NASA/TM–1999-208763
AFDD/TR-99-A-005

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

Jason Colbourne
California Polytechnic University
San Luis Obispo, California
Aeroflightdynamics Directorate
U.S. Army Aviation and Missile Command
Ames Research Center, Moffett Field, California

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

March 1999

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

CONTENTS

LIST OF TABLES

LIST OF FIGURES

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

JASON COLBOURNE*

Aeroflightdynamics Directorate, U.S. Army Aviation and Missile Command,
Ames Research Center

SUMMARY

The fields of control law design and handling quality analysis are complementary areas of work, yet are often separated in practice. The desire to integrate the fields of handling qualities and flight control systems led to the development of the Control Designer's and Unified Interface (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 set of handling quality specifications. CONDUIT allows the designer to retain a preferred control law structure, then tunes the system paranieters to meet the handling quality requirements.

This technical memorandum focuses on the development of CONDUIT and describes the philosophy, components, and capability of this tool. CONDUIT is designed to allow for rapid problem setup and easy evaluation of complex handling quality specifications. The system allows an engineer to design any control law architecture and provide rapid analysis and tuning of that design. Use of the Mathwork's SIMULINK block diagram simulation environment allows industries to work in a design mode that is easy and familiar. CONDUIT's visual environment allows for rapid setup and significant insight into the performance of the control system.

Demonstration of CONDUIT is accomplished with three UH-60A helicopter design problems. The first problem shows a design that focuses on meeting ADS-33 (ref. 1) handling qualities in hover. The second design problem tunes a model following control system while maintaining as many ADS-33 requirements as the chosen model would permit. And a third design problem implements a wind gust rejection control law as well as a wind gust rejection specification. This last problem illustrates CONDUIT's ability to perform trade-off studies by demonstrating the level of wind gust rejection versus actuator authority.

By incorporating CONDUIT in the initial phases of aircraft design, manufacturers can develop improved controllers that are designed to pilot requirements and can understand the true handling of the aircraft in the earlier phases of design. CONDUIT will aid in a more efficient, less costly design process.

* California Polytechnic University, San Luis Obispo, CA 93407.

1.0 INTRODUCTION

Handling qualities analysis and control law design would seem to be naturally complementing 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 into 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.

Designing aircraft control systems to meet the formidable set of handling quality specifications as contained in the MIL-STD-1797A (ref. 2) or ADS-33 (ref. 1) 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. Developing trade-offs that satisfy several requirements is a main aspect of the engineering discipline. The area of handling qualities is rooted in trade-offs. However, the time required to perform a complete handling qualities analysis 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 and not using handling qualities in the design phase. Only after much career experience can 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 control system architecture, and each company seems to have its own standard method that is developed through the projects that are worked. 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 industry engineers. The complex nature of multiple input multiple output (MIMO) control laws, especially in rotorcraft, which have the nature of highly coupled systems, generally prevents the engineer from seeing the trends in the control laws that would enable fine-tuning and that would ensure the "best" design is found.

These aspects of control system design and handling qualities analysis have been under study for the past several years at the Flight Control and Cockpit Integration Branch (ARH) of 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 its name from CONtrol Designer's and Unified InTerface. 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 allows the user to evaluate, compare, and tune the control system. It is designed to evaluate the architecture that is provided to the system.

This report provides a detailed outline of the features of CONDUIT, along with designs performed on the model-following control system of the UH-60A Black Hawk helicopter.

1.1 Research Objectives

The research objectives fall into two main categories: (1) development of a control system design tool that would allow rapid evaluation and design of control systems against handling quality criteria, and (2) use of this tool to evaluate and tune the baseline model-following control laws for the Rotorcraft Aircrew Systems Concepts Airborne Laboratory (RASCAL) research helicopter at Ames Research Center.

The development of the CONDUIT tool came partly out of a desire to facilitate the analysis of RASCAL handling qualities and incorporates this as part of the design process. The design of CONDUIT focuses on the philosophies of providing a visual environment with rapid and complete analysis, combined with an easy-to-use interface that allows for push button setup and evaluation. The second philosophy is a flexible design tool that allows users to create any architecture for their controller with a minimum of restrictions.

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

2.0 OVERVIEW OF CONDUIT

This section will provide an overview of the design for CONDUIT. An outline of criteria and key functions that went into shaping CONDUIT will be provided.

2.1 Statement of Purpose

This paper addresses the concerns of integrating the areas of handling qualities with control system design. Developing the CONDUIT environment and tools that provide detailed analysis of the control system is a major component of this work. A demonstration of CONDUIT's ability is provided, with 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

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 that provide the foundation for making a useful tool for control system design.

CONDUIT is laid out with three principle philosophies. The first principle is to remove the complexity of using handling qualities by automating the calculations involved. The time spent calculating a set of handling qualities specifications can be enormous and is one of the key reasons that most companies don't use the specifications for preliminary design.

The second principle stresses that CONDUIT remain flexible and not 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 evaluates the design of the control system and helps 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 work with the architectures given, so the parameters selected to tune the system indeed produce an effect on the system.

The third principle is to make the program intuitive to use. CONDUIT is a graphical environment that allows for intuitive setup and analysis. The program must be able to communicate a large amount of information with only a few key strokes. The idea is to provide information to users so they can understand the system and trust the results provided.

2.3 The Features

The entire control problem setup works in a Graphical User Interface (GUI) environment, allowing the control system engineer to perform rapid setup and analysis. Providing the user with an in-depth understanding of the 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 set up 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. 3) 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 presented in color using 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 driving the system. The specifications provide an evaluation of the control system as well as criteria that drive the optimization routines for tuning the system. Handling quality specifications (ADS-33 for Military Rotorcraft and MIL-STD 1797A for fixed wing) contain charts categorizing aircraft performance into three regions: Level 1 implies that desired performance is achievable with no more than minimal pilot workload; Level 2 allows the pilot mission to be completed with an increased workload and/or some degradation in performance; and Level 3 implies that the aircraft can be controlled but that the mission cannot be successfully completed (ref. 4). An example of ADS-33 handling quality specifications is provided in figure 1.

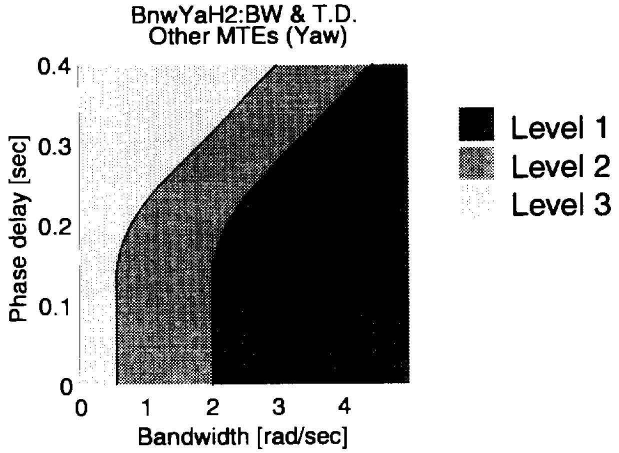


Figure 1. Example ADS-33 bandwidth specification.

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

Specifications 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 easy to perform. The axis and the boundaries can change with the mouse. This gives the user the advantage of customizing specifications and running "what if" scenarios to comprehend elements of the control law. CONDUIT gives the user the ability to relax specification boundaries that cannot be achieved with the current aircraft.

CONDUIT features supporting plots that are contained in the specifications modules. These plots 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 is time to improve the control system design. However, it is not always clear how to continue. To help the engineer tune the given control system, an optimization engine is integrated into CONDUIT. The CONSOL-OPTCAD optimization software is a Feasible Sequential Quadratic Program (FSQP) (ref. 5) 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.

3.0 DEVELOPMENT OF CONDUIT

This chapter discusses the key functions that have been included in the CONDUIT package and gives details to the work performed by the author.

3.1 Previous Work Done

The early development stages of CONDUIT began with a joint program between Ames Research Center and the University of Maryland. It evolved through many stages that incorporated key contributions from several people during its more than seven year development. A primary contribution oame from professor André Tits of the University of Maryland in the form of CONSOL-OPTCAD (ref. 5). 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 was 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 set up, 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 a 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 specifications without the user realizing this. However, the main purpose of CONDUIT—the combining of the handling quality specifications into the control system design phase—was born out of this work.

The next generation of this code was shaped by Patrick Potter (ref. 6.), Chujen Lin (ref. 6.), and Mark Lehene. Their work placed a GUI interface, named GIFCORCODE, over the work done by Gil Yudlevitch (ref. 6). 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 Yudlevitch, it didn't allow for any increased level of ease or flexibility in the problem setup. That was still in the form of a 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 Yudilevitch had set up, Potter was able to 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

The work performed in developing CONDUIT over the past year is the focus of this paper. The elements that have taken work previously completed on this project and transformed it into a releasable project are 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 work come in the areas of continuous-to-digital block diagram conversion, easily altered modular specification, rapid setup and design, a normalization algorithm, robustness in design process, and a new program environment.

3.3 Continuous-to-Digital Block Diagram Conversions

The aircraft and control systems for use in CONDUIT are modeled in SIMULINK. It is in 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 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 time response done in the digital domain. The effect of this on an optimization process requiring hundreds of time domain simulations is a severe time penalty. 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 would appear 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 chosen conversion method 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 digital conversion (digcon). 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 are seen in the discrete diagram. A conversion of the end-to-end response to the discrete domain will eliminate the nonlinear effects.

3.4 Specifications

Handling quality specifications are the driving force behind CONDUIT. They provide the unique features that allow for the control system to be designed to pilot ratings. Control system design is usually done with methods having to do with frequency response shaping. With CONDUIT the engineer can actually use handling qualities specifications 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 (ref. 1) or MIL-STD-1797A (ref. 2). Additional specifications can be created by the users to provide design criteria to investigate issues specific to the project or general criteria that enforce common requirements; for example, crossover frequency and actuator energy.

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. Two key elements necessary to keep the spirit of the design are ability for easy implementation, and what is more important, seeing what they are calculating and how they influence the design. The key elements of the specifications are discussed in the following sections.

CONDUIT contains the specifications 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 generate supporting plots and data that justify the results. The module is chosen when the user wishes 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 1, 2, and 3 regions. A key feature is the ability to move the splines so that the requirements used in the optimization can be altered quickly and visually. The mouse can move boundaries to drive certain specifications harder and relax other specifications that may be limiting the design process.

The first thing that needs to be done is turning the specifications into modules that can be added and removed according to the designer's desires. Having this done effectively requires an area from which to choose the specifications. 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 the mouse on the specifications that the user requires for the design. This new spec is then added to the handling quality window. The handling quality window is the primary work region in which the chosen specifications are displayed. 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 design philosophy. Within CONDUIT, supporting plots are 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 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 provides. The supporting plots are activated by clicking on a spec after results have appeared on the specifications. A supporting plot is a window that 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.

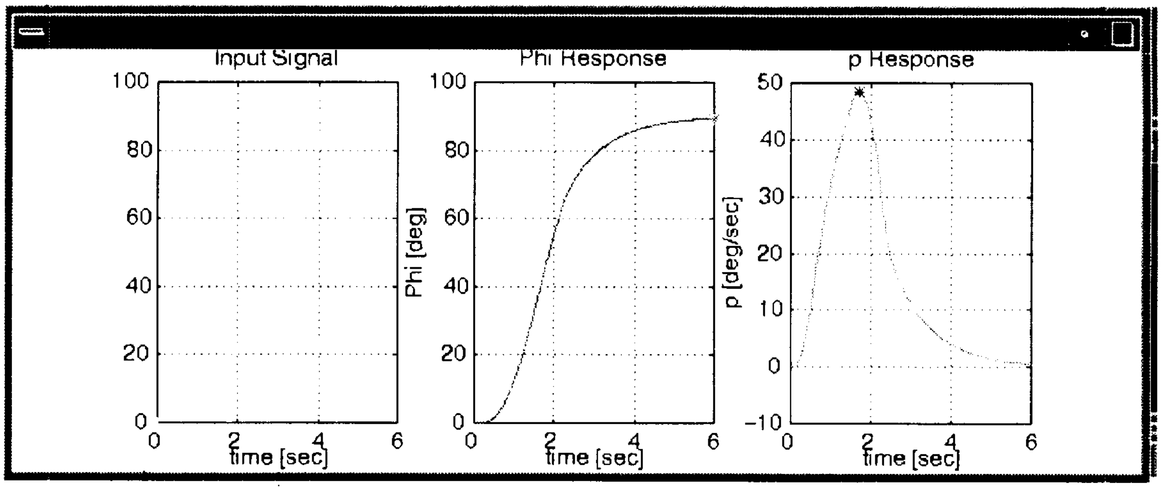


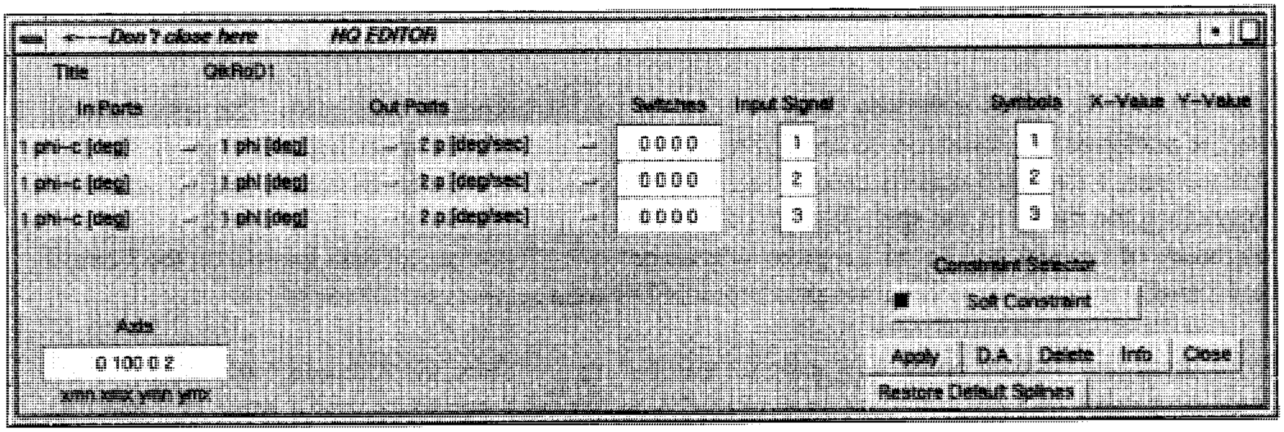
Figure 2. Example of supporting information required for the ADS-33 quickness specification.

