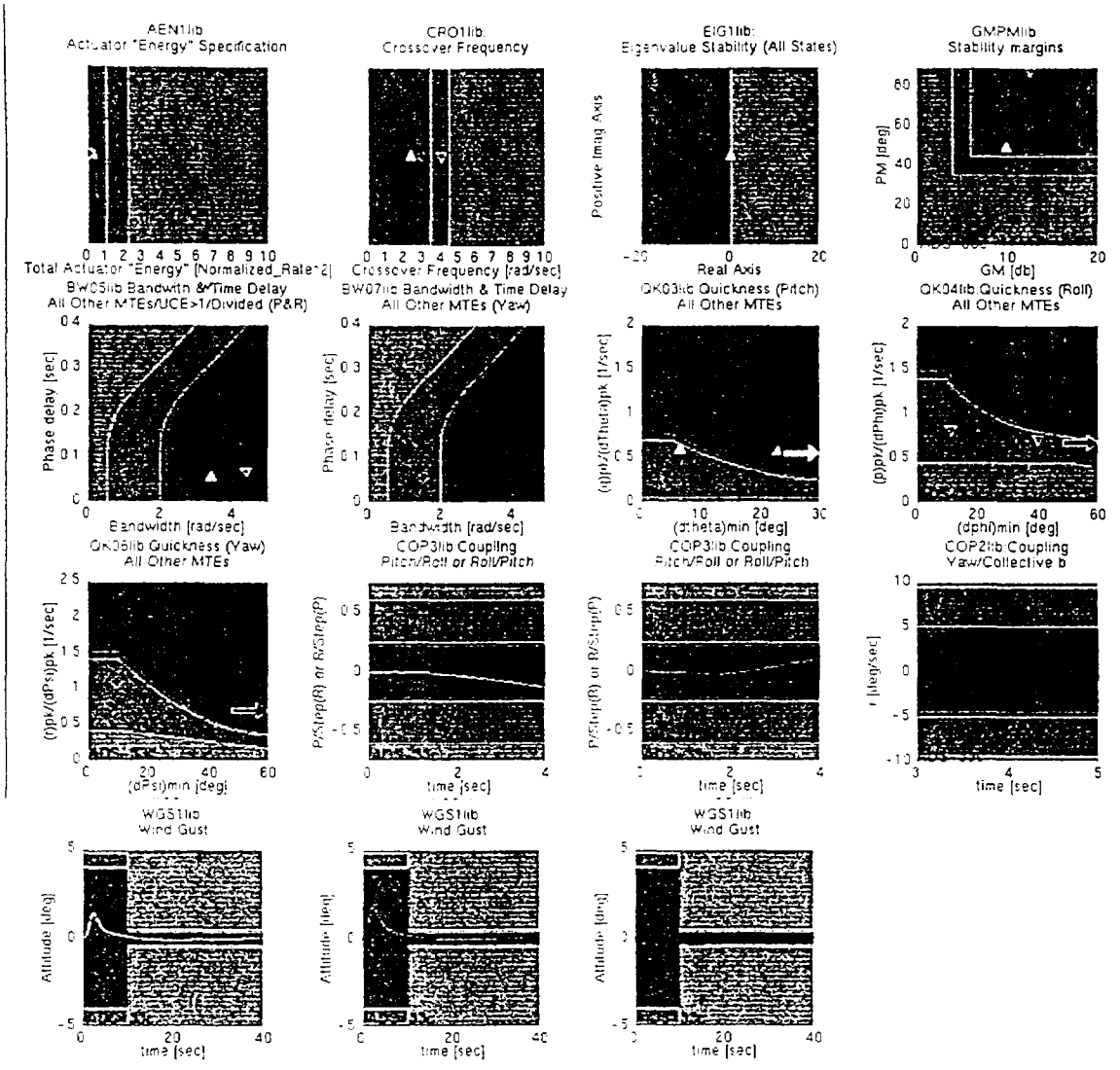


Figure 13

Handling quality analysis of Boeing's control laws

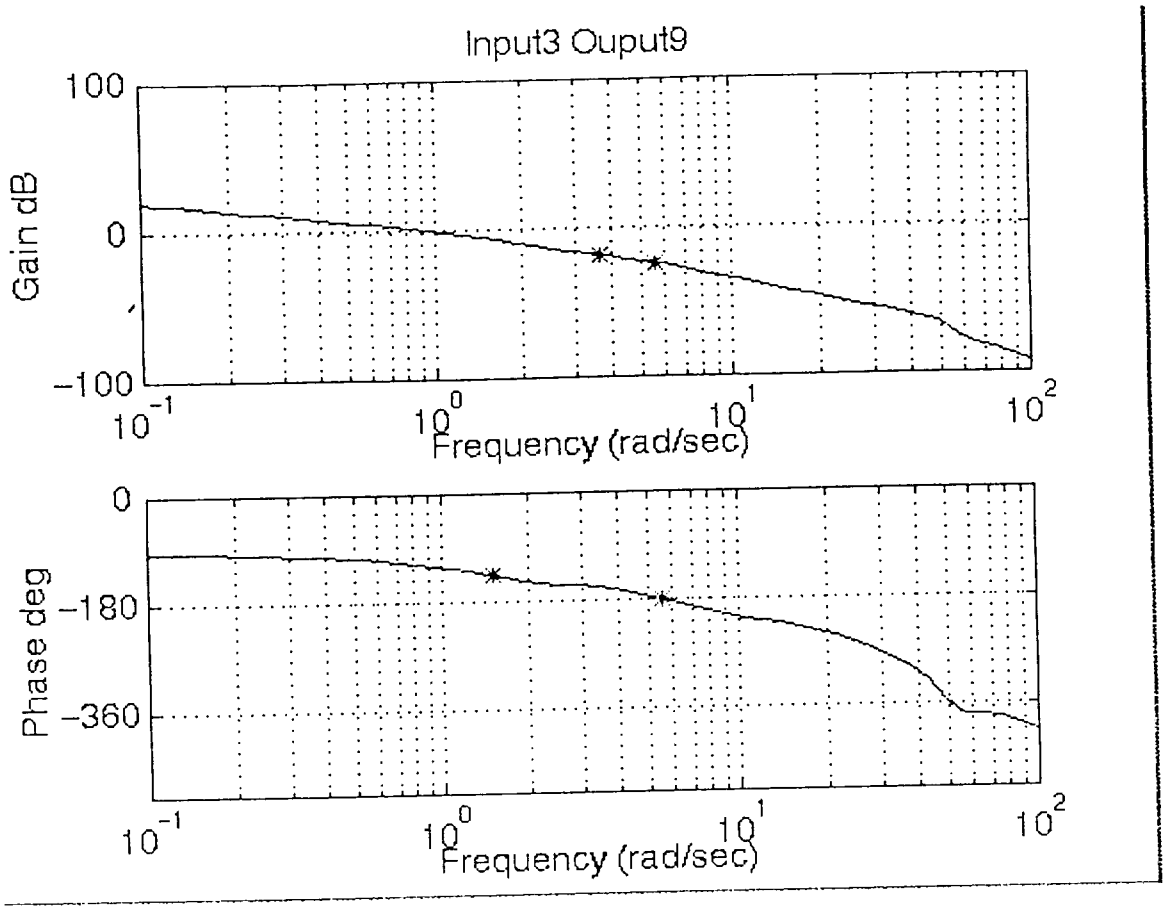


Figure 14

Showing the bandwidth for the yaw channel using Boeing's design parameters.

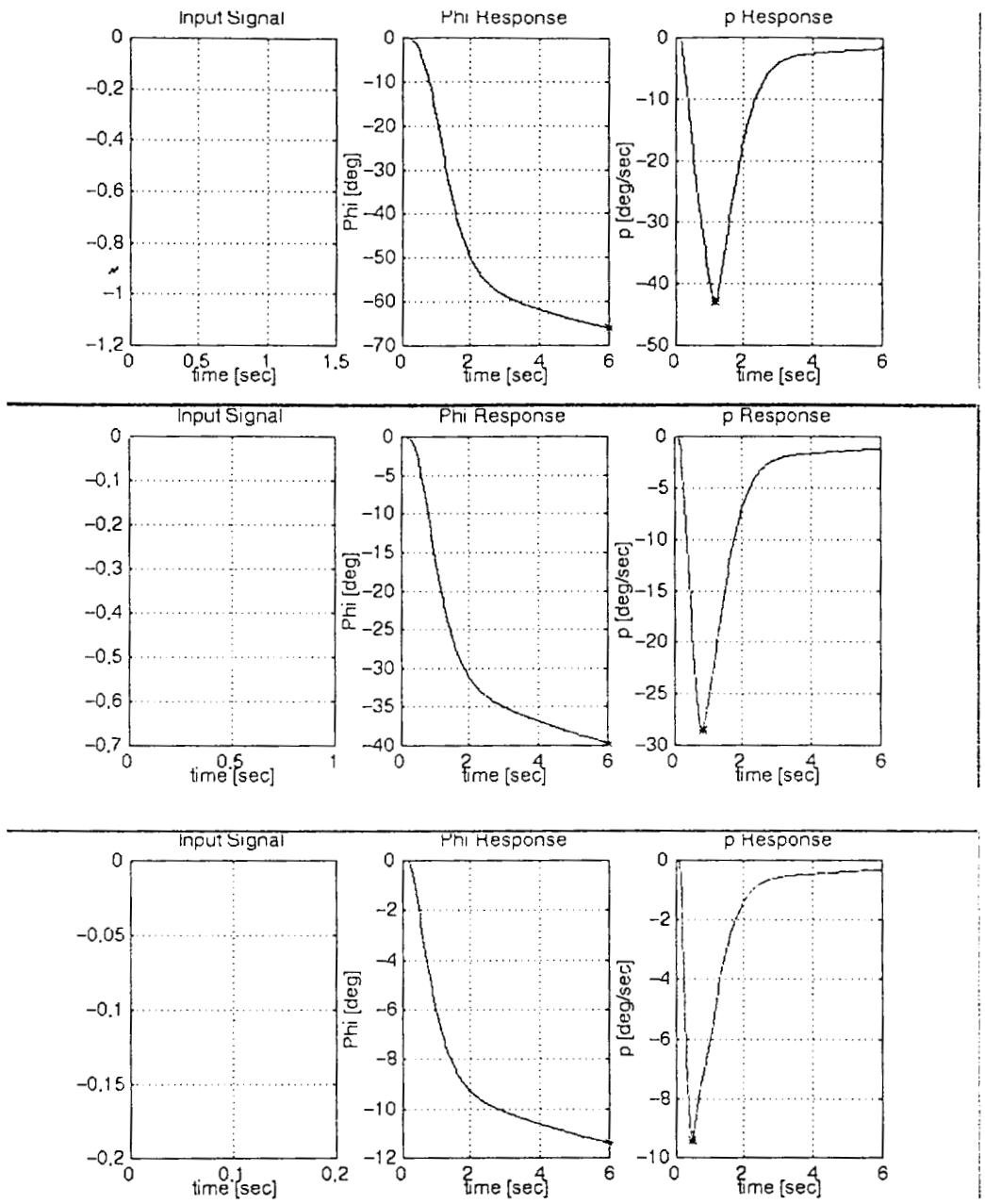


Figure 15

Supporting plots for roll quickness handling quality specification. The input signal is negative because in this direction the aircraft exhibits a poorer response.

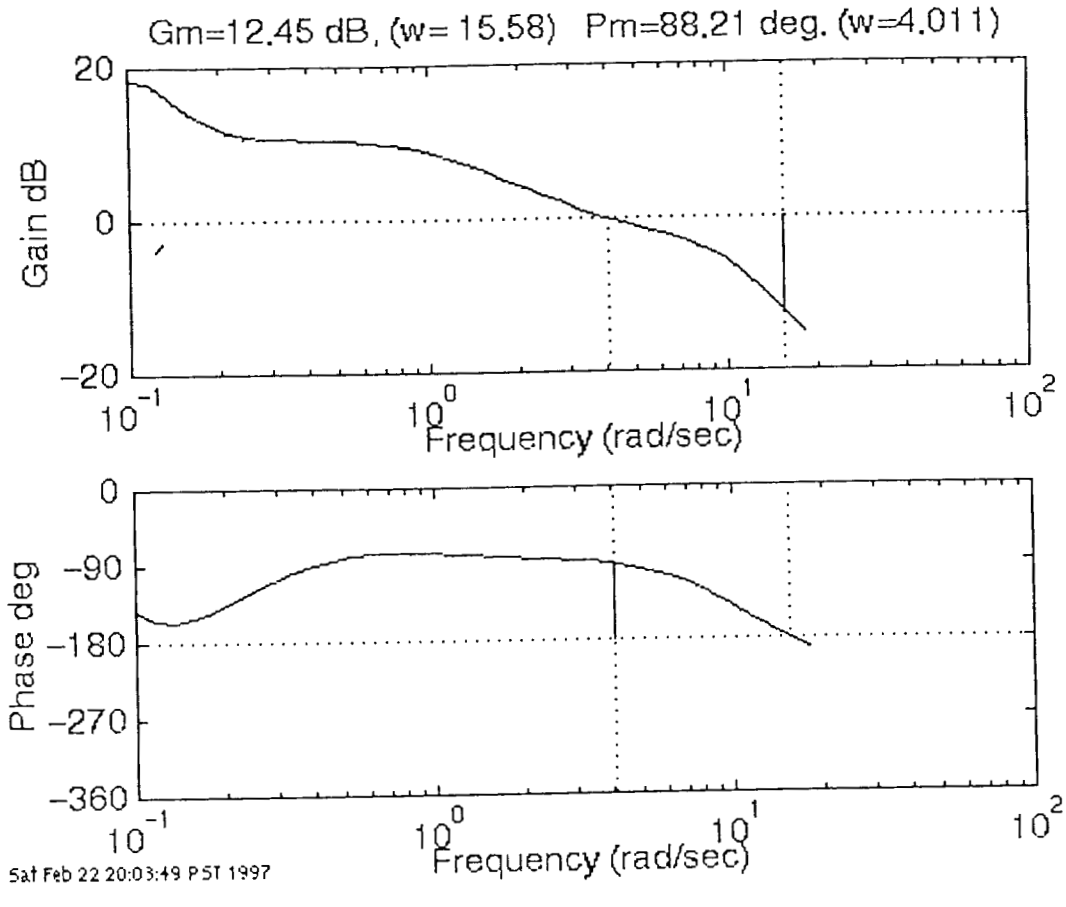


Figure 16

The broken loop frequency response for the roll channel for the Boeing solution.

actuator energy. After running CONDUIT for 44 iterations a solution was reached. Figure 17 depicts this solution. CONDUIT was able to satisfy all the handling quality requirements and drive down the crossover frequency. The design parameters for the initial and final case are given in Table 1. The increase in the model design parameters allowed the control system to meet all the handling quality specifications. The most noticeable increase is in the roll and yaw model design parameters (Mphi and Mpsi) so the quickness and bandwidth specifications could be met.

Design Parameters	Initial	Final	Design Parameters	Initial	Final
Ktheta	13.6	12.065	Mtheta	2	2.470
Kphi	8	8.488	Mphi	2.5	4.600
Kpsi	7.6	8.348	Mpsi	2	4.134
Kq	6.4	6.418	KItheta	0	1.000
Kp	2.4	.2809	KIphi	0	2.328
Kr	3.2	3.068	KIpsi	0	1.088

Table 1. The design parameters taken from Tischler (Ref. 9) paper and CONDUIT solution.

Figure 18 shows the supporting plot for the yaw bandwidth specification. The increase in the model parameter has shifted the phase curve causing the crossing of -135 degrees at 3 rad/s. The roll quickness has also improved due to the increase in the model design parameter. Figure 19 shows the supporting plots that justify this response. Also it can be noticed that the angle and rate responses are clean and don't possess any unwanted oscillations. The last item to notice is the reduction in phase margin in the roll channel attributed to a large decrease in Kp with the increase of KIphi. The significant reduction

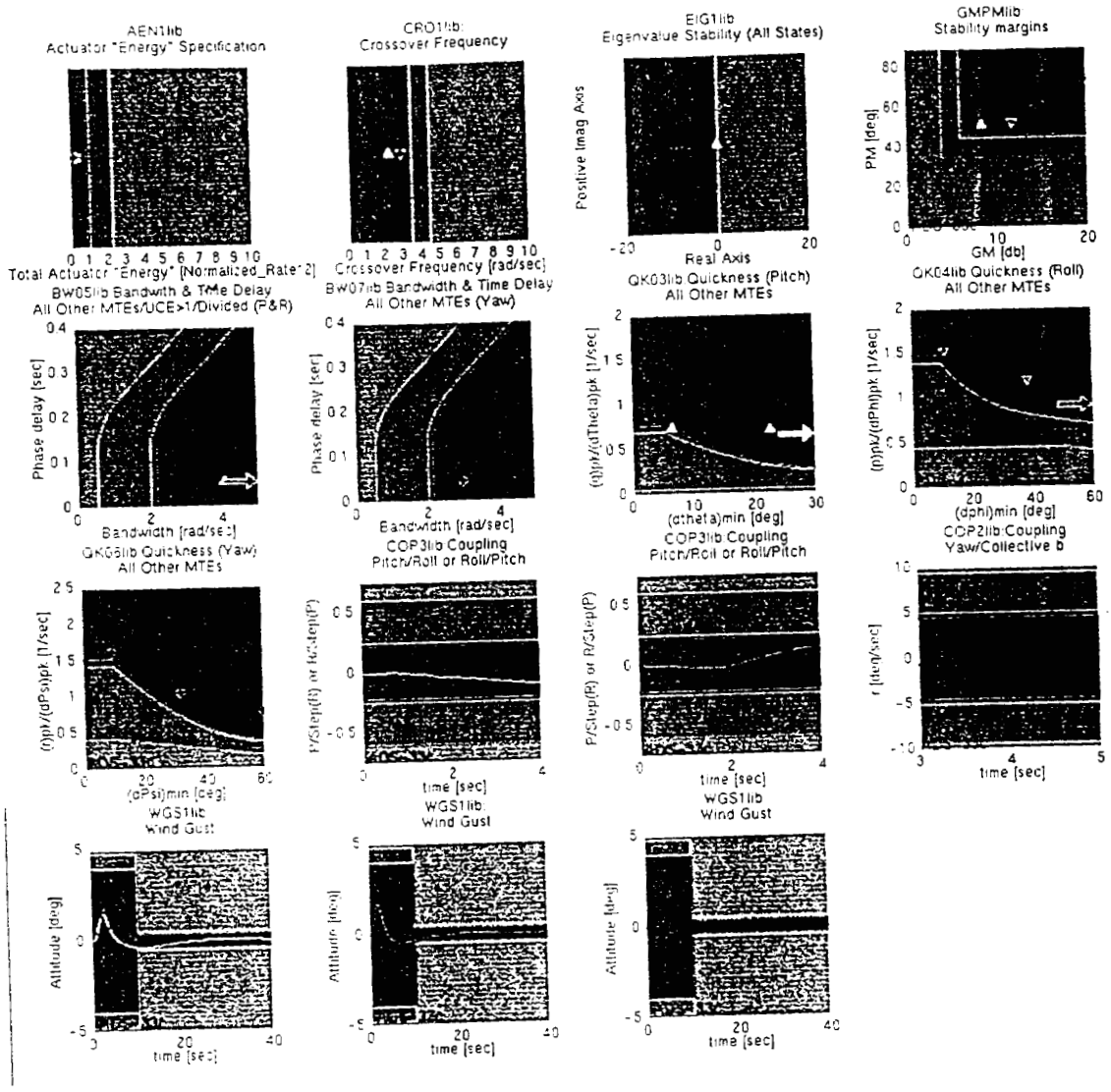


Figure 17

ADS-33 solution after tuned within the CONDUIT environment.

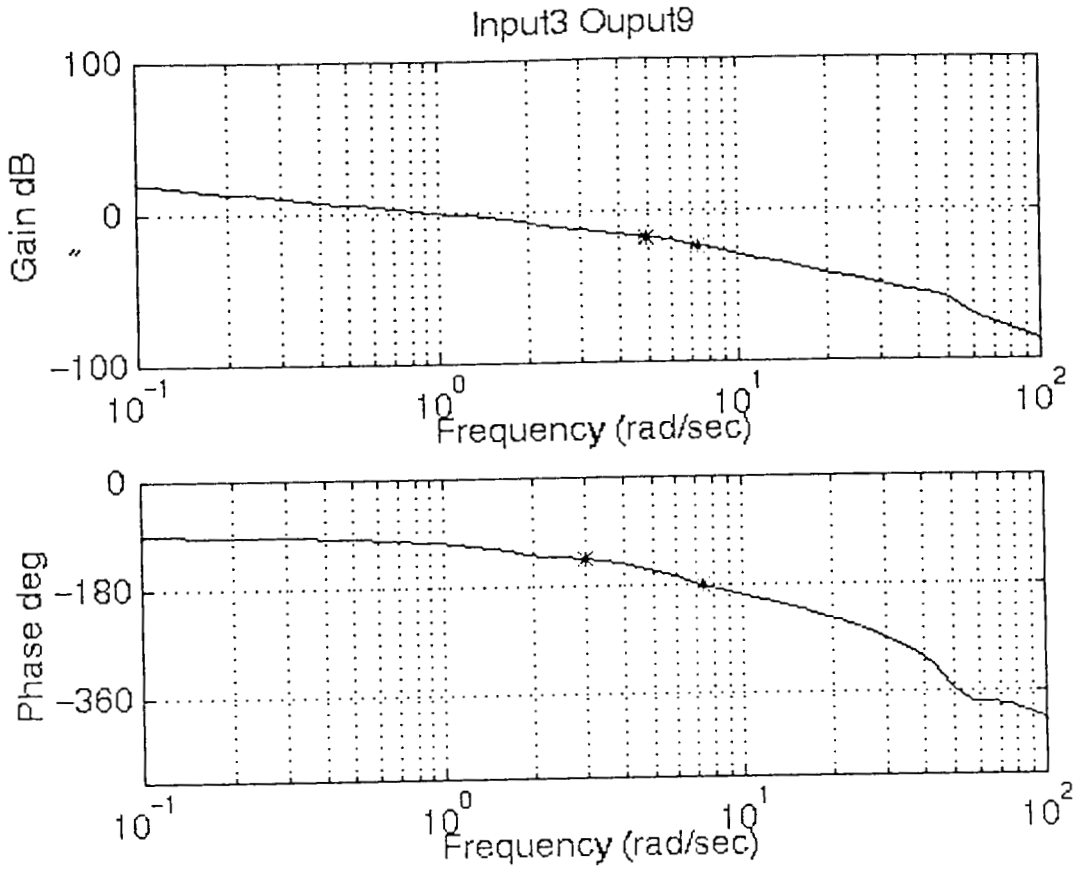


Figure 18

Yaw bandwidth supporting plots.

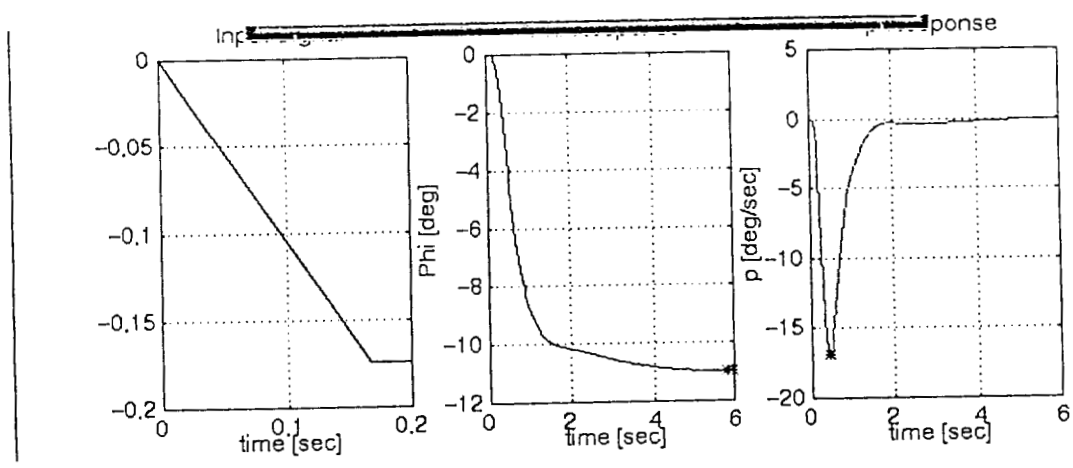


Figure 19

The supporting plots for the roll quickness specification for small input signal once it has achieved Level I.

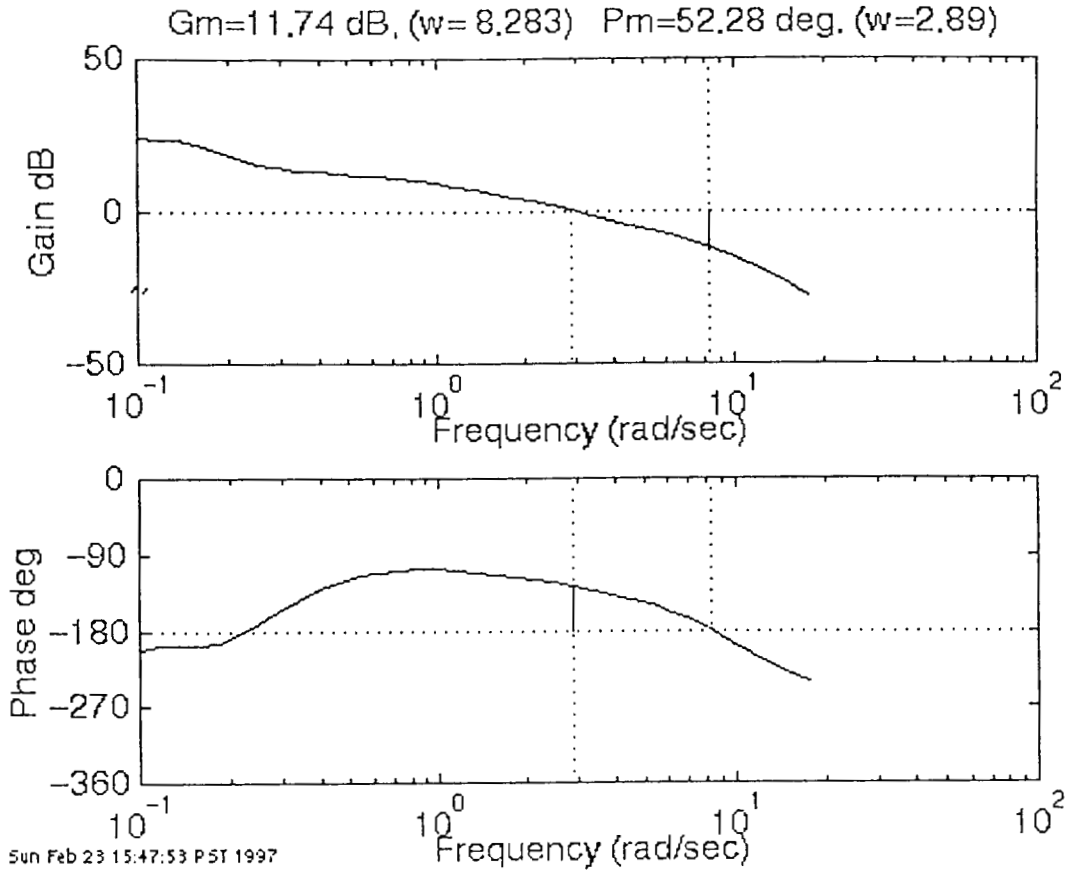


Figure 20

Roll channel gain and phase margin supporting plot after tuning in CONDUIT environment.

in crossover frequency in the roll channel can certainly be tied into the reduction of the phase margin. The supporting plot for the roll channel gain and phase margin in figure 20 can be compared to the previous case (figure 16). The reduction in Kp has had this effect in the design.