3.5 Handling Quality Editor

The handling quality editor is used to "wire" specifications to the block diagram. This window (fig. 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 in port 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 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 specifications will require two out ports, 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.



Title QkRoD1

In Ports	Out Ports		Switches	Input Signal	Symbols	X-Value	Y-Value
1 phi-c [deg]	1 phi [deg]	2 p [deg/sec]	0 0 0 0	1	1		
1 phi-c [deg]	1 phi [deg]	2 p [deg/sec]	0 0 0 0	2	2		
1 phi-c [deg]	1 phi [deg]	2 p [deg/sec]	0 0 0 0	3	3		

■ Soft Constraint

Axis
0 100 0 2
xmn xmx ymn ymx

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

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 indicating the horizontal and vertical values of the 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 provides access to a series of features.

The Update button sets the entered parameters to the spec once they have been entered 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 Delete button is used to remove the selected spec from the handling quality window.

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 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.

The Restore Default Splines button restores the default graphic of the spec by resetting the geometry of the splines and axis values to the default values for that specification.

3.6 Distance Algorithm

The distance algorithm is the key element allowing for the tuning of the system. The optimization engine performs the comparisons in between the handling qualities specification. To perform the comparison, a standard scale is used to rate all the specs. The boundaries are the starting point for measuring the rating of the spec. It seems obvious that specifications that have results resting on the boundary of Level 1 and Level 2 regions will all be equal; likewise, all data located on the Level 2 and Level 3 boundaries will be weighted equally between all the specifications.

With these two starting points as a scale for comparing the specs, the following rules apply to the regions: a value of 1 given to data on the Level 1 and Level 2 boundary, and a value of 2 given to data lying on the border of the Level 2 and Level 3 regions.

The distance algorithm makes comparisons during the optimization of the specifications in a single problem. Normalizing the value of specifications against other specifications is crucial in evaluating the overall performance of the controller and aircraft. The bandwidth and quickness spec in figure 4 shows how the properties used in determining the algorithm in the distance algorithm are determined.

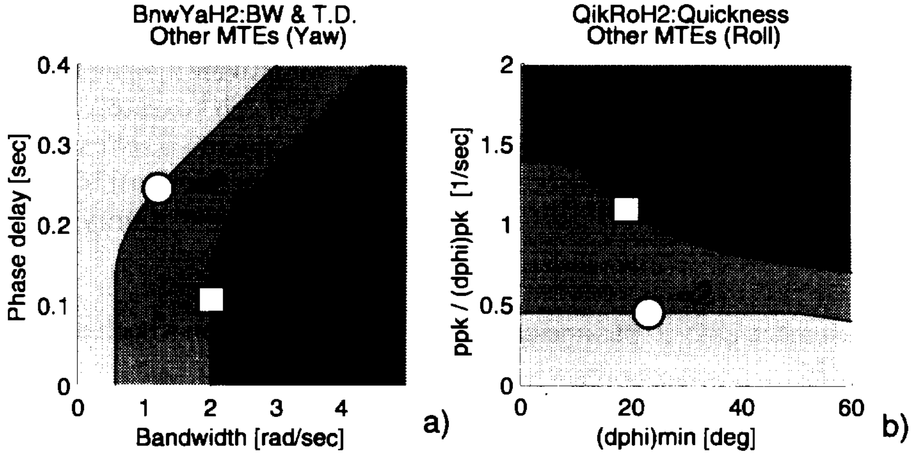


Figure 4. The normalized distance criteria. (a) All specifications that have data located on the Level 1/Level 2 boundary are rated equal in performance to all other specifications that have the data on this boundary. (b) Similarly, the data located on the Level 2/Level 3 border are rated equally across all specs.

As shown in the bandwidth spec, there are two borders that separate the different regions on the plot. A value of 1 is given to any data lying on the border that separates the Level 1/Level 2 boundaries. Likewise, a value of 2 is given to data lying on the border separating the Level 2/ Level 3 areas on the plot. Using these two marks, the rest of the algorithm returns values based on where the data point lies with respect to the two points. This basic rule implicates that the drop from the Level 1/Level 2 border to a Level 2/Level 3 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 one border to the other in all other specs. With this normalization all specifications can be measured in the distance separating the borders and scaled to a value of 1 or 2. The complexity of the algorithm goes into determining how to measure the borders to the data point. By 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. Figure 5 depicts 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 is done so that line distances are taken from the same direction and problems are avoided that arise in specifications 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 2 region is d3. This is found according to equations 1, 2, and 3.

$$\mathrm{d3} = \mathrm{d2} - \mathrm{d1} : \text{for } \ \mathrm{d1} < \mathrm{d2} \text{ (Level I)} \tag{1}$$

$$\mathrm{d3} = \mathrm{d2} + \mathrm{d1} : \text{for } \ \mathrm{d1}, \mathrm{d2} < \mathrm{d3} \text{ (Level II)} \tag{2}$$

$$\mathrm{d3} = \mathrm{d1} - \mathrm{d2} : \text{for } \ \mathrm{d1} > \mathrm{d2} \text{ (Level III)} \tag{3}$$

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 1, 2, and 3 regions, respectively.

$$nd = 1 + \left(\frac{-d1}{d2 - d1} \right) \tag{4}$$

$$or$$

$$nd = 1 + \left(\frac{d1}{d2 - d1} \right) \tag{5}$$

$$or$$

$$nd = 1 + \left(\frac{d1}{d1 - d2} \right) \tag{6}$$

For 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.

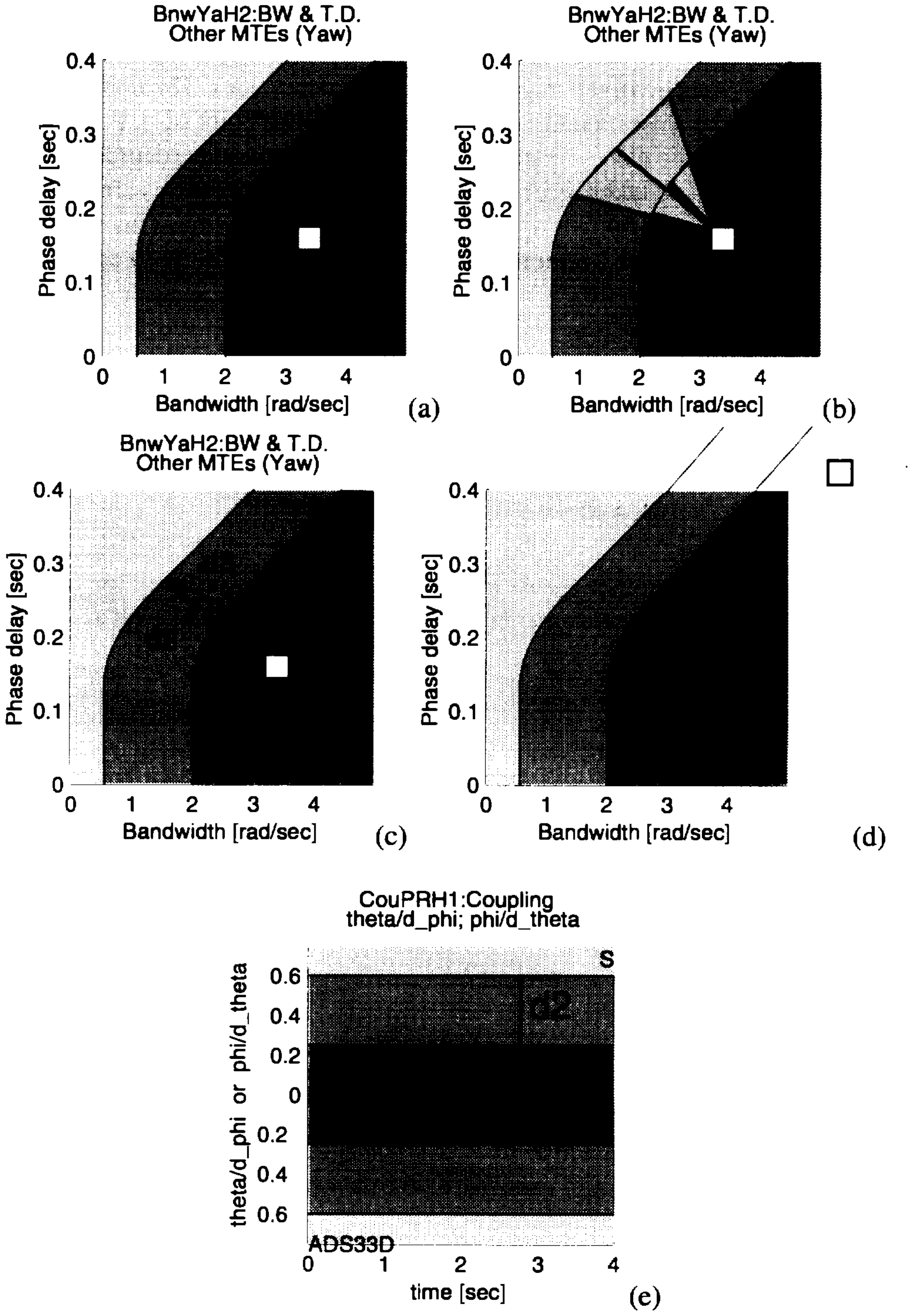


Figure 5. Distance algorithm calculation. (a) The distance algorithm calculates the distance of the point to the nearest point on the Level 1/Level 2 spline. (b) The distance to the second spline is then chosen from a point that is within 30 degrees of the first point. (c) The normalized distance is 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.

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 do not reduce the aircraft to a Level 2 from a Level 1. 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 1 and Level 3 areas (the width of the Level 2 region). This weighting distance normalizes the specifications between each other. This normalizing length is also used in calculating the design margin. The design margin represents a region that extends into the Level 1 region by a factor of the normalized distance. This is represented on the normalized scale that possesses a value of 1 on the Level 1/Level 2 border and a value of 2 on the Level 2/Level 3 border. The value of the design margin represents this value on this normalized scale. For example, a design margin value of 0.1 produces a region of ten percent of the normalized distance for that spec, extending into the Level 1 region. Figure 6 demonstrates the 0.4 design margin on a bandwidth and phase delay spec. The varying width of the Level 2 region is reflected in the distance that the design margin extends into the Level 1 region.

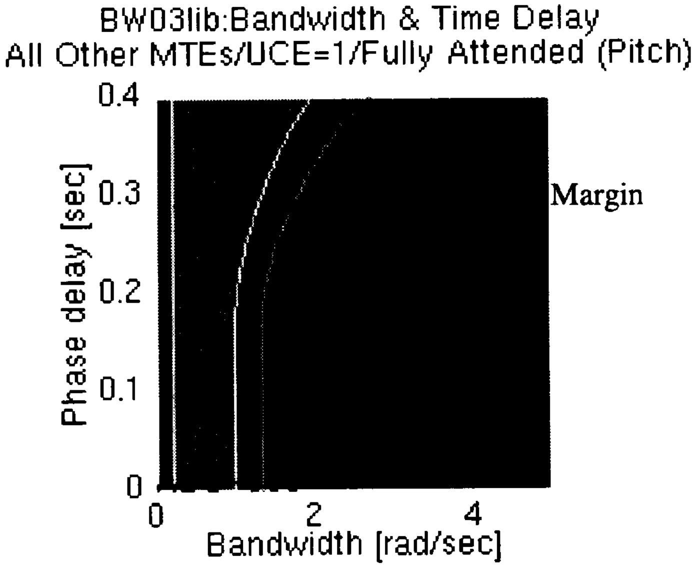
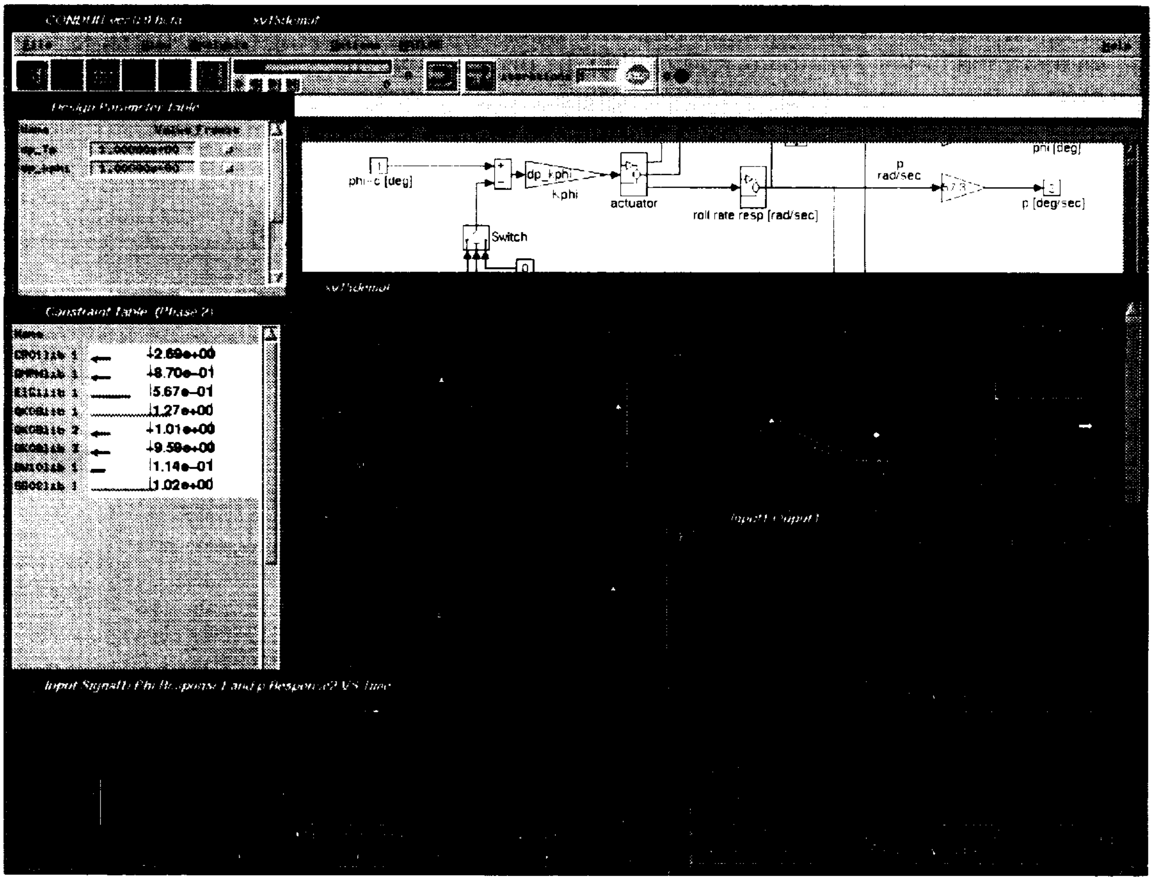


Figure 6. Bandwidth specification with a 0.4 design margin.