With the CONDUIT system a single engineer has tuned the control laws to meet ADS-33 handling quality specs and reduced the crossover frequency to provide a better design. CONDUIT has produced a design that allows the single engineer to see the handling qualities as well as the frequency and time response data. The design could be completed in a day once a given architecture and list of specifications have been chosen. The engineer can either alter the design parameters by hand or allow the optimization engine drive a solution that will end on the design margin.

5.3 Model Following Design for RASCAL

The model following design used by RASCAL uses a second order model that represents desired aircraft performance that is believed to meet the specifications. The use of high gains in the feedback loops is to remove any discrepancies between the actual dynamics and the inverse approximation of the dynamics. Still in this method there is no expedient way for checking the model against a set of handling quality specs.

The idea behind model following is to implement a model that will represent the desired system dynamics and have the existing system be canceled out by an approximation of the inverse of the present system. The goal of this solution is to achieve a set of gains that would allow the control system to behave like the model and have the existing plant be canceled out. The handling qualities that measure the end-to-end response aren't the focus of this solution, but rather the ability to track a given model.

The purpose of this solution is to demonstrate that a model can be implemented and the system dynamics can be effectively canceled out by producing an inverse approximation of the system and allowing feedback to compensate for the discrepancies. The job of CONDUIT is to find a set of gains that allows the end-to-end response to closely match the model's response. With the system essentially reducing to M, a model of the desired performance can be placed in the system. Referring to figure 10; equation 11 shows that if P^{-1} is exactly equal to the inverse of the aircraft dynamics then the resulting system will be the model (M). The gains in the H block help minimize the difference between P and P^{-1}.

$$M[P^{-1}+H][P/1+PH] \quad \text{Eqn 11}$$

$$M(P^{-1}P+PH)/(1+PH)$$

$$M(1+PH)/(1+PH)$$

$$M$$

Perfect model following cannot be achieved in reality because the high frequency dynamics cannot be fully inverted in P^{-1}. The high frequency dynamics can be approximated by an equivalent time delay, τ. To maintain good model following, this equivalent delay must be added to the feed forward loop (Table 2). However this penalized the roll channel too severely, thus the delay is left from the roll channel so it behaves as lead (Ref. 4). Also matching the end-to-end response of the aircraft with the model is impossible since the highest feedback gains still won't get the high frequency phase curves to match. The best that can hoped for is to match the model time the plant delay **M*D**.

Channel	Filter (ms)	Actuator (ms)	Helicopter Dynamics (ms)	Total Delay (ms)
Pitch	25	20	29.7	~75
Roll	25	20	26.8	~75
Yaw	25	20	22.8	~75

Table 2. Delay parameters.

The model used for the model block was taken from Tischler's paper (Ref. 9). It is their model (equations 12, equations 13, and equations 14) being implemented in this design.

$$\theta_c/\theta_m = \frac{4}{(s+2)^2} \qquad \text{Eqn. 12}$$

$$\phi_c/\phi_m = \frac{6.25}{(s+2.5)^2} \qquad \text{Eqn. 13}$$

$$\psi_c/\psi_m = \frac{2}{s(s+2)} \qquad \text{Eqn. 14}$$

This model is not guaranteed to satisfy ADS-33 handling quality requirements. The purpose of this problem is to utilize CONDUIT to track the model, and optimize for handling qualities. To prevent the optimization phase from trying to drive the model, these design parameters were frozen at the values that Boeing provided (Ref. 9). The

bandwidth and quickness specifications were turned to check only since there could be no improvement with the fixed model.

In order to enforce model following a model following specification was created. It is based on a cost function that measures the mismatch between the two curves. To measure the degree of model following mismatch between the two frequency curves the parameter cost function was utilized. The cost function is defined in equation 15 (Ref. 5).

$$Cost = \frac{20}{N}\left(\left(Mag_{\cdot end-to-end} - Mag_{\cdot model}\right)^2 + 0.01745\left(Phase_{end-to-end} - Phase_{model}\right)^2\right) \quad \text{Eqn. 15}$$

Where N is the number of data points being used.

The idea of the model following specification is to have the ability to see the difference between the two curves and show the contribution to the cost of the gain curve and the phase curve separately. The model following is shown in figure 21. The cost due to mismatch in the gain curve is the value that come from the first half of equation 15. This value is plotted on the vertical axis of the specification. The mismatch of the phase curves is calculated by second half of the cost function. This value is plotted on the horizontal axis. These two values added together equal the total cost. The spec works by limiting the total cost. Equation 16 has the cost equal the x plus the y values. Restrict the model following to a single cost then solving for y you produce the limiting splines on the model following spec.

$$Cost = y + x$$

Eqns. 16 & 17

$$y = Cost - x$$

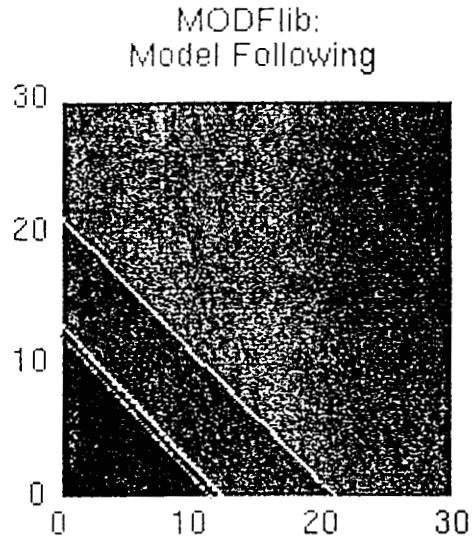


Figure 21

The model following specification.

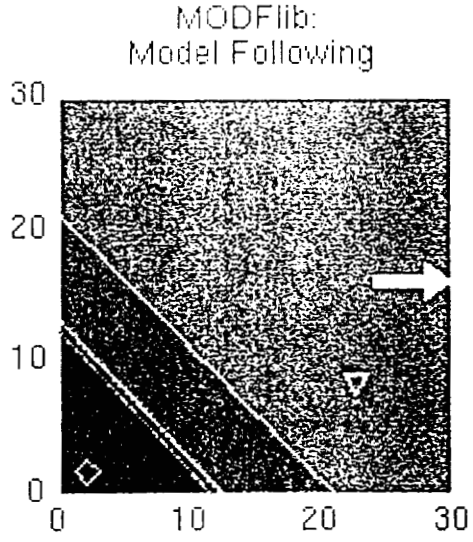


Figure 22.

Model following specification for the Boeing design parameters.

The requirements to be met are the overall stability, the gain and phase margins, the off-axis coupling, and the wind gust response. The program was setup to optimize for the cost function of the model and end-to-end mismatch, the actuator energy, and the crossover frequency. The initial case was the same as the initial case in section 5.2 (figure 13). The model following result for this case is seen in figure 22. The yaw channel has good model tracking where the other two cases have significantly less. Although the tracking at the cost functions is relatively good it would not be ideal if the model following exercise was to act as a simulator for another aircraft model. Figures 23, 24, and 25 show the mismatch between the model and end-to-end response. It became obvious that integral feedback loops would be needed to be closed around the system to ensure a high level of model following. These gains were initially set to 1. The optimization tuned them to a higher value eliminating the mismatch between the plant and inverse plant.

Design Parameters	Initial	Final	Design Parameters	Initial	Final
Ktheta	13.6	11.580	Mtheta	2	2
Kphi	8	5.513	Mphi	2.5	2.5
Kpsi	7.6	6.613	Mpsi	2	2
Kq	6.4	7.674	KItheta	0	9.032
Kp	2.4	1.361	KIphi	0	8.977
Kr	3.2	1.661	KIpsi	0	1.61027

Table 3.

Initial design parameters as provided by Tischler's paper (Ref. 9) with CONDUIT optimized parameters for the model following solution.

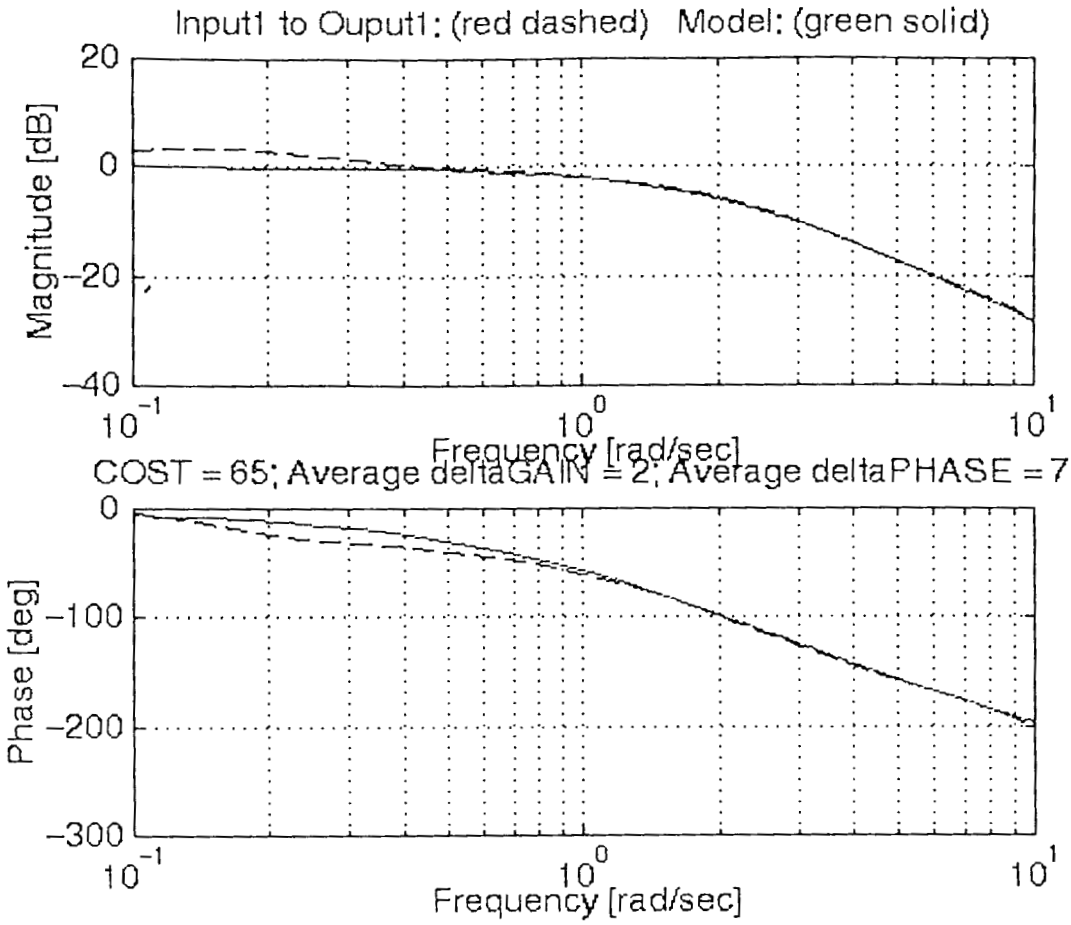


Figure 23.

Supporting plot for the model following spec (pitch).

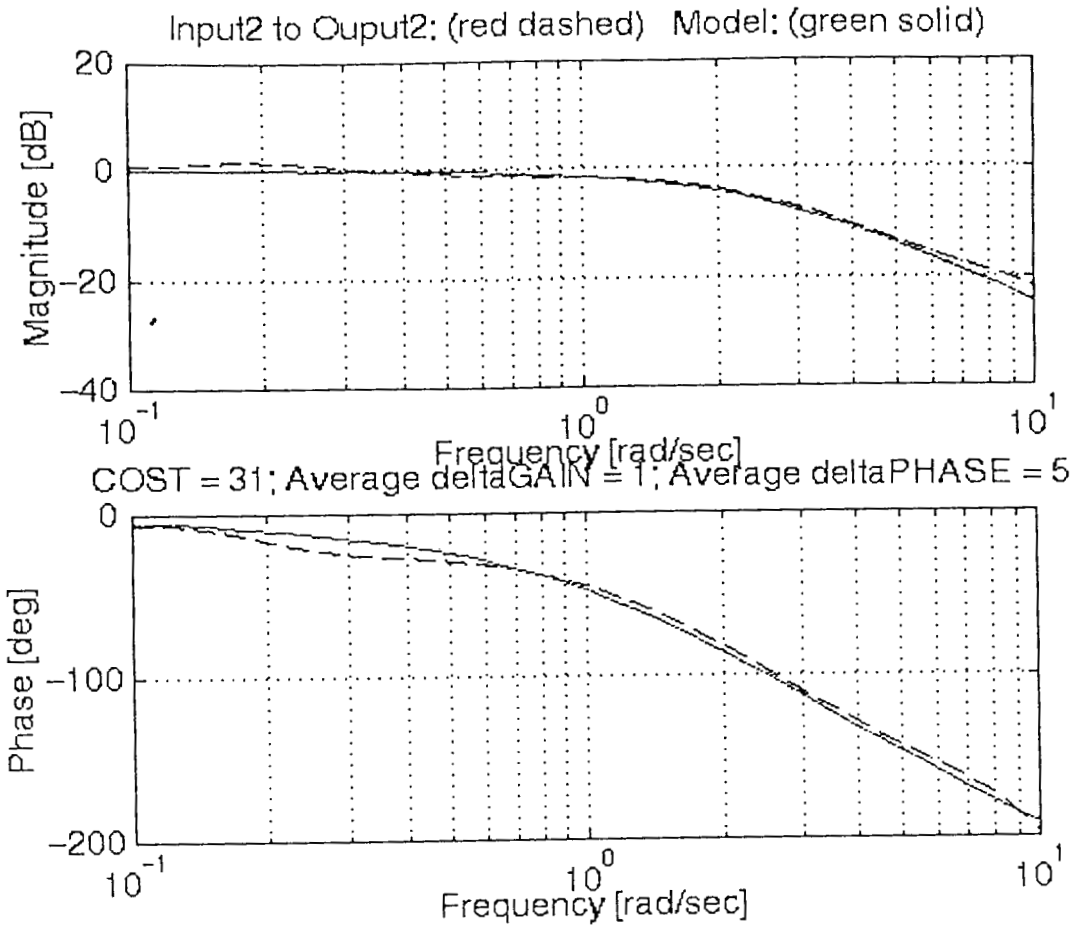


Figure 24.

Supporting plot for the model following spec (roll).

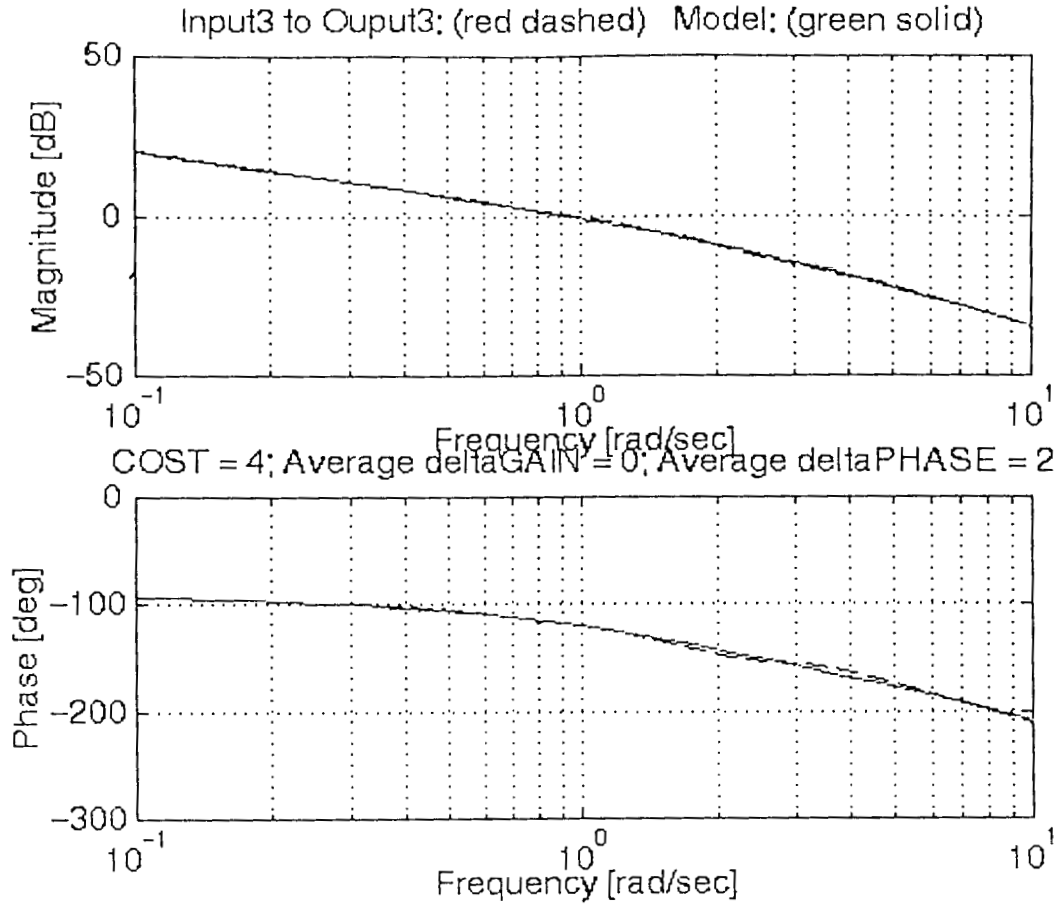


Figure 25.

Supporting plot for the model following spec (yaw).

The result was a match of the end-to-end response to the model with costs of 12, 11 and 7 for pitch roll and yaw respectively (Figure 26). This is considered an excellent match as can be seen by the supporting bode plots in Figures 27, 28, and 29. Little change occurred in the bandwidth and quickness spec since the model is essentially the same as the end-to-end although the end-to-end response has changed slightly since it tracks the model closer. A new model is required to satisfy the quickness and bandwidth specifications. We see that the bandwidth on the yaw channel falls below the requirements for quickness and fails on the roll and yaw channel. The bandwidth is the result of the model for the yaw channel. Since such good agreement has been demonstrated between the model and the end-to-end response it seems that a better model would satisfy these requirements. The design parameters show the integrator gain in the pitch and roll channels to be much higher to compensate for the model following spec. As a result the phase margins are compromised and the wind gust show more overshoot than the ADS-33 solution. The model following solution showed that the crossover frequencies were forced even lower. The limiting factor for the crossover frequencies in this optimized design seemed to be the wind gust response. With the high integrator gain in the pitch and roll the crossover frequency could be reduced in these channels without driving the wind gust spec to the edge. The yaw crossover frequency is driven down as well. This occurred because when optimizing CONSOL-OPTCAD works only on the 'worst' specification data point. Since the roll crossover frequency was allowed to be reduced this allowed the optimization engine to reduce the yaw crossover frequency. This explains why in this case the optimization was able to drive the yaw to the lowest allowable phase margin and this didn't occur in the ADS-33 solution.

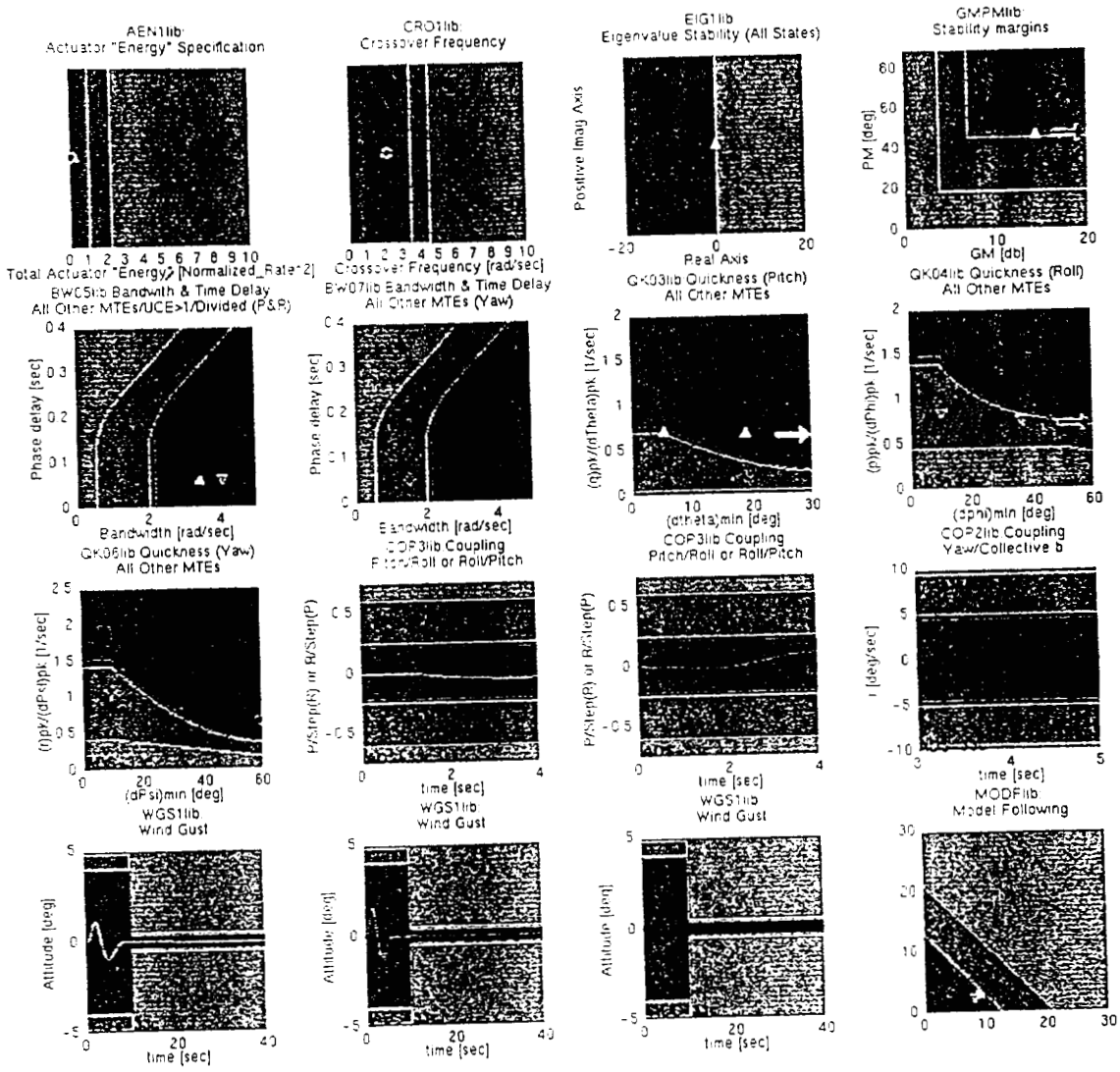


Figure 26

CONDUIT's optimized parameters for the model following solution.

		ADS-33	Model Following
Crossover frequency	Pitch	2.19	2.07
	Roll	2.89	2.05
	Yaw	2.89	2.07
Actuator energy	Pitch	0.285	0.178
	Roll	0.283	0.089
	Yaw	0.197	0.063

Table 4. The final values of the optimization constraints.

The mismatch of the phase curves as discussed in the Tischler's paper (Ref. 9) which was attributed to the improper use of delay in the feedforward system of the block diagram was shown to be corrected when the time delay was implemented.

This solution shows much lower crossover frequencies and these seem to be limited by the wind gust response. The actuator energy, previously thought to be a good reduction parameter was traded for crossover frequencies. Looking at the actuator responses for the several sets of design parameters shows that the actuator specification might not be smooth enough for optimization. The spec might need to be used as check only or have the sensitivity reduced in the optimization algorithm.

The solution was able to meet all the feedback loop specifications while driving the crossover down and the phase margin to the boundary for all three channels.

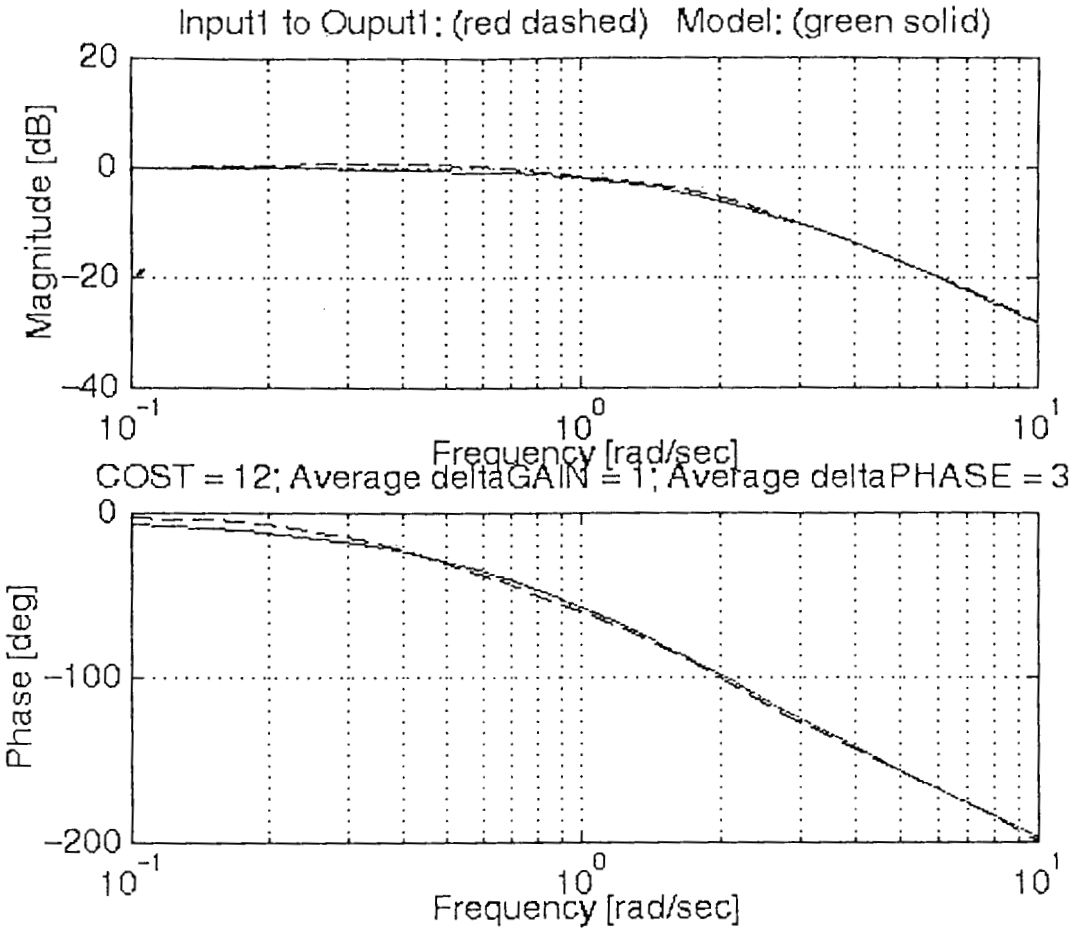


Figure 27

Supporting plot for the model following spec with optimized solution (pitch).