3.8 Environment

A considerable change in CONDUIT over the previous prototypes is the centralized integrated environment in which CONDUIT incorporates all the function and features. 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 is easy-to-use tools that do not require significant knowledge of programming to set up and analyze problems. A key advantage over earlier versions is that any modifications previously 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 specifications was a tedious undertaking. CONDUIT allows the specifications to be manipulated via the mouse. The addition, modification, and removal of the specifications are easily performed by the click of a mouse. Setup is performed with pull down menus allowing the user to install specifications with easy push button features. This is done without compromising the flexibility of the system and is what allows it to be so powerful, with 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.



Name	
[illegible]	2.69e+00
[illegible]	8.70e-01
[illegible]	5.67e-01
[illegible]	1.27e+00
[illegible]	1.01e+00
[illegible]	9.58e+00
[illegible]	1.14e-01
[illegible]	1.02e+00

Figure 7. The CONDUIT environment.

4.0 REQUIRED ELEMENTS FOR A DESIGN PROBLEM

Chapter 4 details the key elements for a design problem setup in the CONDUIT environment. Also the evaluation and optimization features are explained.

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; therefore, an obvious trade-off between accuracy and computational time exists. The SIMULINK environment supports sophisticated nonlinear designs expressed in the graphical environment 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 formatting that will be accepted by CONDUIT. The rules pertain to the selection of mnemonics for the design parameters, switches, and input signals.

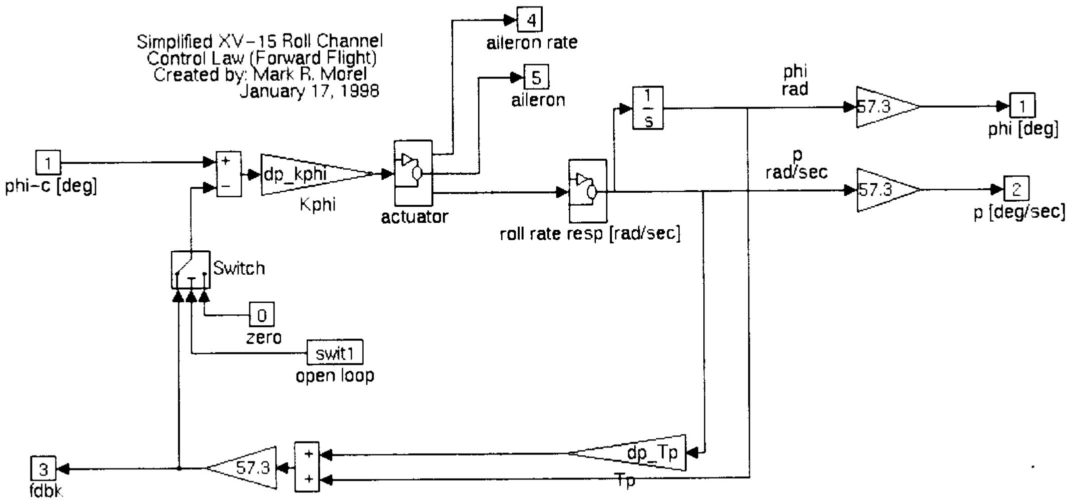


Figure 8. Example of a SIMULINK block diagram.

An initialization file must be created to support the block diagram to store information that could not be included in the model. This file contains the definitions for all the key variables, functions, and defining mnemonics. The initialization file is comprised of (1) default parameters that define the input signals that will be required for time domain simulations, (2) modeling data, and (3) 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, specifications are selected that allow the user to analyze and improve the controller. The user must pick specifications that provide a complete picture to determine if the controller and aircraft perform as a Level 1 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, specifications are chosen from the pull down library window with the mouse and added to the handling qualities window. Once here, the specifications are set up and modified by clicking on the individual specification and opening the handling quality editor. The specifications work between ports in the block diagram. The handling quality editor (section 3.5) displays the information required to "wire" the spec to the block diagram. The editor allows 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 boundaries can be moved by the user to suit the desired design. Once the specifications 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. The design is evaluated 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 give poor results. By selecting the spec, a supporting plot can be shown that illustrates why the value is calculated to be that way. 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 suitable initial design point. CONDUIT is 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, since other methods may require a week to run a new set of design parameters through the system. The user can run "what if" scenarios by changing different aspects of the control system and almost instantly see the effect those changes have on a set of 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 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 twelve design parameters, it becomes difficult to determine which design parameters and which combination of values will produce an overall 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 are divided into three priorities: 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 is conducted in three phases. In phase 1 the program works on satisfying the hard constraints. CONSOL-OPTCAD adjusts the design parameters to achieve Level 1 for all the specifications that are set to hard constraints. Typically, hard constraints are chosen to be the specifications that guarantee stability or that are of particular interest to the user for this problem. These are specifications that must be met, with no exceptions. In phase 2 CONSOL-OPTCAD attempts to satisfy the specifications that are set to be soft constraint or objective functions into the Level 1 region. In this phase these two types of specifications behave in a similar manner. These specifications are ones that you would like to meet Level 1, 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 1 area, the optimization engine reaches the third phase. In the third phase CONSOL-OPTCAD attempts to minimize the specifications chosen to be objectives, which drive the data points for the specification further into the Level 1 region, without violating any hard or soft constraints, until an optimal design has been reached. This design is normally characterized by a local minimum.

5.0 UH-60A DESIGN EXAMPLES

Chapter 5 discusses three design problems that feature the use of CONDUIT in performing the deisgn. All three design examples use the UH-60A Black Hawk as the design vehicle.

5.1 Introduction to Design Problems

The following designs demonstrate the use of CONDUIT with the Sikorsky UH-60A Black Hawk helicopter. The Ames Research Center operates a Black Hawk research helicopter under the RASCAL program (ref. 7; fig. 9). A baseline model following control law is being developed using conventional design techniques and based closely on the ADOCS FBW design (ref. 8) the baseline ADOCS control laws for the RASCAL helicopter were run through a design session with CONDUIT. Designs were 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 used twelve design parameters, nine feedback gains, and three model parameters for the pitch, roll, and yaw loops. They were chosen to control pitch, roll, and yaw feedback of a proportional integral derivative (PID) controller. The three model gains affect the bandwidth of the command model.

Figure 9. The RASCAL UH-60 helicopter.

The first problem shows how the RASCAL architecture is 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 Black Hawk Design for ADS-33 Requirements

The first design problem addressed in this paper is concerned with establishing a baseline set of design parameters that allows the helicopter to function in accordance with ADS-33 control laws (ref. 1). This set of design parameters establishes a confirmed standard that can be referred to in future designs and specialized optimizations. The Black Hawk 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 (fig. 10) serves 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.

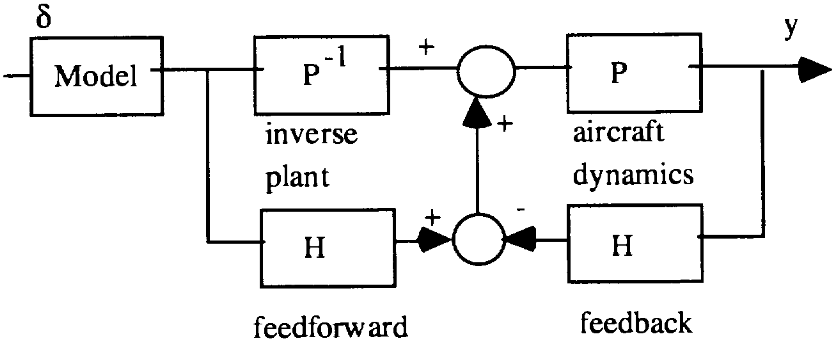


Figure 10. Model following block diagram.

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.

This first design problem was to tune this architecture (including model) so 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-degrees-of-freedom state-space representation of the UH-60A helicopter dynamics (ref. 9). 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 nine feedback gains (three for each of the pitch, roll, and yaw channels) as nine of the twelve design parameters. The remaining three design parameters make up the second order model transfer functions.

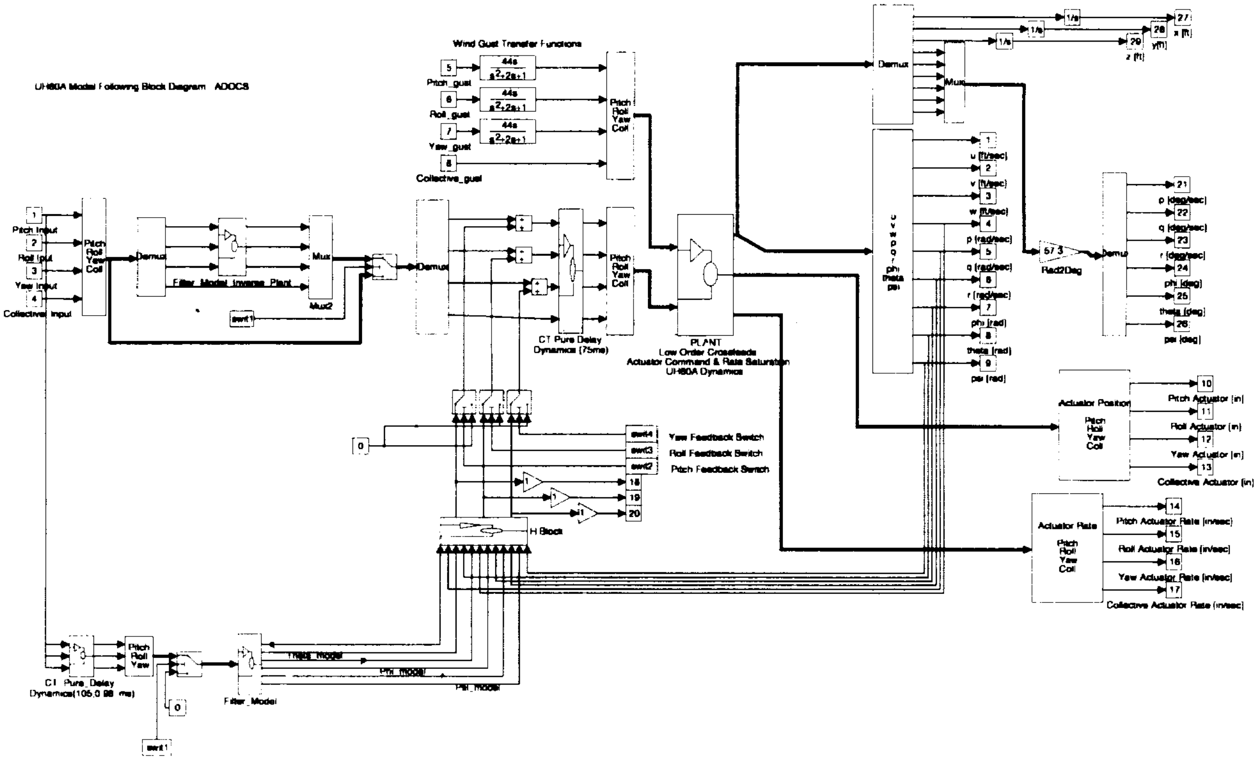


Figure 11. UH-60A model following control law.

The second order inverse of the plant dynamics is obtained by taking frequency data from the plant and crossfeeds together. These data are fed into NAVFIT, a curve fitting utility of CIFER (ref. 10) that produces the transfer functions in equations 7 through 10.

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

$$\frac{q}{\delta}=\frac{3.20647}{(s+0.58468)(s+11.770)} \tag{7}$$

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

$$\frac{p}{\delta}=\frac{16.6734}{(s+6.1178\pm3.39789i)} \tag{8}$$

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

$$\frac{r}{\delta}=\frac{397.616}{(s+592.64)(s+0.79536)} \tag{9}$$

It is obvious that the yaw channel has an unusually high pole and gain. This indicates that the first order inverse is adequate for the plant inverse. Previous papers (ref. 11) indicate that the first order is adequate for the Yaw channel. The first order expression is given as:

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

The integral of these three transfer functions (converting q/δ, p/δ, and r/δ to θ/δ, ϕ/δ, and ψ/δ, respectively) is 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. 1) and those developed at Ames Research Center based on classical control theory. The specifications chosen are crossover frequency, actuator energy, eigenvalue stability, gain and phase margins, bandwidth, quickness, coupling, and wind gust. Figure 12 shows the specifications that are used in this and the next example design problem. The specifications are ranked by their optimization type (hard constraints, soft constraints, and optimization function). The eigenvalue stability specification (fig. 12(a)) uses the real part of the roots of all the poles for the system. It drives them into the left hand plane to ensure overall stability of the plant and controller. The Level 2 region on this spec is made very small to enforce the idea that either the roots are stable (Level 1) or they are not (Level 2).

The gain and phase margin spec (fig. 12(b)) 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 specifications were selected as hard constraints.

The bandwidth and phase delay spec (fig. 12(c)) 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 quickness specification (fig. 12(d)) is a measure of agility of the aircraft. The requirement ensures that the aircraft is more agile for small attitude changes and allows 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 specifications (fig. 12(e)) help minimize the amount of coupling in the off-axes.

The wind gust specifications (fig. 12(f)) 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.