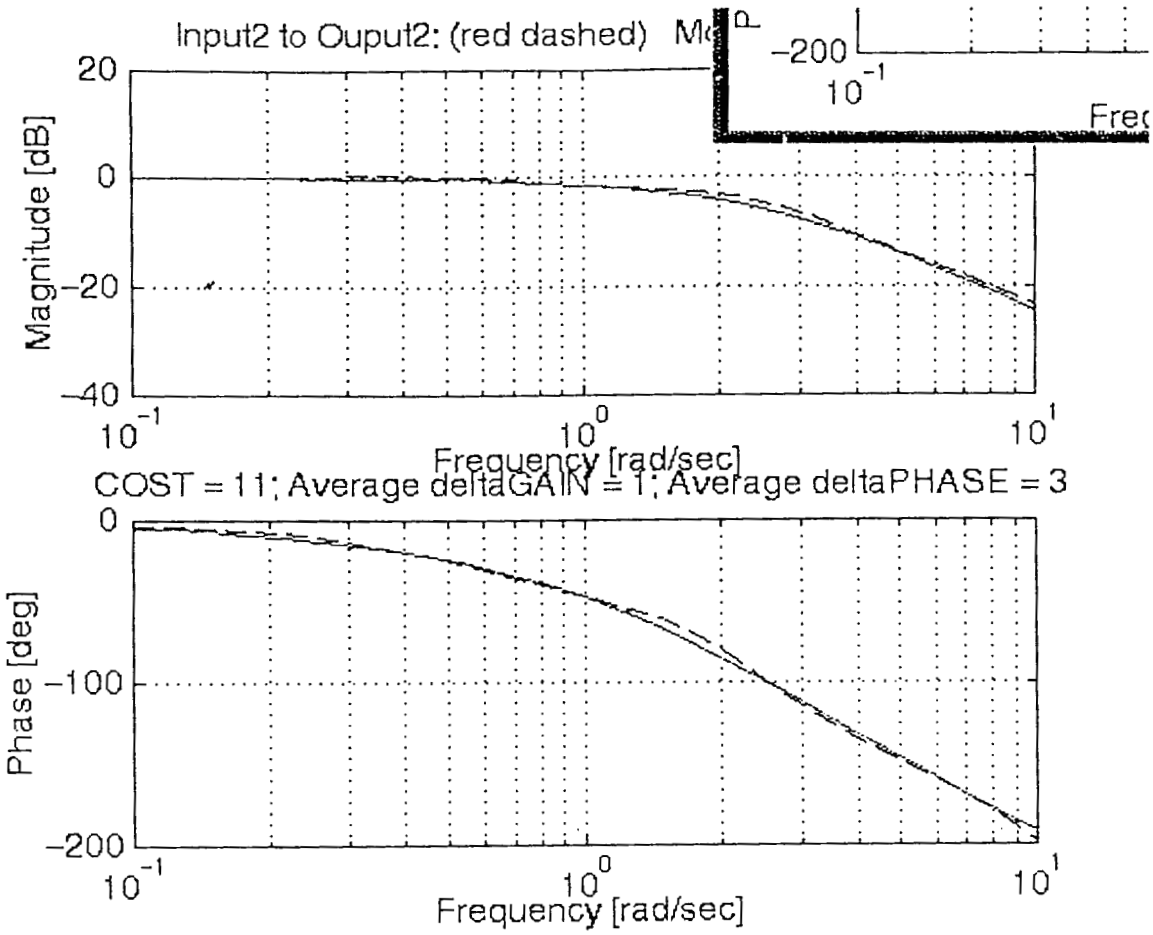


Figure 28.

Supporting plot for the model following spec with optimized solution (roll).

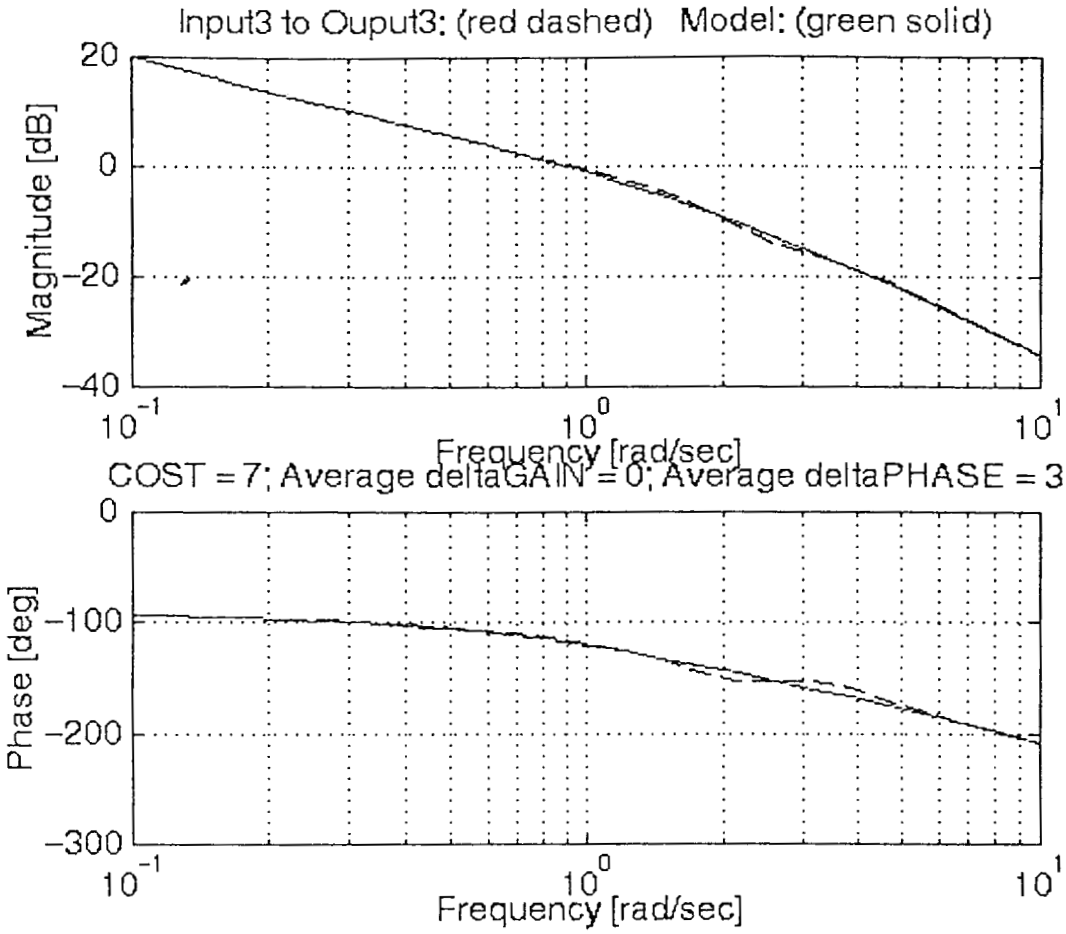


Figure 29

Supporting plot for the model following spec with optimized solution (yaw).

5.3 Windgust Rejection

This problem will be used to demonstrate the trade-off possibilities that can be conducted by an engineer using CONDUIT. CONDUIT provides the engineer the ability of to compare families of designs which can provide insight into the relationship of competing specifications. This problem was also used as an example of a simpler problem that could be used as a demonstration problem that would run quickly and still have a complex real world application. The problem statement was to design a controller that would allow the UH-60A Blackhawk helicopter to maintain it's position within a series of 10, 20, and 30 foot diameter circles while under the effect of turbulent wind for 100 seconds.

The feedback architecture consists of simple loops on the position and speed channels as seen in figure 30. The x, y, and z position vectors are calculated by feeding the integrated speed vectors u, v, and w. The position and speed are then fed back in to the pitch, roll, and collective inputs. For the small angles feeding angles into position is sufficient. The feedback gains are denoted as dp_Kx, dp_Ky and dp_Kz for the position and dp_Ku, dp_Kv and dp_Kw. For the body axis of the helicopter u and w are in opposite directions to the pitch and collective inputs this results in negative gains for the dp_Kx, dp_Ku, dp_Kw and dp_Kz gains.

Turbulence excitation was simply by adding noise into the pitch, roll and yaw channels directly before the aircraft dynamics. The wind model being used is placed into the B matrix of the 14 state space model pitch, roll and yaw channels directly.

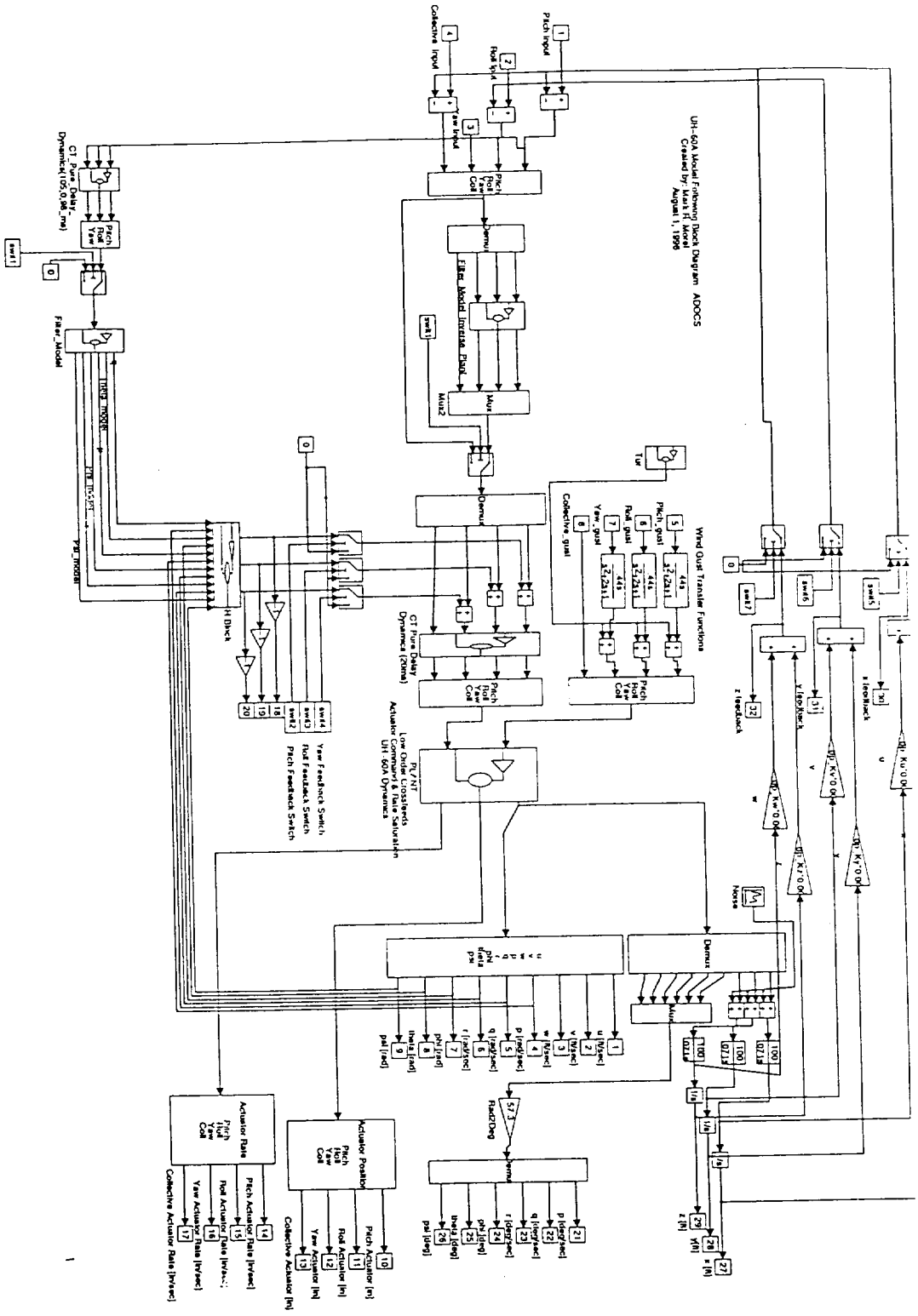


Figure 30

Position and velocity feedback loops for position hold example. Includes sensor noise and turbulence models.

Measurement noise was added to the feedback loops in order to simulate sensor error. This became crucial since actuator energy is driven up as crossover frequency increases allowing higher frequency noise to affect the actuators.

The actuator energy spec is the same one that is used in the previous examples. It is determined as the square of the actuator rate normalized to the maximum rate summed over time. However a new spec that would maintain the helicopter within the boundaries of the circle needed to be created. This specification is shown in Figure 31. It consists of two axis and a curve spline. The quarter circle radius contains the boundaries for the helicopter's RMS radius during the gust to that point. The RMS radius is used over maximum radius because it is more continuous. The spec was set up in this case with the yellow triangle representing the x position of the helicopter along the horizontal axis and the y position of the helicopter along the vertical axis. The y position and the z position are represented by the inverted green triangle. That gives the radius of the circle. The axis used were x and y position of the helicopter for the first point and y and z position of the helicopter for the second point.

The trade-off that would be demonstrated was actuator energy versus the RMS of the position within the series of three circles. The specs used in this example are crossover frequency, gain and phase margin, actuator energy and position hold. No design margin was used since the wind gust was an approximation and not based on a reliable wind models, this being a demonstration model the design would have no real meaning other than forcing our design to be within slightly smaller circles.

For deciding on initial gains classical methods were used once the broken loop frequency responses were taken and then the location of 45 degrees of phase margin was

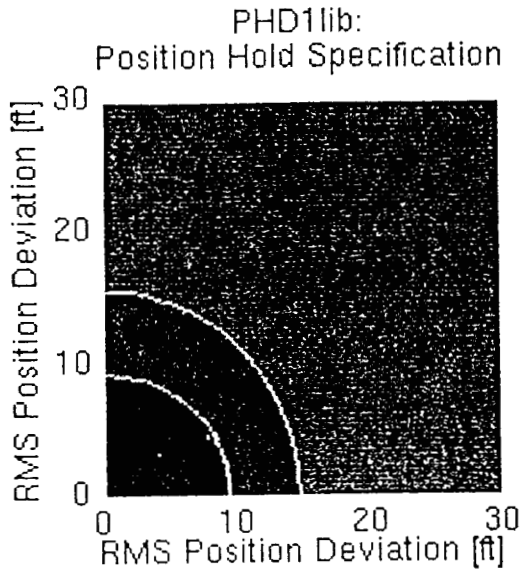


Figure 31

The position hold specification.

read along with the required crossover frequency. Gains were found in order to drop the frequency response to the required crossover frequency.

The optimization scheme was set up to drive the RMS position within the radius of the three curves. The initial solution has the RMS value of the on the Level I curve of a 10 ft radius (Figure 32). This is considered the solution for the minimum RMS value. The actuator energy is the highest in the roll channel indicating that the wind has the most effect on the helicopter in the rolling and lateral directions. This is supported since the moment of inertia is the least in this axis. The supporting plot to the position hold specification is a time history of the helicopter's x and y position (Figure 33). Although the helicopter moves outside the 10 ft circle the RMS value is within the 10 ft. circle. The trade-off of holding the helicopter in a small circle is the actuator energy. Since the roll energy was so high this will be looked at in comparison with RMS position. Figure 34 shows the supporting plots for the actuator. The actuator rate is used to determine the energy used. The actuator rate is the bottom left graph in figure 34. The maximum rate for this actuator is 10 in/sec. The figure shows us that the actuator saturates frequently during the simulation hitting values of 10 several times. CONDUIT will try and drive the actuator energy down for the roll channel down with the expected result of a larger RMS value for the position of the helicopter.

The position hold specification was relaxed to allow greater movement as shown in figure 35. Figure 36 shows the actuator energy after CONDUIT has driven it down. The RMS value now lies at about the 14 ft radius. The time history of the helicopter shows the increase in movement in the y direction (figure 37). This increase in RMS value has allowed the actuator rate to reduce to a maximum value of 8.5 in/sec. The trade

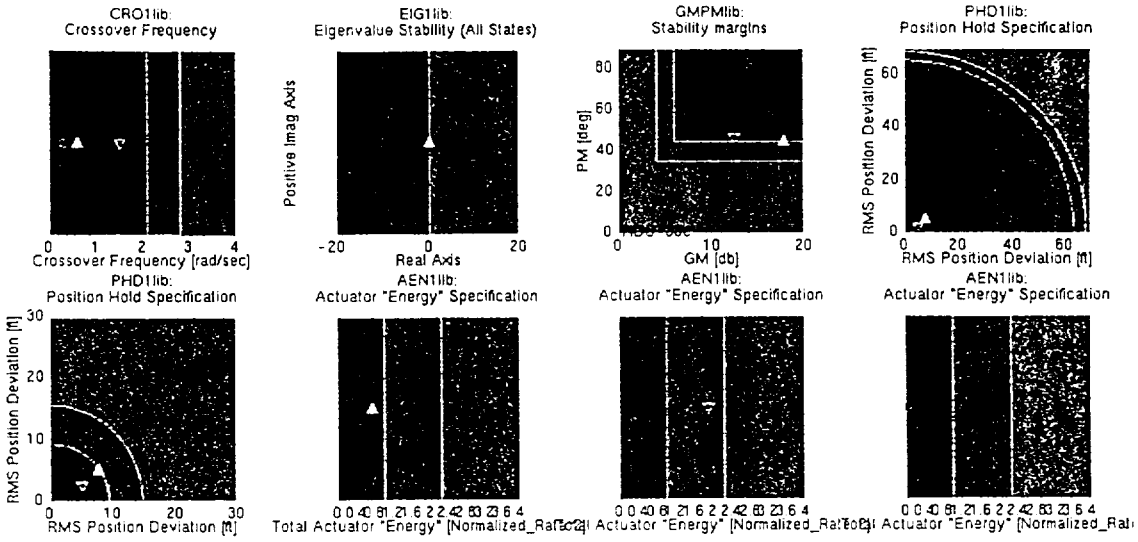


Figure 32

Windgust solution for minimized RMS position.

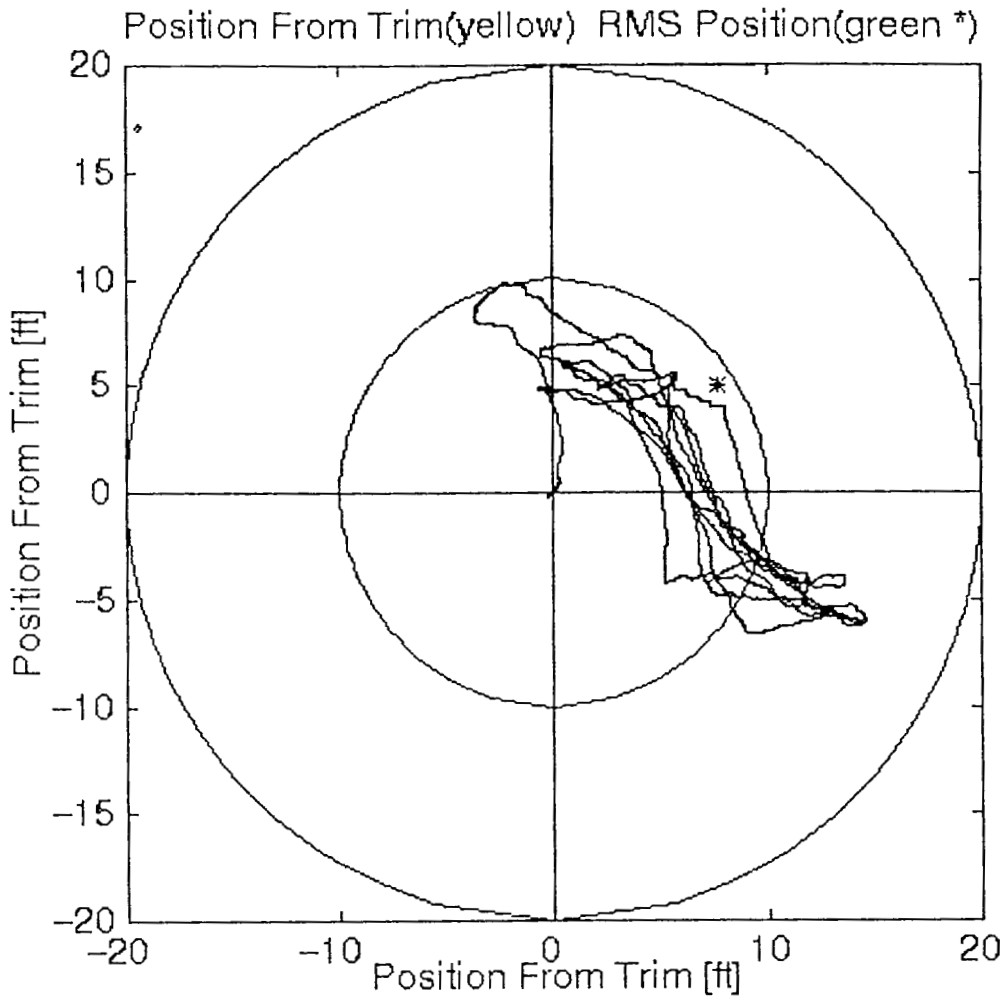


Figure 33.

Time history plot of the helicopter's position for minimized RMS position.

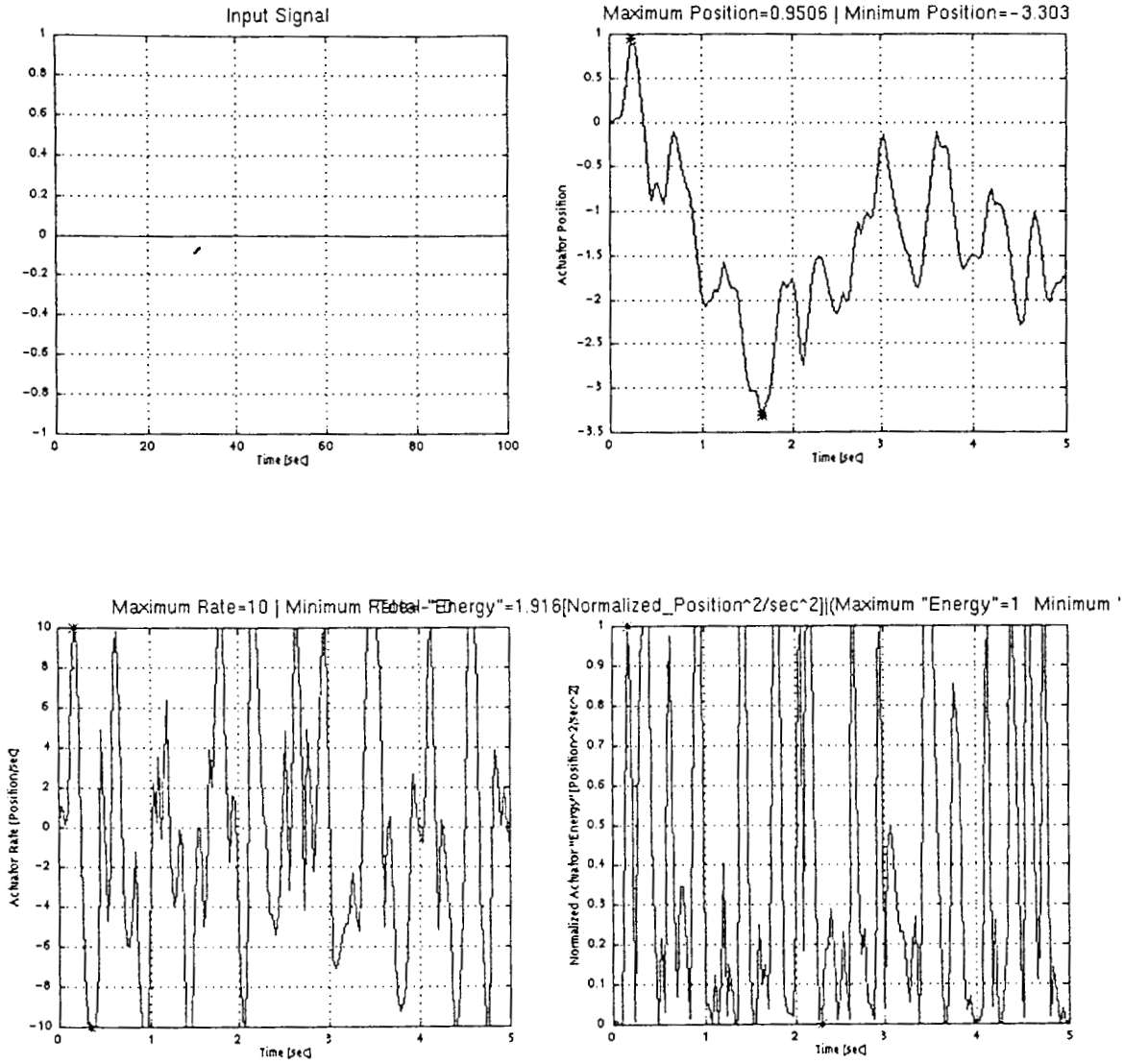


Figure 34

Supporting plot for actuator energy (roll) with saturation in the actuator rate.