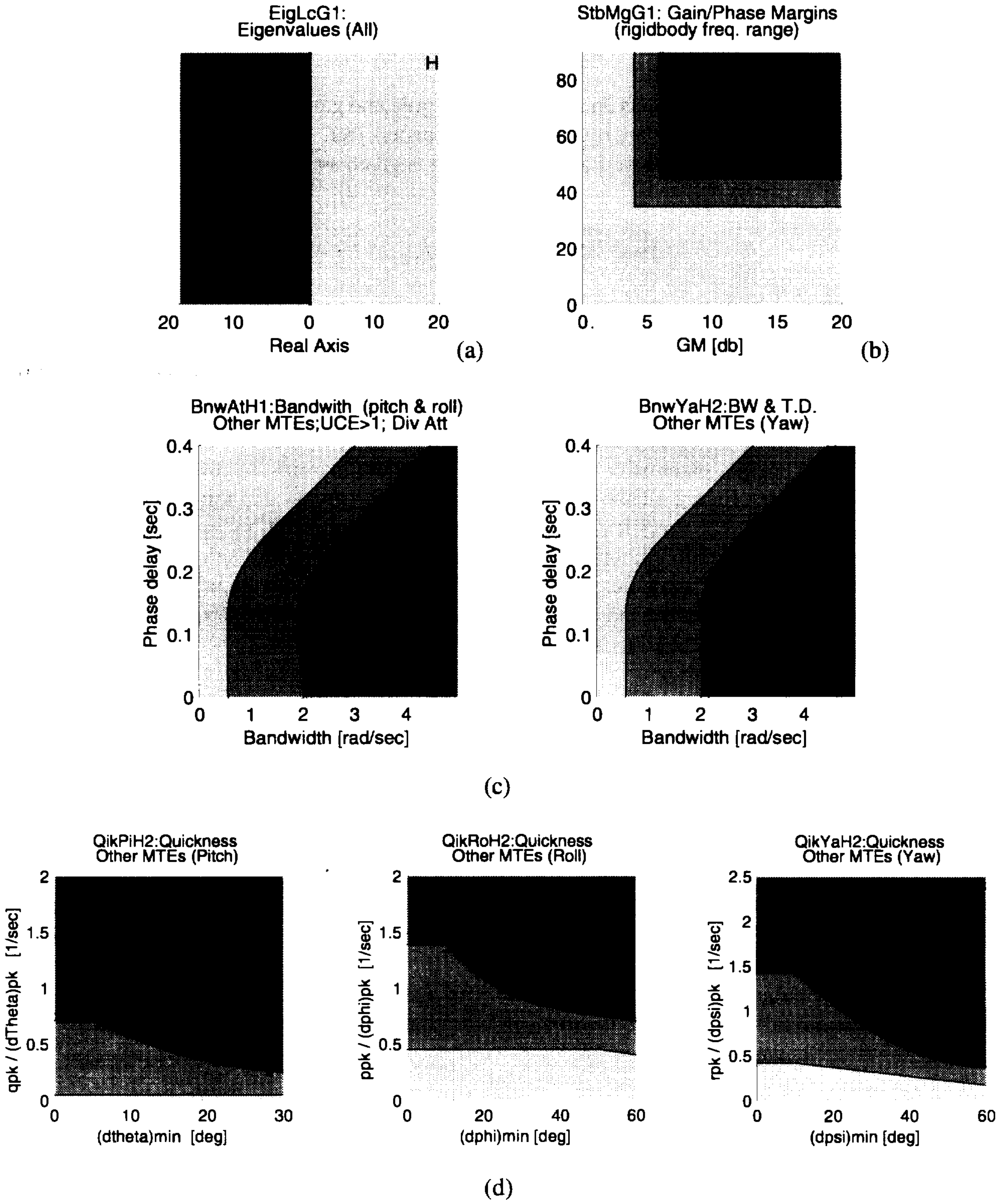
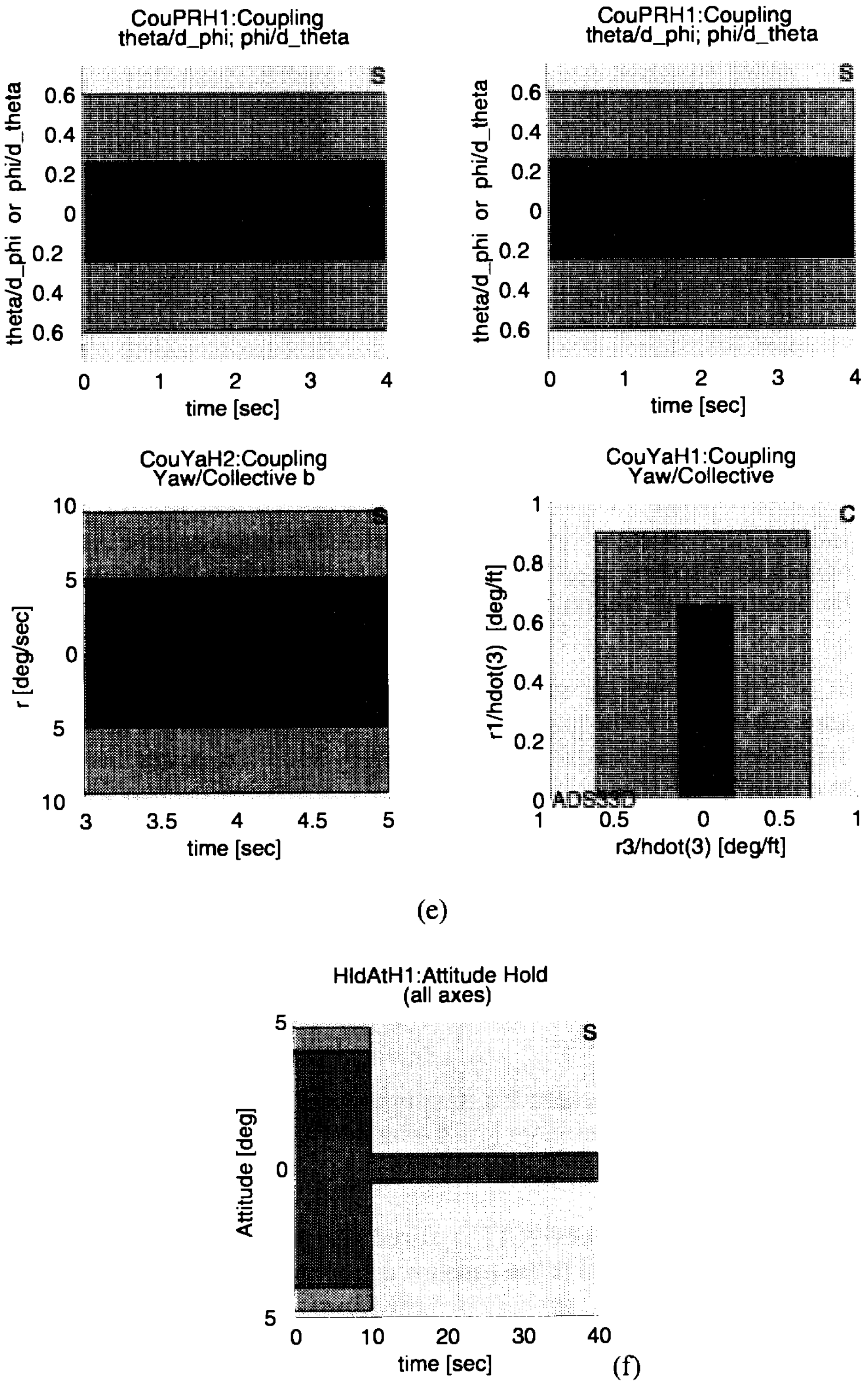


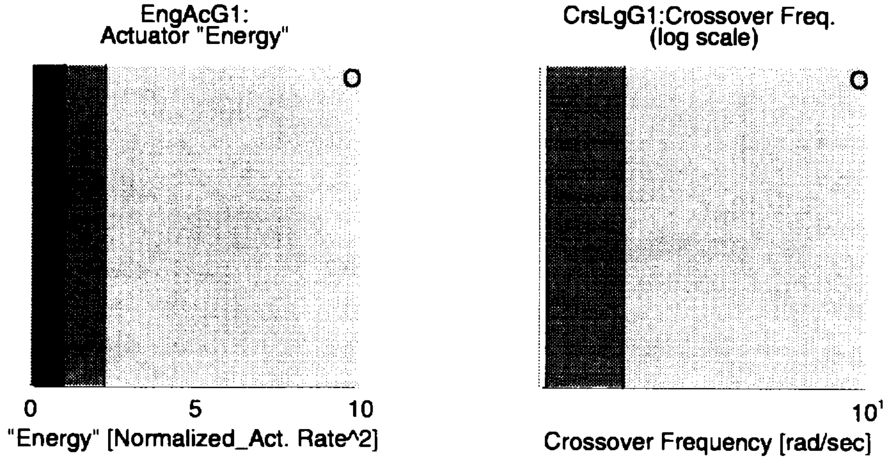
Figure 12. Specifications used in ADS-33 control law design. (a) Eigenvalue stability. (b) Gain and phase margin. (c) Bandwidth. (d) Quickness specifications.



(e)

(f)

Figure 12 Continued. (e) Coupling specifications. (f) Wind gust specifications.



(g)

Figure 12 Concluded. (g) Crossover 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, quickness, coupling, and windgust specifications are set as soft constraints since these requirements are not as crucial as stability. It is possible that a Level 2 solution in some of these specifications might be accepted if limited by the aircraft's structure, power, hardware, control surfaces, or aerodynamics.

The two specifications for optimization (fig. 12(g)) 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, increasing fatigue wear. Driving down the crossover frequency drives down the actuator energy. These specifications are set to objective functions in an effort to minimize the values and reduce any extra actuator energy not required for a Level 1 design.

A design margin of 0.1 is selected to ensure a minimum level of robustness. The 0.1 design margin extends the Level 2 region ten percent more into the Level 1 region so that solution will not be a borderline Level 1/Level 2 solution.

After the specifications are set up, CONDUIT is placed into the run (analysis, optimization) mode. The design parameters are initially set to the baseline default parameters and the analysis is run (fig. 13). Unless specified otherwise, the data in these figures use the triangle for pitch, the inverted triangle for roll, and the 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 specifications appear from left to right in the following figures. The example of this order is shown in the wind gust specification. When the baseline 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 2 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 is obvious that the design parameters dp_Mpsi of

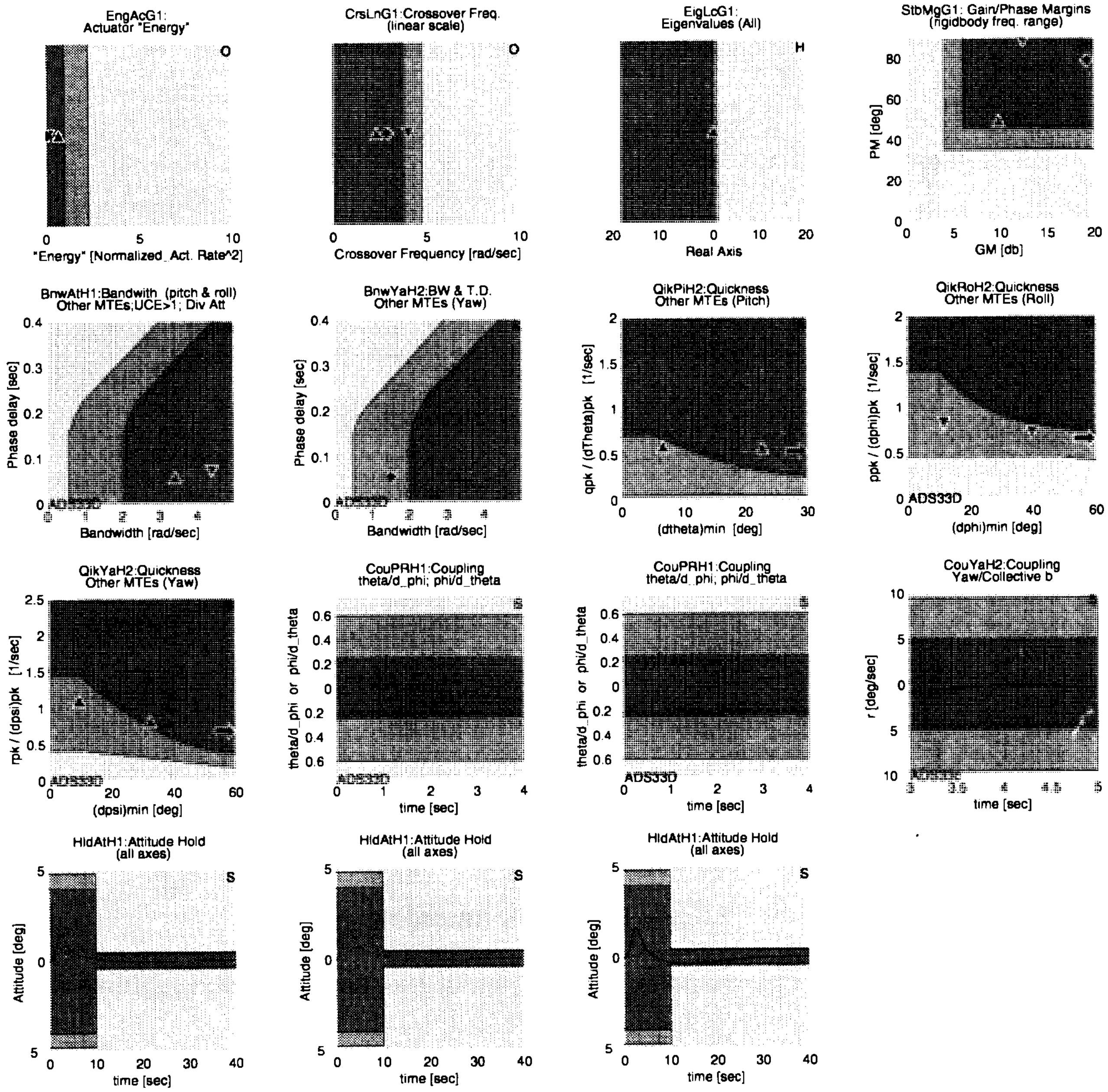


Figure 13. Handling quality analysis of baseline control laws.

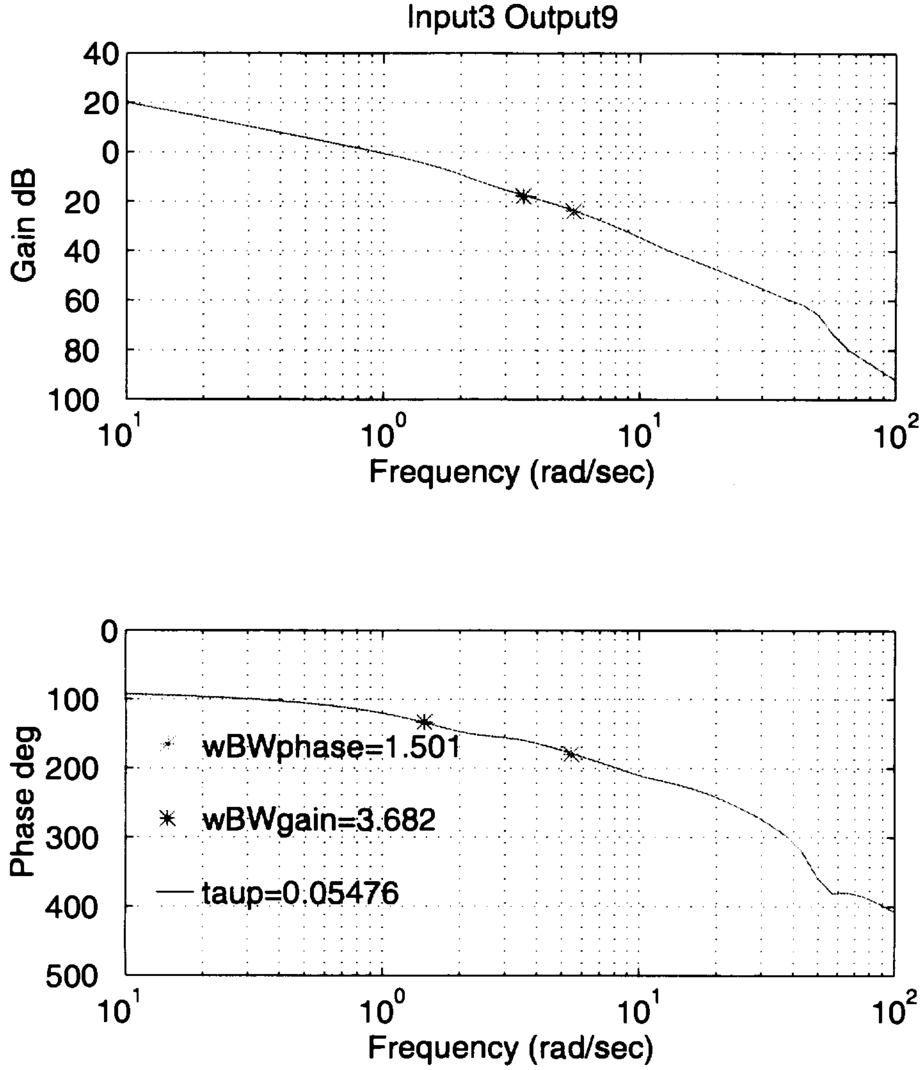


Figure 14. Showing the bandwidth for the yaw channel using baseline design parameters.

the yaw model will need to increase. To examine why the quickness specification does not meet the requirements for the small yaw inputs, figure 15 contains the supporting plot for the small input yaw channel.

The roll feedback channel exhibits a large amount of phase margin (fig. 13). The supporting plot for the roll stability margin specification depicts the frequency response for the roll feedback (fig. 16). CONDUIT is able to take advantage of that excessive phase margin to improve other specification when the control laws are optimized.

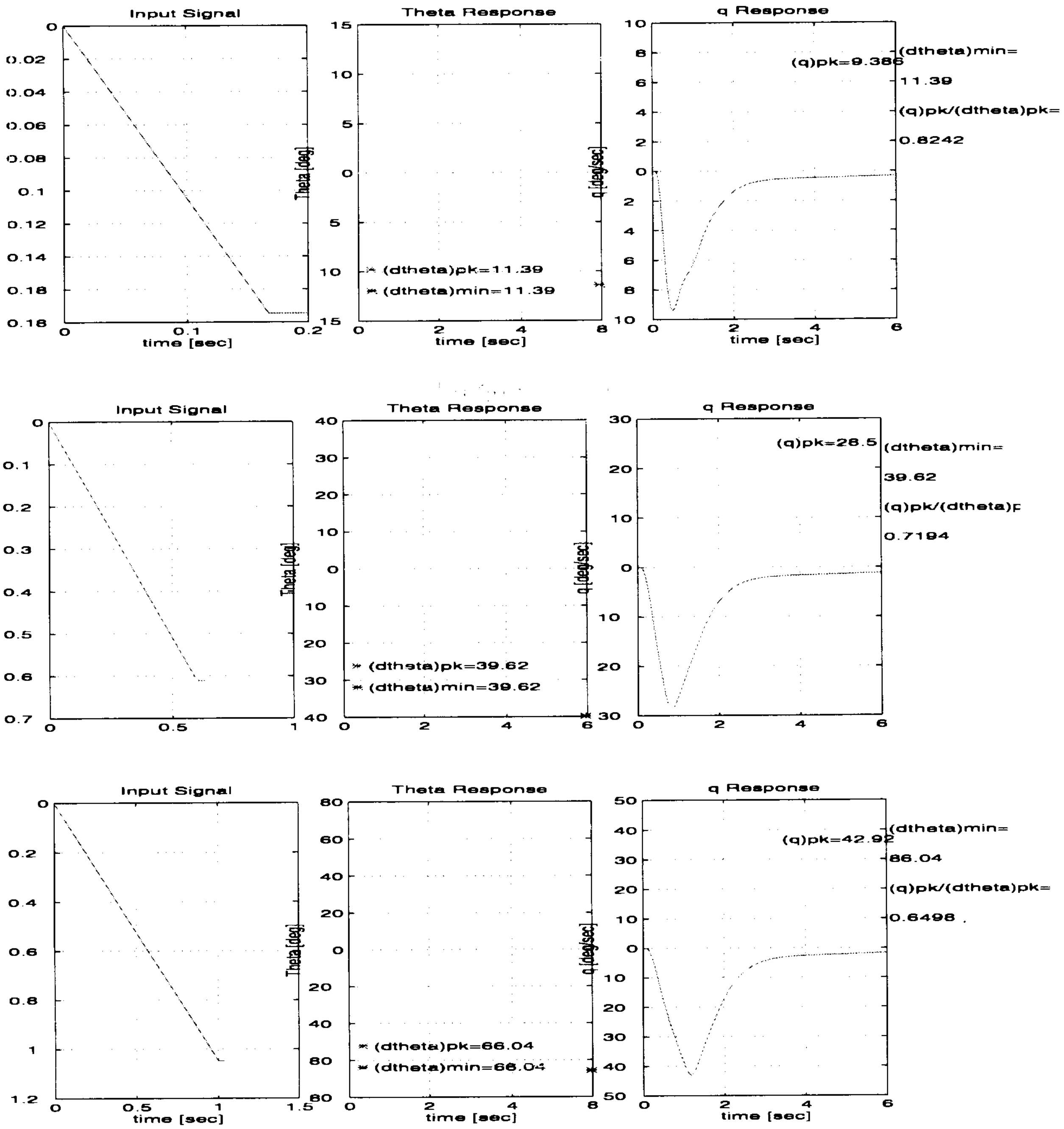


Figure 15. Supporting plots for roll quickness handling quality specification. The input signal is negative because in this direction the aircraft exhibits a poorer response.

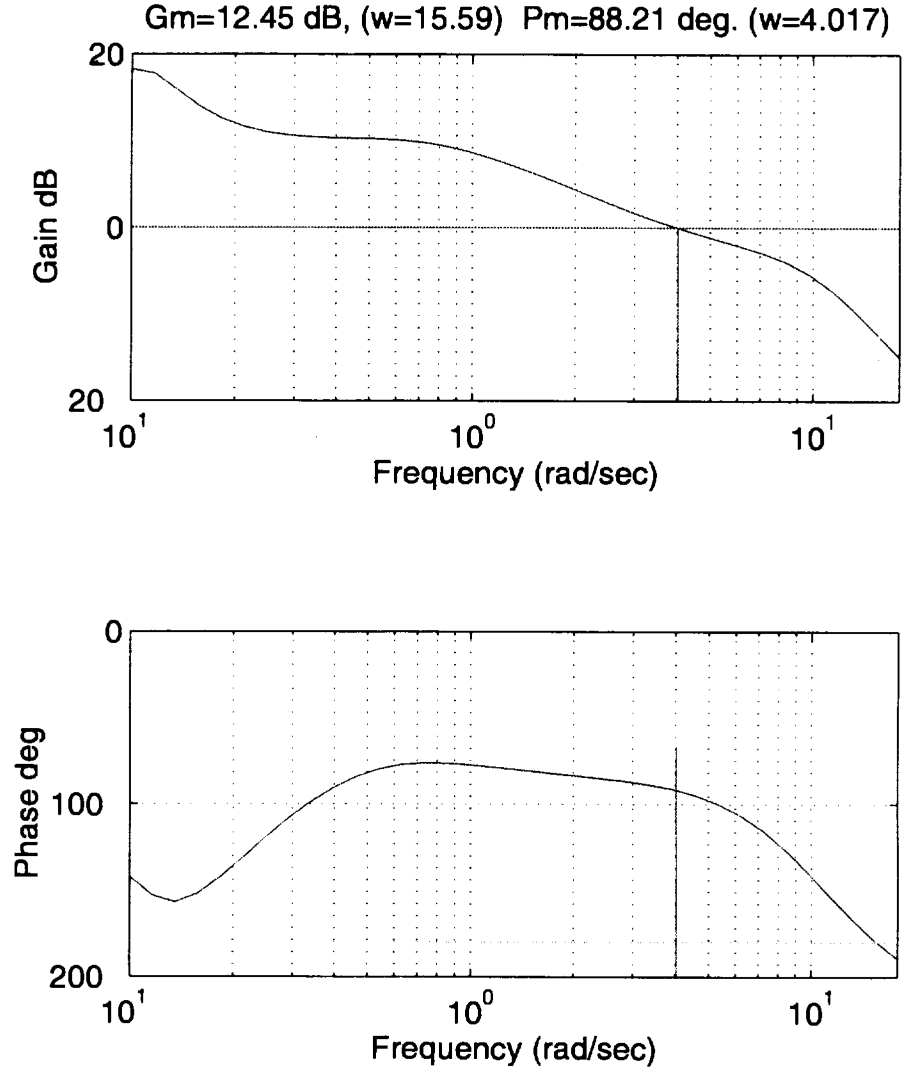


Figure 16. The broken loop frequency response for the roll channel for the baseline solution.

Next, CONDUIT is exercised to optimize the design parameters so the controller will meet all the specifications and drive down the crossover frequency and depict this solution. CONDUIT is able to satisfy all the handling quality requirements (fig. 17) 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 allows 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 can be met.

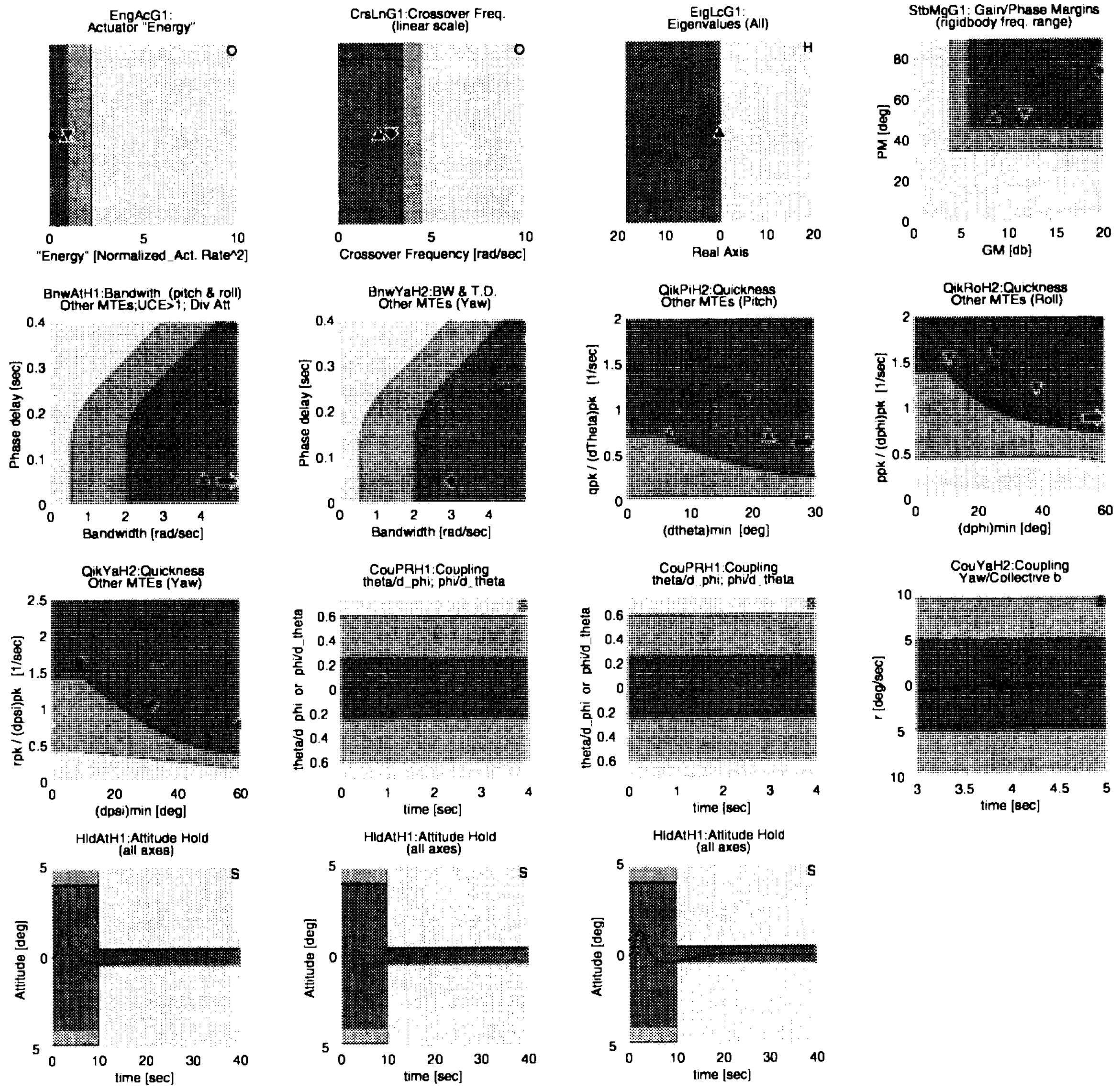


Figure 17. CONDUIT optimized solution.

Table 1. The design parameters taken from Tischler (ref. 12) paper and CONDUIT solution.