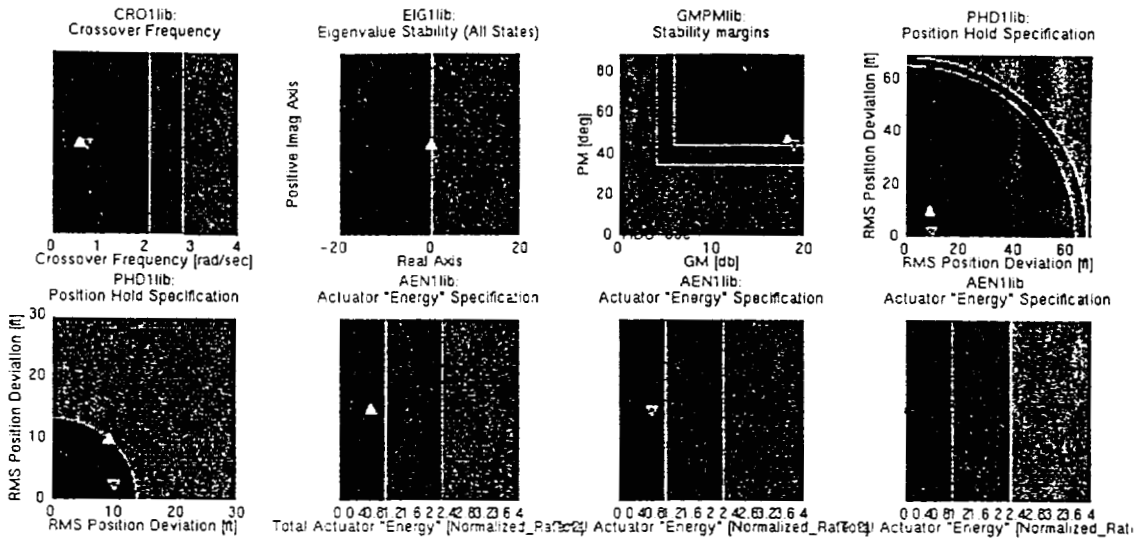


Figure 35

Medium RMS position hold solution.

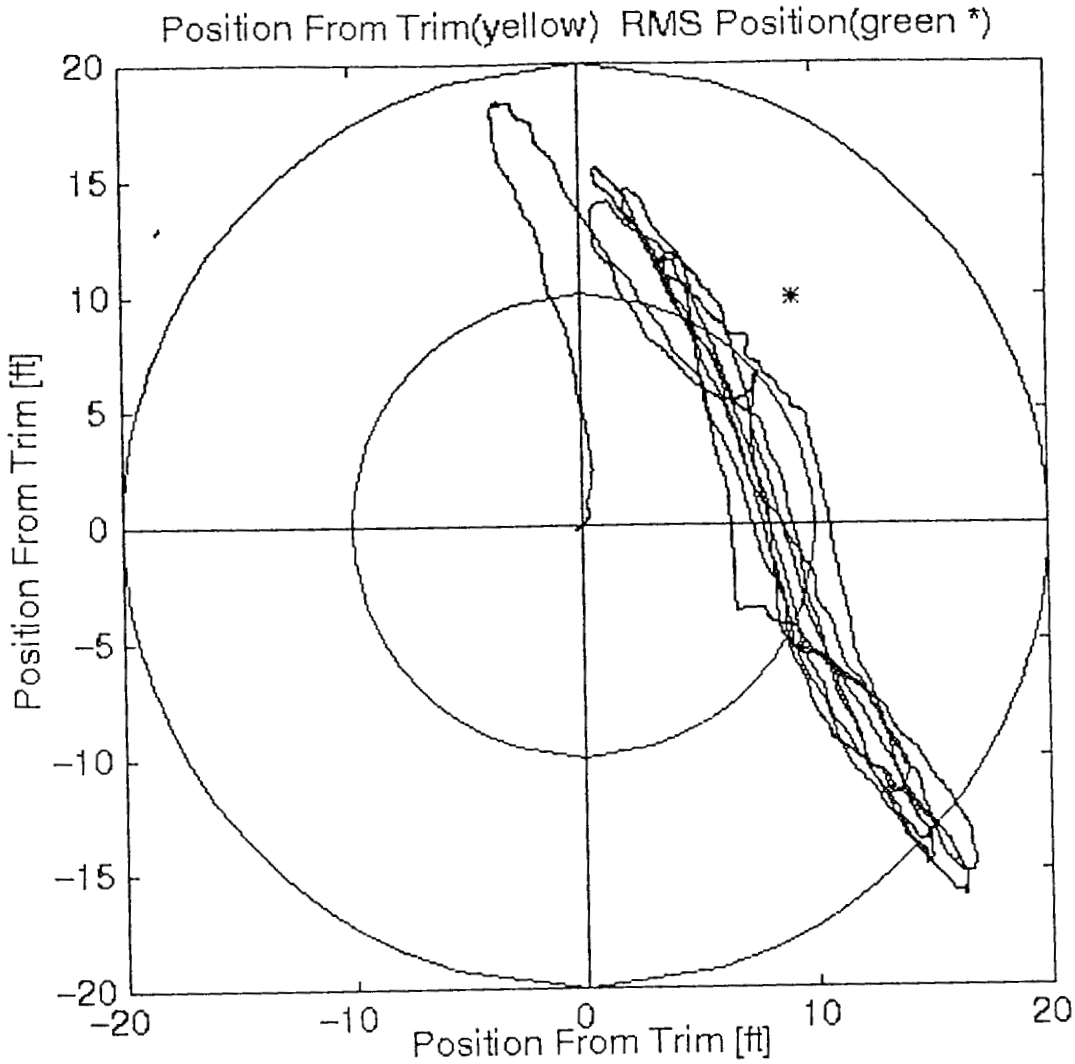


Figure 36

Time history plot of the helicopter's position for medium RMS position.

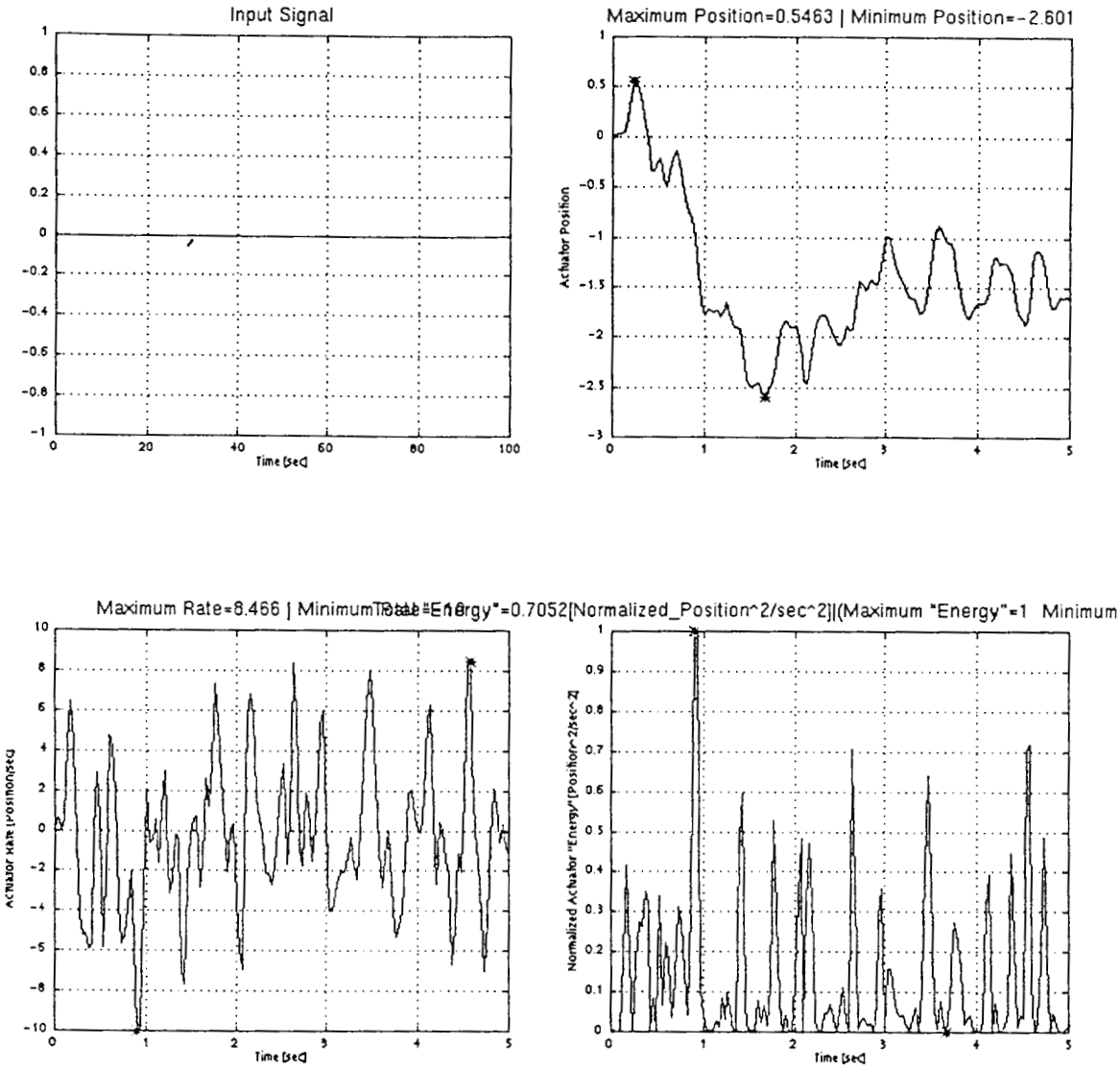


Figure 37

Supporting plot for actuator energy (roll) with saturation in the actuator rate.

off is becoming obvious, actuator energy for position hold. After reducing the actuator energy further (figure 38) the RMS value of the helicopter's position is near 27 ft. The supporting plot shows the helicopter approaching 60 ft in the y direction (figure 39). The actuator rates again decreased as seen in figure 40 to a level of 5.9 in/sec. The complimentary nature between the actuator energy and the crossover frequency can be noticed as they both decreased during the optimizations. The lower crossover frequency acts as a filter screening out higher frequencies of sensor noise which drive the actuators excessively.

The large difference between the RMS value and the maximum value, seen in the last case, will need to be investigated in order to determine an algorithm that might properly contain the helicopter within a specified radius. However, for the purpose of this example the correct trade-off has been demonstrated to show the trade-off between the actuator energy (and rate) and the position held by the helicopter (Figure 41). These trade-offs allow the engineer the ability to make decisions about actuator performance for required tasks. Engineers have the option to investigate whether or not the most costly, faster actuator is required without having to rework the control system parameters.

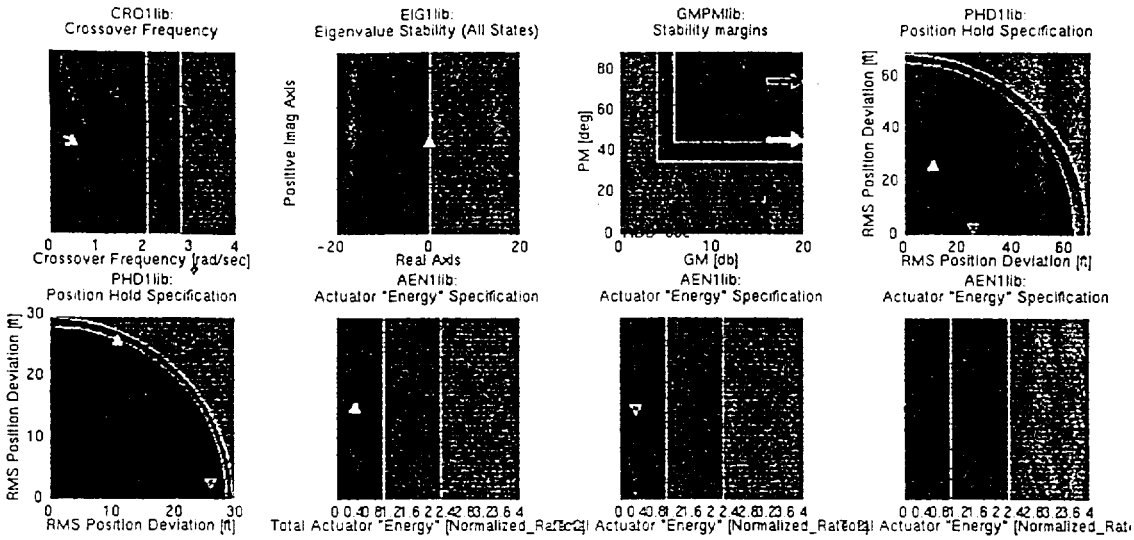


Figure 38

Large RMS position (minimized roll actuator energy)