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	0.2809	KIphi	0	2.328
Kr	3.2	3.068	KIpsi	0	1.088

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 attitude and rate responses are clean and do not possess any unwanted oscillations. The last item to notice is the

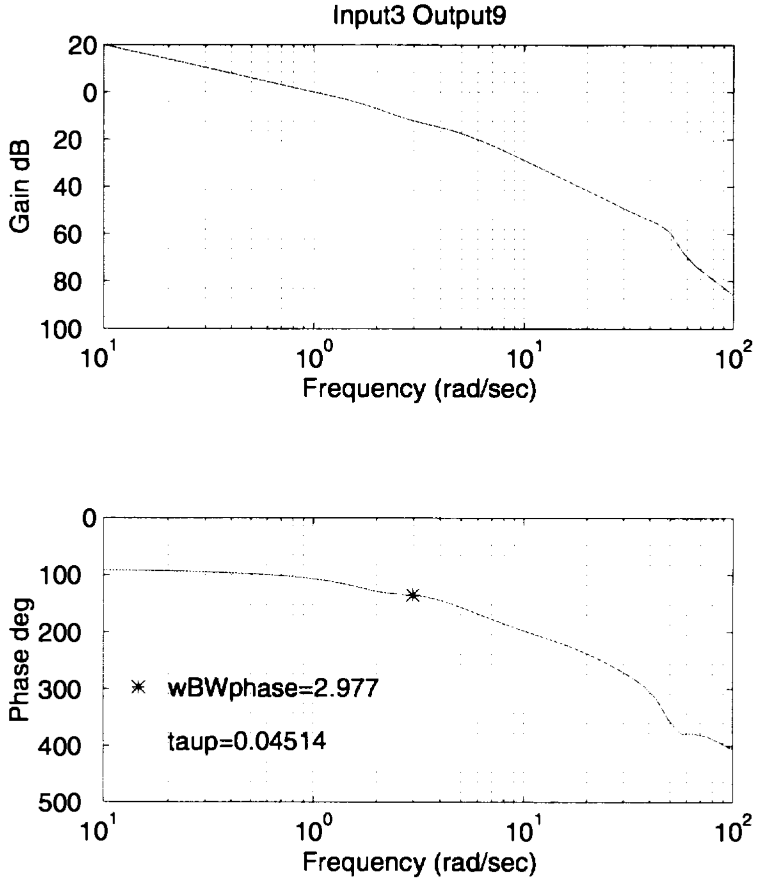


Figure 18.Yaw bandwidth supporting plots.

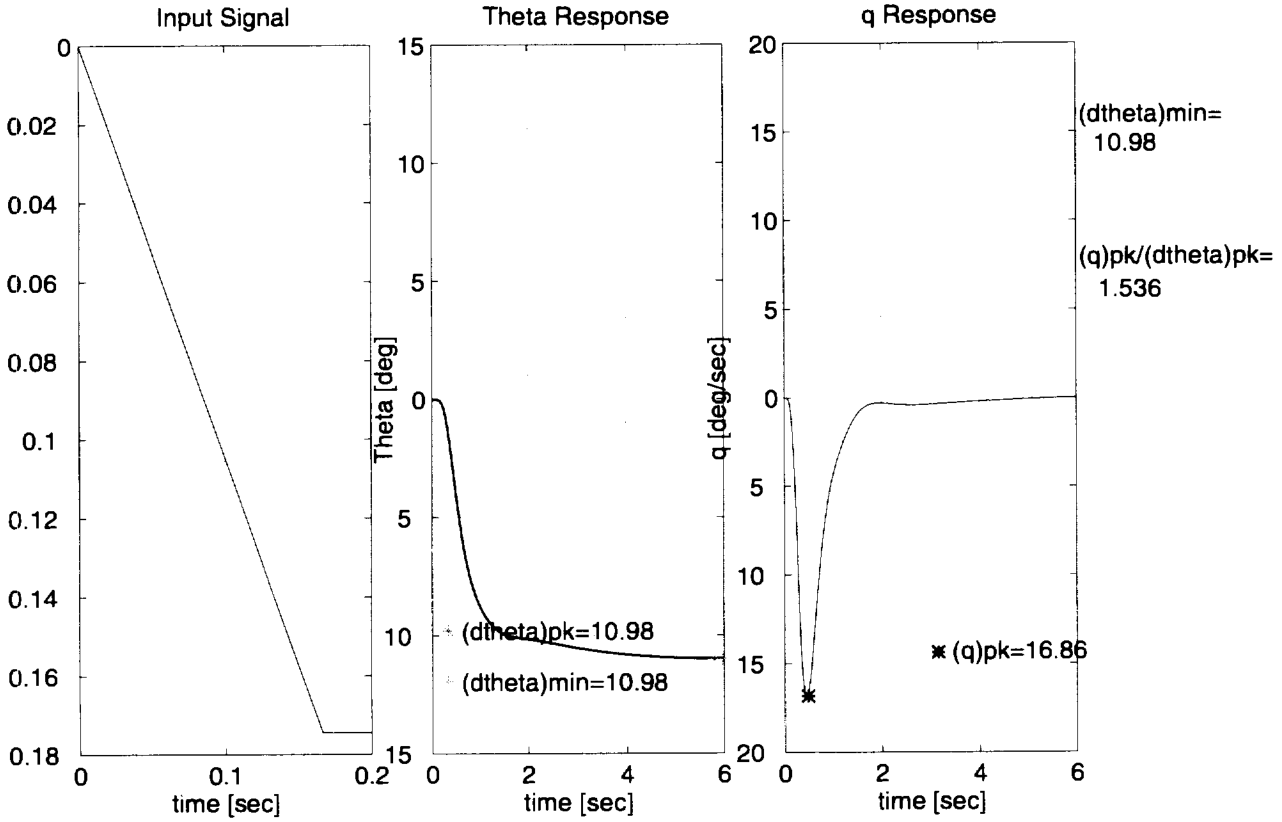


Figure 19. The supporting plots for the roll quickness specification for small input signal once it has achieved Level 1.

reduction in phase margin in the roll channel attributed to a large decrease in Kp with the increase of KIphi. The significant reduction 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 (fig. 16). The reduction in Kp has had this effect in the design.

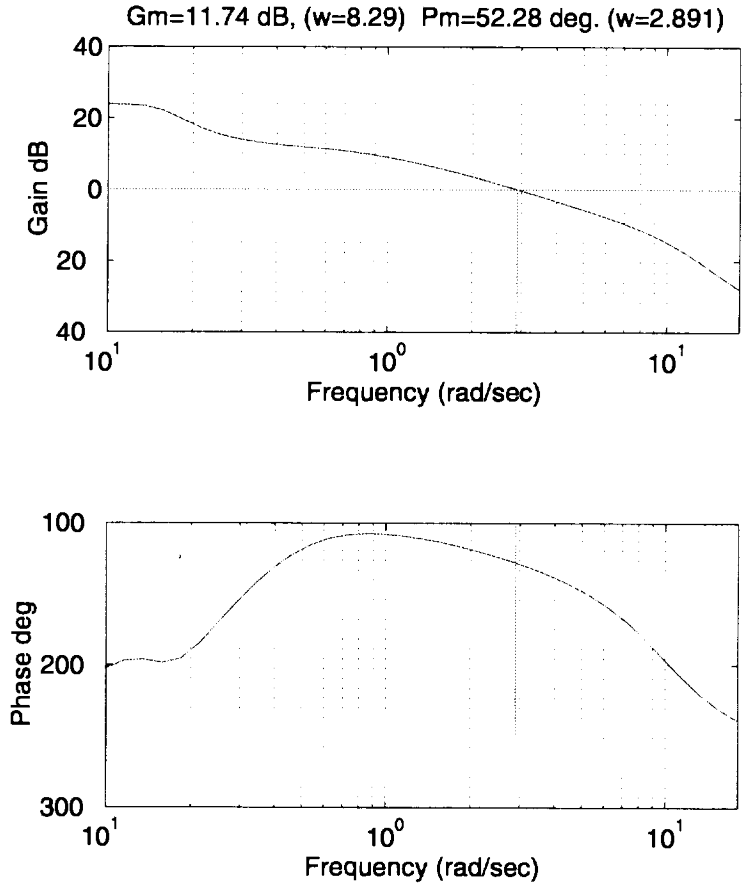


Figure 20. Roll channel gain and phase margin supporting plot after tuning in CONDUIT environment.

With the CONDUIT system, a single engineer has tuned the control laws to meet ADS-33 handling quality specifications 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 can 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 are not the focus of this solution, but rather the ability to track a given model.

The purpose of this solution demonstrates 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}.

$$\begin{aligned} y/\delta &= M(P^{-1} + H)(P/1 + PH) \\ y/\delta &= (M^{-1}P + PH)/(1 + PH) \\ y/\delta &= M(1 + PH)/(1 + PH) \\ y/\delta &= M \end{aligned} \tag{11}$$

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 feedforward 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. 13). Also matching the end-to-end response of the aircraft with the model is impossible, since the highest feedback gains still will not get the high frequency phase curves to match. The best that can be hoped for is to match the model time plant delay **M*D**.

Table 2. Delay parameters

Channel	Filter (ms)	Actuator (ms)	Helicopter dynamics (ms)	Total delay (ms)
Pitch	25	20	60	105
Roll	25	20	57	102
Yaw	25	20	53	98

The model used for the model block was taken from Tischler's paper (ref. 12).

$$\theta_c / \theta_m = \frac{4}{(s+2)^2} \tag{12}$$

$$\phi_c / \phi_m = \frac{6.25}{(s+2.5)^2} \tag{13}$$

$$\Psi_c / \Psi_m = \frac{2}{s(s+2)} \tag{14}$$

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 baseline values provided (ref. 12). 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. 14).

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

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 comes 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.

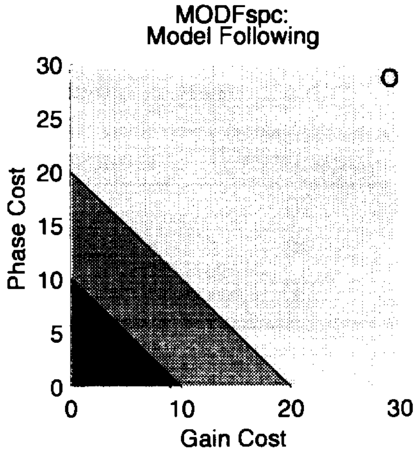


Figure 21. The model following specification.

The requirements to be met are overall stability, gain and phase margins, off-axis coupling, and wind gust response. The program was set up 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 (fig. 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 becomes obvious that integral feedback loops need 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.

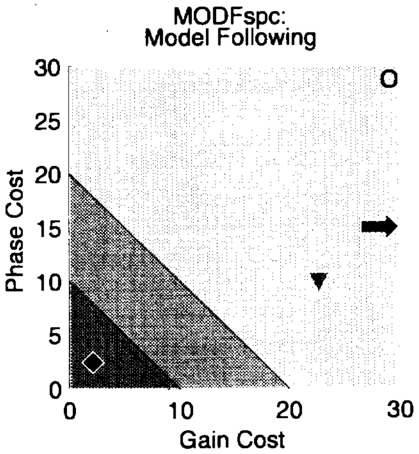


Figure 22. Model following specification for the baseline design parameters.

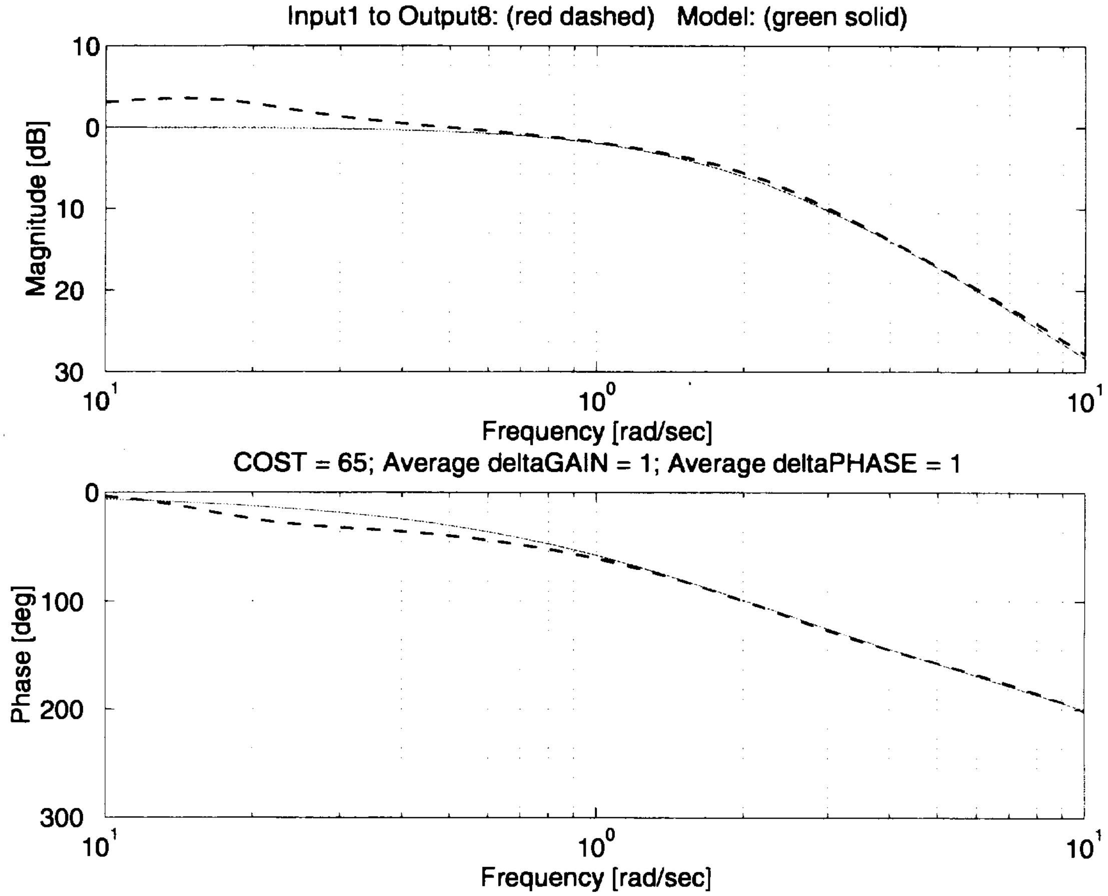


Figure 23. Supporting plot for the model following spec (pitch).

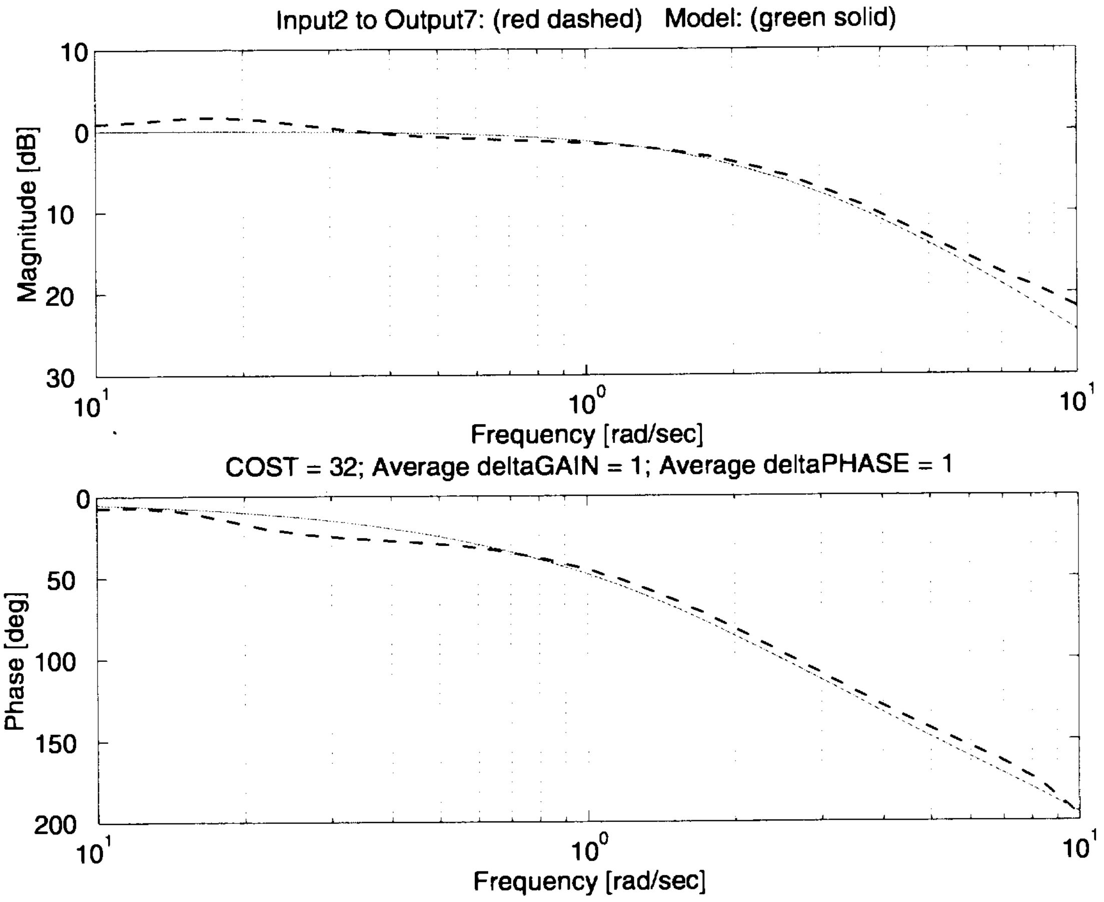


Figure 24. Supporting plot for the model following spec (roll).

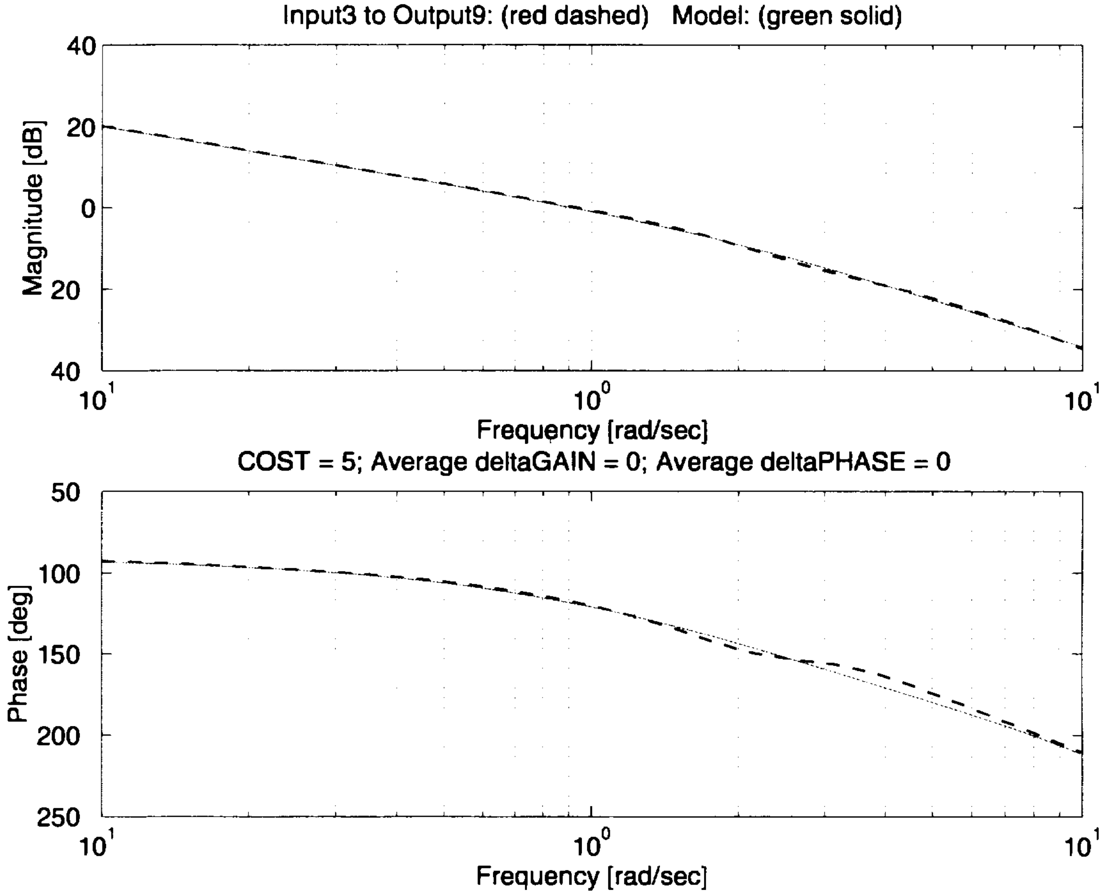


Figure 25. Supporting plot for the model following spec (yaw).

The result is a match of the end-to-end response to the model with costs of 12, 11, and 7 for pitch roll, and yaw, respectively (fig. 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 more closely. 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 (table 3). As a result, the phase margins are compromised and the wind gust shows more overshoot than the ADS-33 solution. The model following solution shows that the crossover frequencies were forced even lower. The limiting factor for the crossover frequencies in this optimized design seems to be the wind gust response. With the high integrator gain in the pitch and roll, the crossover frequency can be reduced in these channels without driving the wind gust spec to the edge. The yaw crossover frequency is driven down as well (table 4). This occurs 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 when this did not occur in the ADS-33 solution.

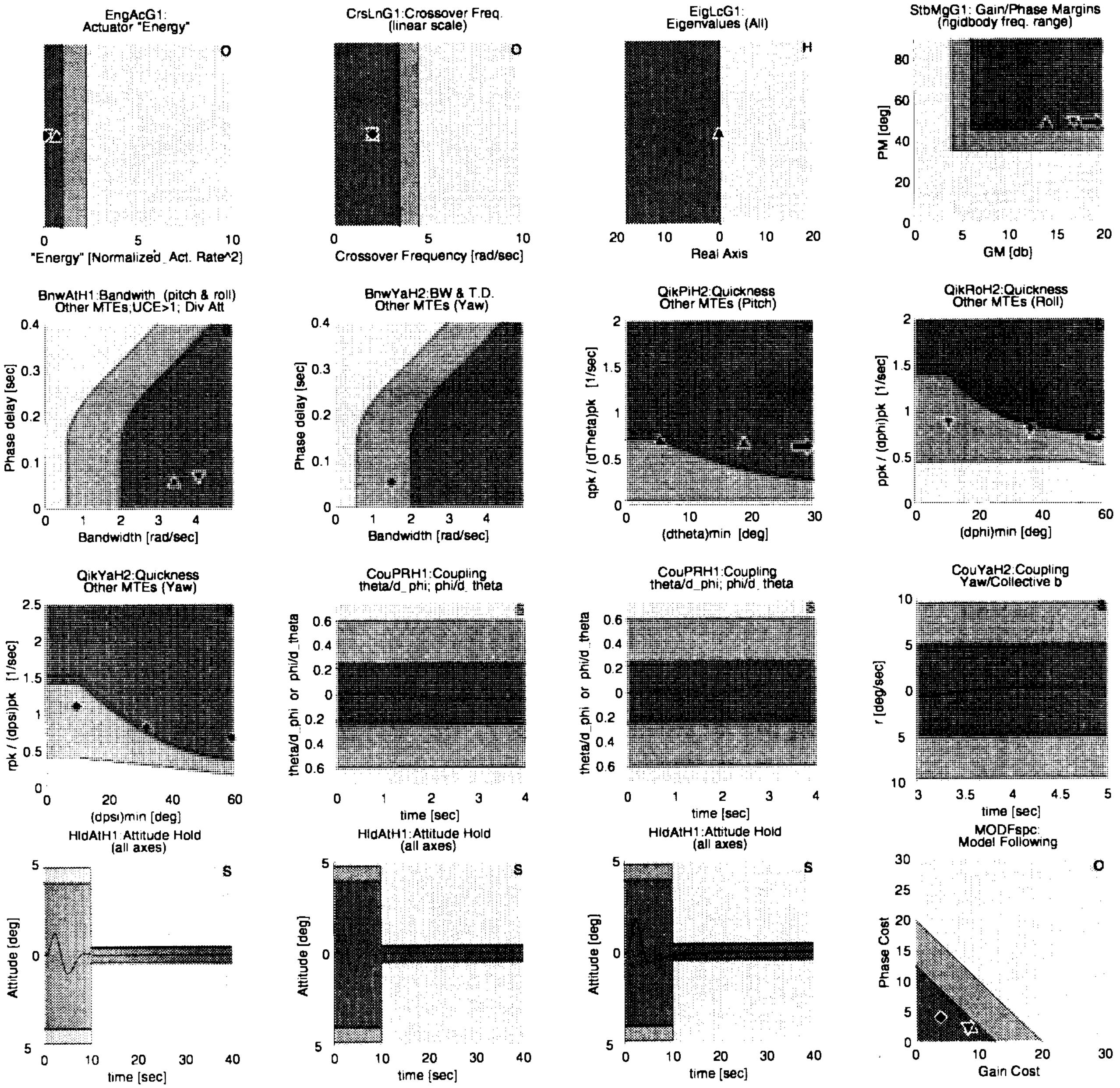


Figure 26. CONDUIT's optimized parameters for the model following solution.

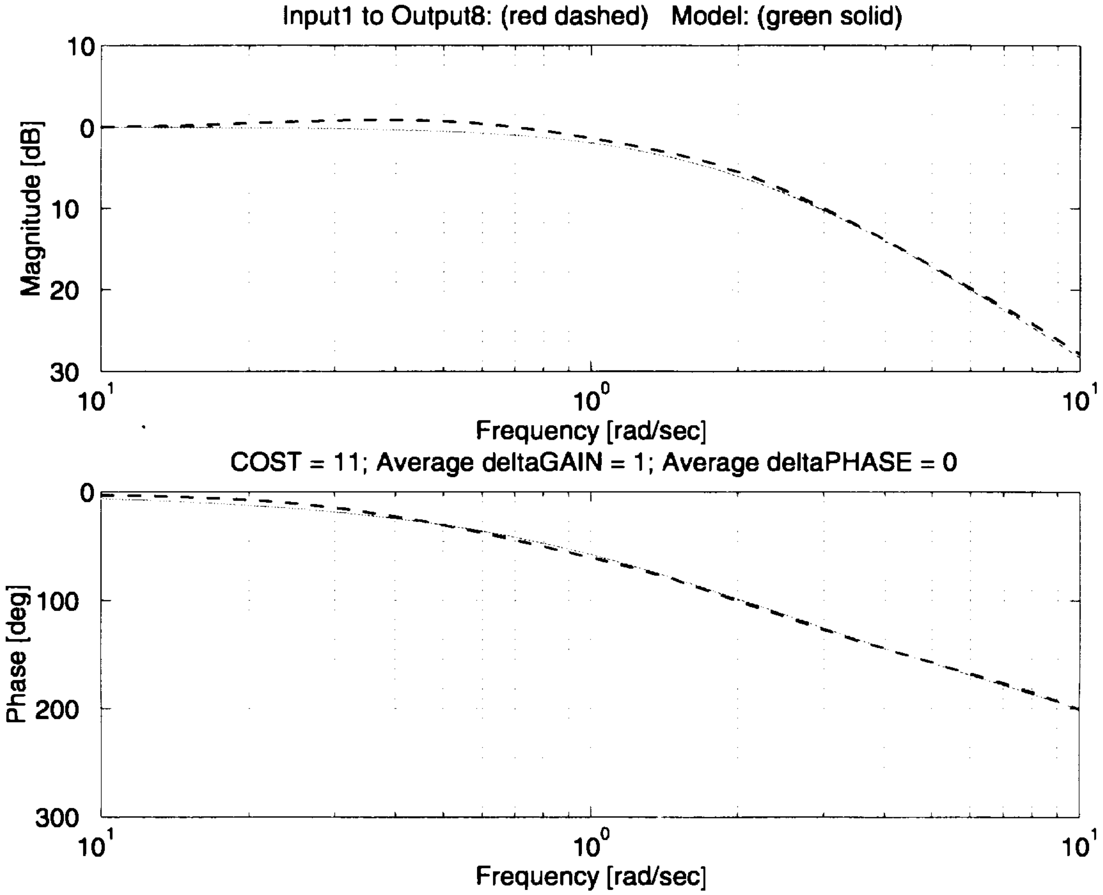


Figure 27. Supporting plot for the model following spec with optimized solution (pitch).