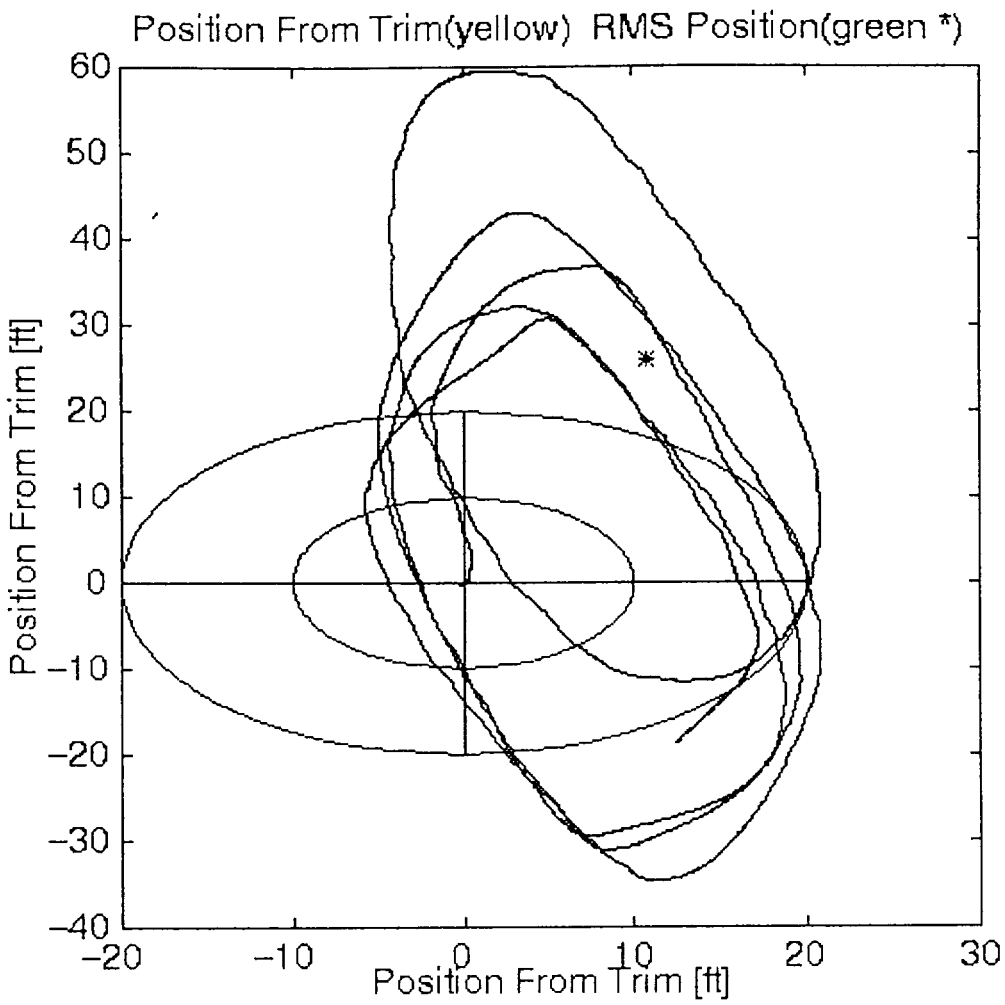


Figure 39

Time history plot of the helicopter's position for large RMS position.

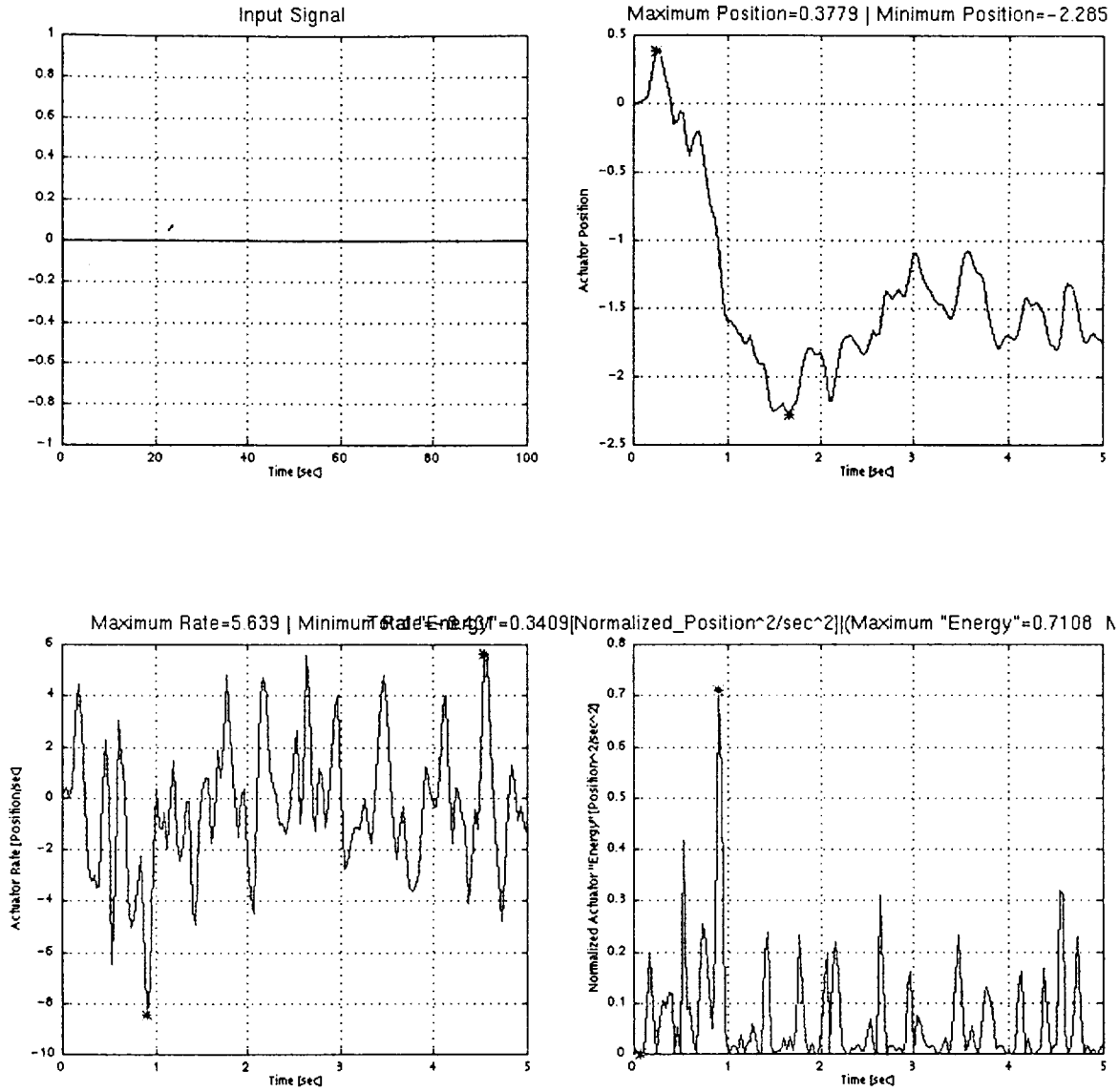


Figure 40

Supporting plot for actuator energy (roll) with saturation in the actuator rate.

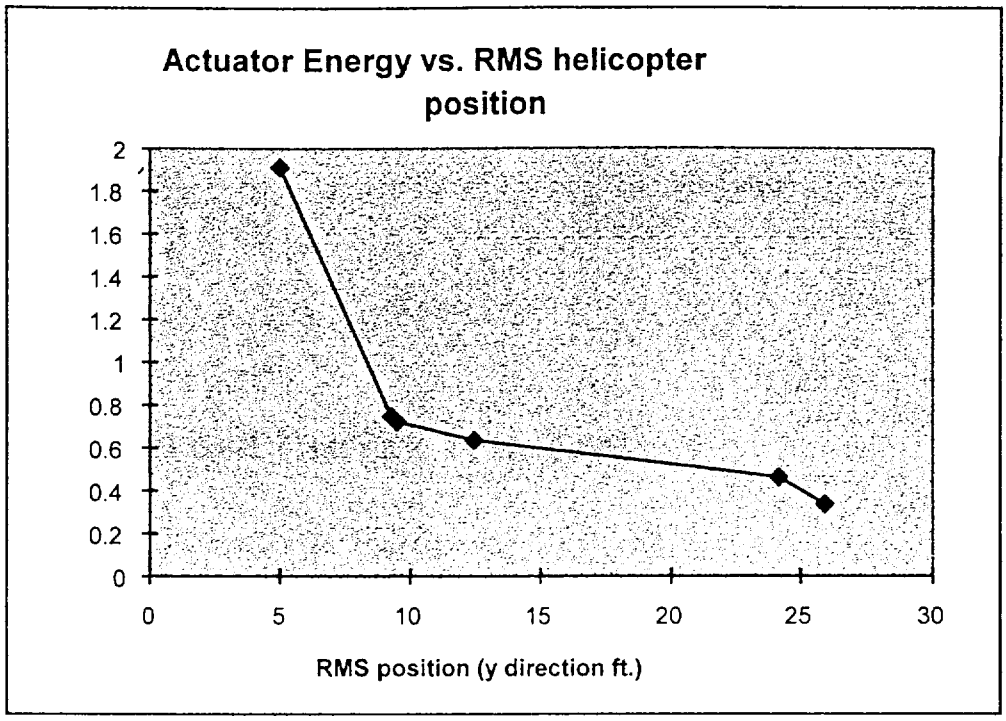


Figure 41

The trade-off between hover hold and actuator energy.

Chapter 6.0

Lessons Learned During This Thesis

6.1 Additional information learned during the thesis

During the course of my research grant at NASA Ames I was able to participate in the development of a real world application and actual helicopter control system design. Before coming to work at Ames I had no experience with helicopters. The opportunity at Ames educated me to the basics of helicopters and how they are a unique and amazing aircraft with properties not found in fixed-wing vehicles. I was able to develop skills in classical control design as well as, optimization, handling qualities analysis especially for rotorcraft. I learned project development skills working with team members at Ames, the University of Maryland, and Cal Poly.

Chapter 7.0

Conclusions

7.1 Conclusions

The development of CONDUIT has produced a powerful tool for control system design and analysis. It provides an interface that allows for the design of control systems that satisfy handling qualities. CONDUIT calculates a set of specifications, that typically took weeks, within a few minutes. This power allows for a more detailed analysis of control systems and allows the user to experiment with different controllers within a short period of time. It succeeded in being flexible to use any control architecture that the engineer desired. This power and flexibility placed into a GUI environment that allows for simple and rapid setup procedures combined with detailed analysis that provides the user with information that supports the results all with simple mouse commands.

The three examples demonstrated the power of CONDUIT as well as illustrating the point that this is a tool for the engineer with the knowledge of control system design to work it properly. The design problems showed how quickly it could be demonstrated that the controller wouldn't meet the specifications. It showed how tuning the design could find a solution that could meet the handling quality specifications.

Like any new program or tool there are limitations to CONDUIT. Several specs have been encountered that aren't based on a continuous function and stall the system

when trying to optimize against these specs. Although they can be evaluated in CONDUIT a substitute specification will be needed to be used to enforce the other specification during tuning. Another limit to the optimization process is the minimization of objectives is limited to the largest value of any of the objective functions. The one area that isn't so easy to use is creating new specs. Although they are written in a straightforward manner a knowledge of MATLAB is required. Work is being done on make new design simpler.

7.2 Discussion of Future Work

Currently work is being done on CONDUIT to incorporate LOES Lower Order Equivalent Systems specifications. These are a popular subset of fixed wing specifications that fit a lower order model to a high order frequency response curve from which dynamic properties can be determined. Work on this is being performed by Mark Morel, another Cal Poly graduate student. Mark is currently working on a control system design for the X-29 as a part of his master's thesis. Demonstrations of CONDUIT to industry have created a lot of interest in the program.

Future work will focus on improving the optimization package, developing tools that allow the user to develop new specs in a easy-to-use environment. Plans are to integrate CONDUIT into a design cycle that will allow the designs produced on CONDUIT to be integrated into simulation environments and actual aircraft controllers with limited conversion and reprogramming of the control system design.

More rigorous testing of the ADS control laws needs to be performed before the model following control system arrives this fall for the RASCAL helicopter.

References

6) Aiken, E., Jacobsen, R., Eshow, M., Hindson, W., and Doane, D., "Preliminary Design Features of the RASCAL--A NASA/Army Rotorcraft In-Flight Simulator," AIAA-92-4175, Aug 1992.

2) Fan, M. K. H., Tits, A. L., Zhou, J. L., Wang L. S., and Koninckx, J., "CONSOL OPTCAD, User's Manual. Technical Report 87-212r2a, Department of Electrical Engineering and Institute for Systems Research, University of Maryland, College Park, MD, 1991.

7) Fletcher, J. W., "Identification of UH-60 Stability Derivative Models in Hover from Flight Test Data," J. American Helicopter Society, vol. 40, no. 1, 1995, pp. 32-46.

10) Glusman, S. I., Conover, H. W., and Black, T. M., "Advanced Digital Optical Control System (ADOCS) Volume III - Handling Qualities." USAAVSCOM TR 90-D-11C, Nov 1990.

11) Hoh, R. H., Mitchell, D. G., Ashkenas, I. L., Klein, R. H., Heffley, R. K., and Hodgkinson, J., "Proposed MIL Standard and Handbook -- Flying Qualities of Air Vehicles, Vol. II: Proposed MIL Handbook," AFWAL-TR-82_3081, 1982.

1) MATLAB®: High-Performance Numeric Computation and Visualization Software. The Math Works, Inc., Reference Guide, 1992.

8) Takahashi, M. D., " Flight Control Design Using Rotor-State Feedback for an Articulated Rotor Helicopter in Hover. NASA Technical Memorandum 103967, Jan 1993.

5) Tischler, B. Mark., Advances in Aircraft Flight Control, Taylor & Francis Ltd., Bristol, PA, 1996.

9) Tischler, M. B., "System Identification Requirements for High-Bandwidth Rotorcraft Flight Control System Design", J. Guid., Control., and Dyn., vol. 13, no. 5, 1990, pp. 835-841.

4) United States Department of Defense, "Flying Qualities of Piloted Vehicles", MIL-STD-1797, March 1987.

3) United States Army Aviation Systems Command, "Handling Qualities Requirements for Military Rotorcraft", ADS-33C, ST. Louis, MO., August, 1989.