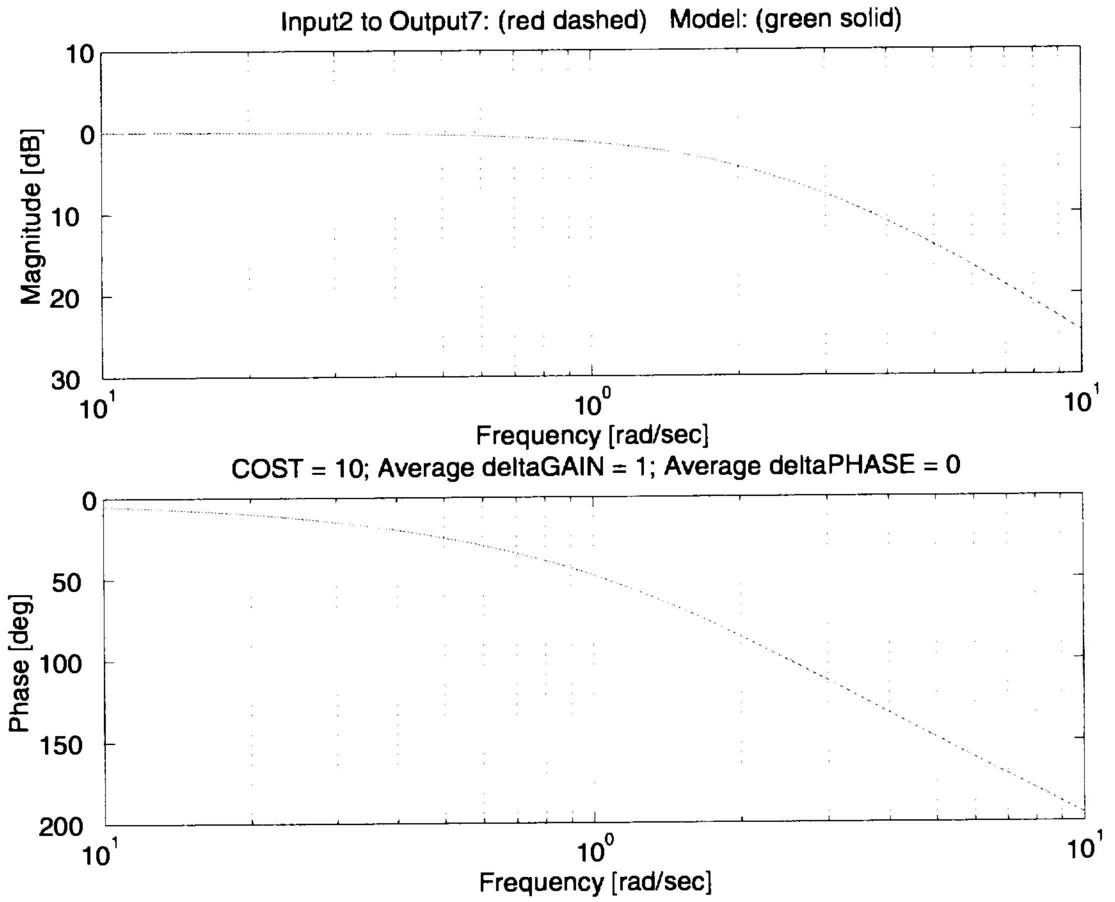


Figure 28. Supporting plot for the model following spec with optimized solution (roll).

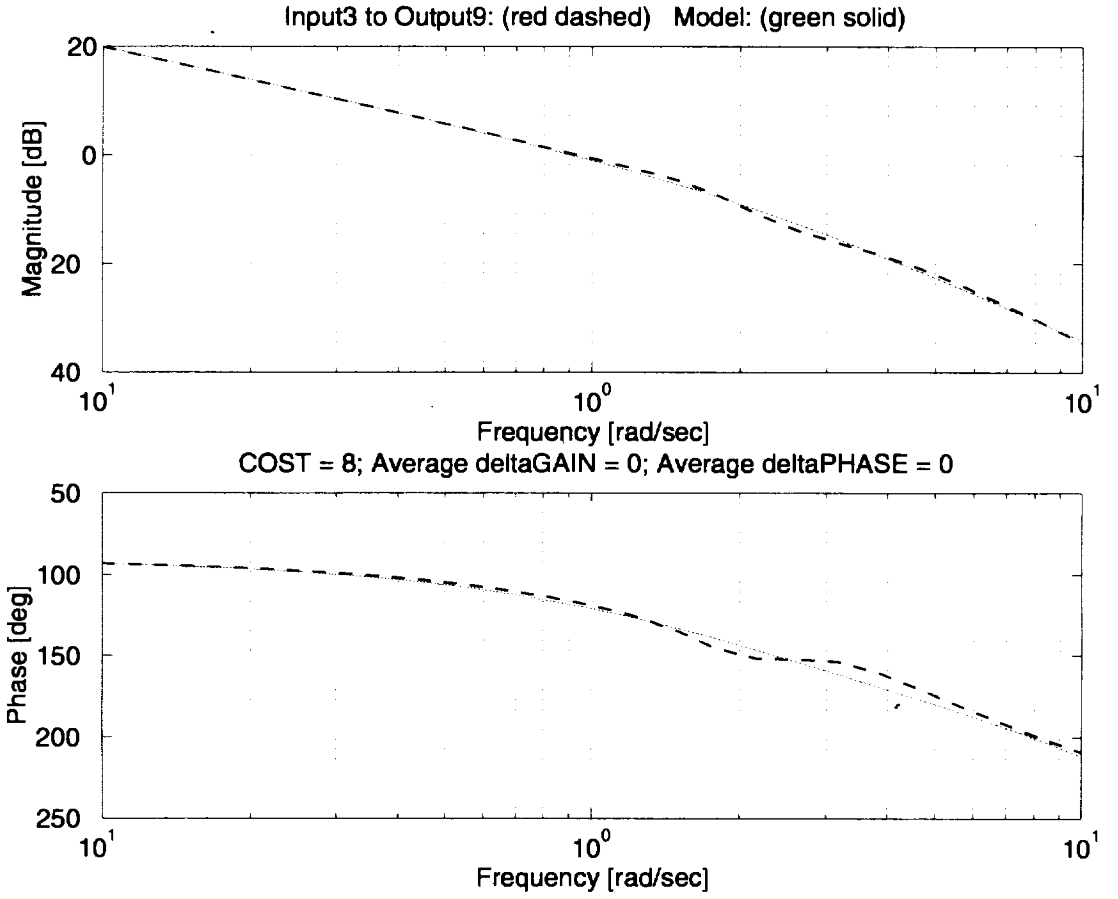


Figure 29. Supporting plot for the model following spec with optimized solution (yaw).

Table 3. Initial design parameters as provided by Tischler's paper (ref. 12) with CONDUIT optimized parameters for the model following solution

Design parameters	Initial	Final	Design parameters	Initial	Final
Ktheta	13.6	11.6	Mtheta	2	2
Kphi	8	5.5	Mphi	2.5	2.5
Kpsi	7.6	6.6	Mpsi	2	2
Kq	6.4	7.7	KItheta	0	9.0
Kp	2.4	1.4	KIphi	0	9.0
Kr	3.2	1.7	KIpsi	0	1.6

The mismatch of the phase curves as discussed in the Tischler paper (ref. 12) was attributed to the improper use of delay in the feedforward system of the block diagram, 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 a 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.

Table 4. The final values of the optimization constraints

		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

5.4 Windgust Rejection

This problem is used to demonstrate the trade-off possibilities that can be conducted by an engineer using CONDUIT. CONDUIT provides the ability to compare families of designs, providing insight into the relationship of competing specifications. This problem is 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 Black Hawk helicopter to maintain its 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 into 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 velocity.

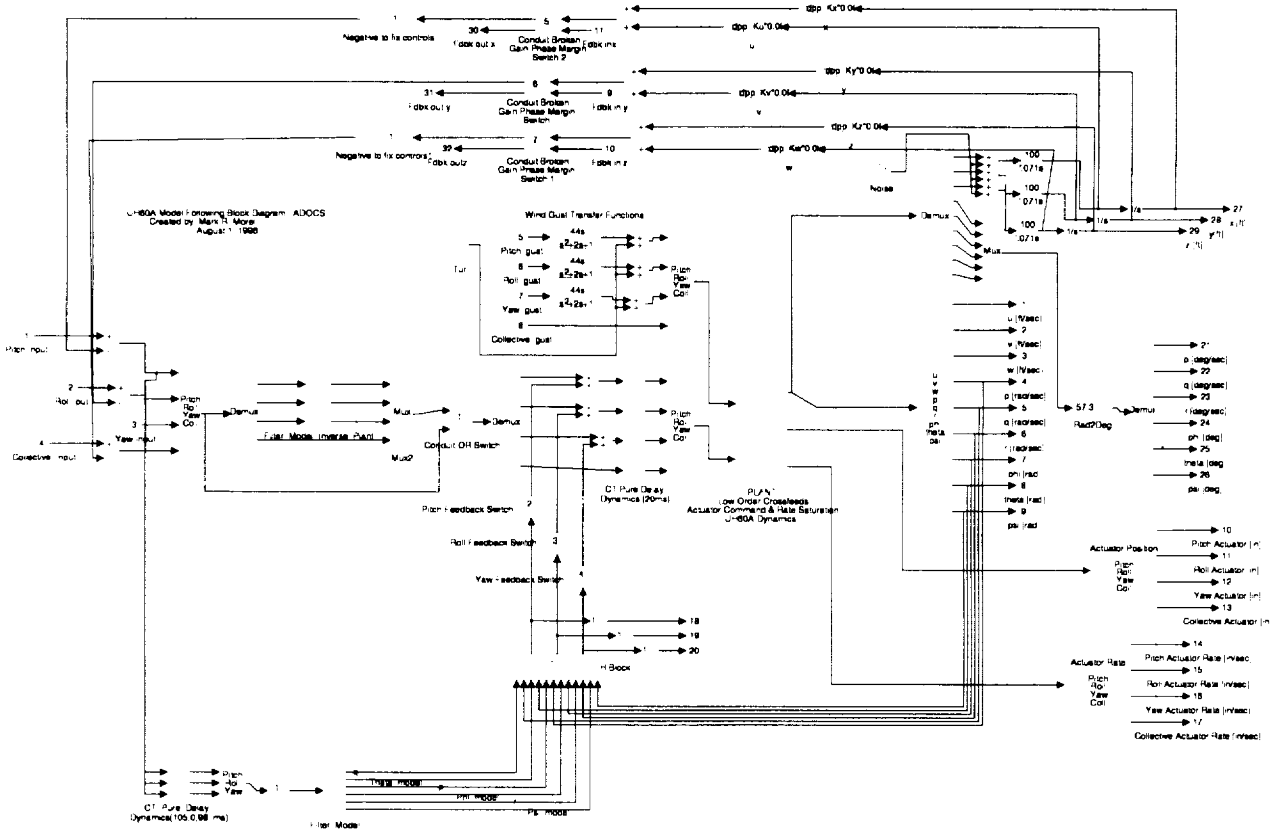


Figure 30. Position and velocity feedback loops for position hold example. Includes sensor noise and turbulence models.

Turbulence excitation was achieved 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. 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 needs to be created. This specification is shown in figure 31. It consists of two axes and a curve spline. The quarter circle radius contains the boundaries for the helicopter's root mean square (RMS) radius during the gust to that point. The RMS radius is used over maximum radius because it is more continuous since the maximum could be a spike in the position. The spec is set up in this case with the triangle (fig. 32) 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 triangle (fig. 32). This gives the radius of the circle. The axes used are x and y position of the helicopter for the first point, and y and z position of the helicopter for the second point.

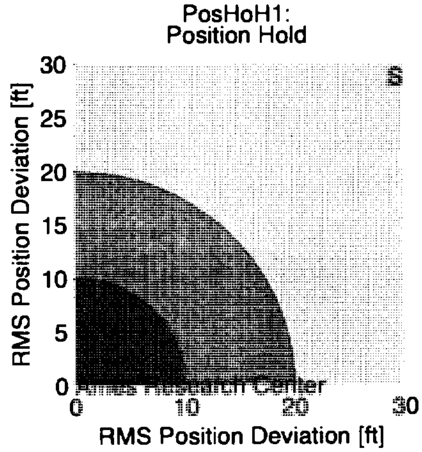


Figure 31. The position hold specification.

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

For deciding on initial gains, classical methods are used once the broken loop frequency responses are taken, then the location of 45 degrees of phase margin is read along with the required crossover frequency. Gains are found in order to drop the frequency response to the required crossover frequency.

The optimization scheme is set up to drive the RMS position within the radius of the three curves. The initial solution has the RMS value of the Level I curve of a 10 ft radius (fig. 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, which 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 (fig. 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, with the expected result of a larger RMS value for the position of the helicopter.

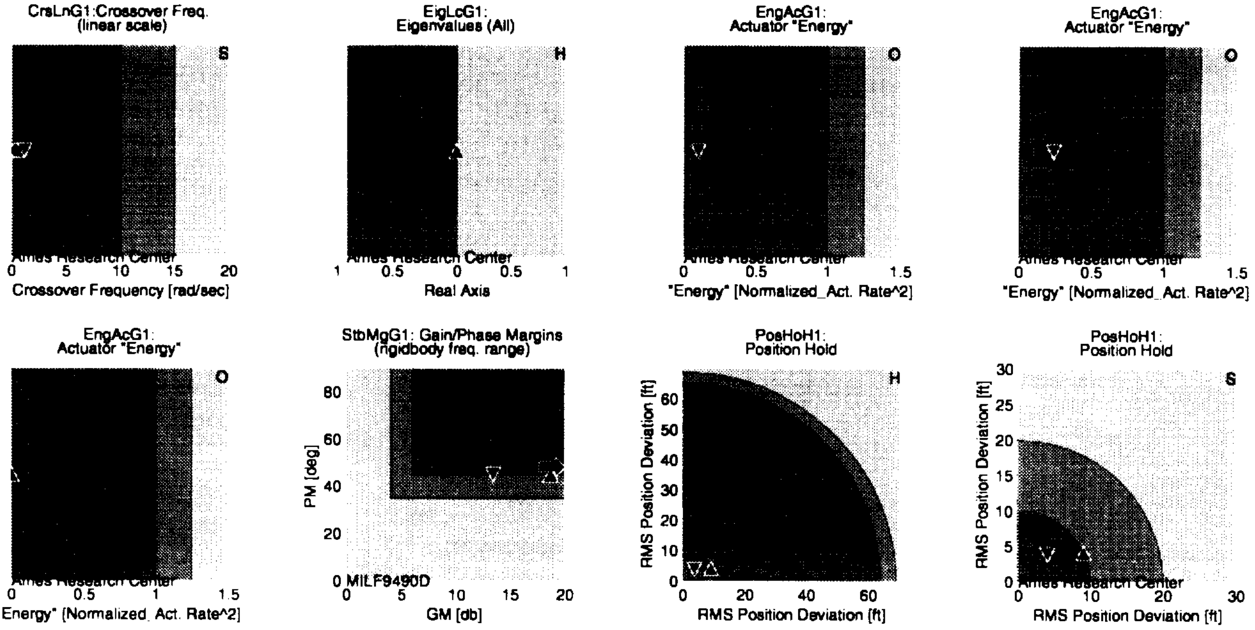


Figure 32. Wind gust solution for minimized RMS position.

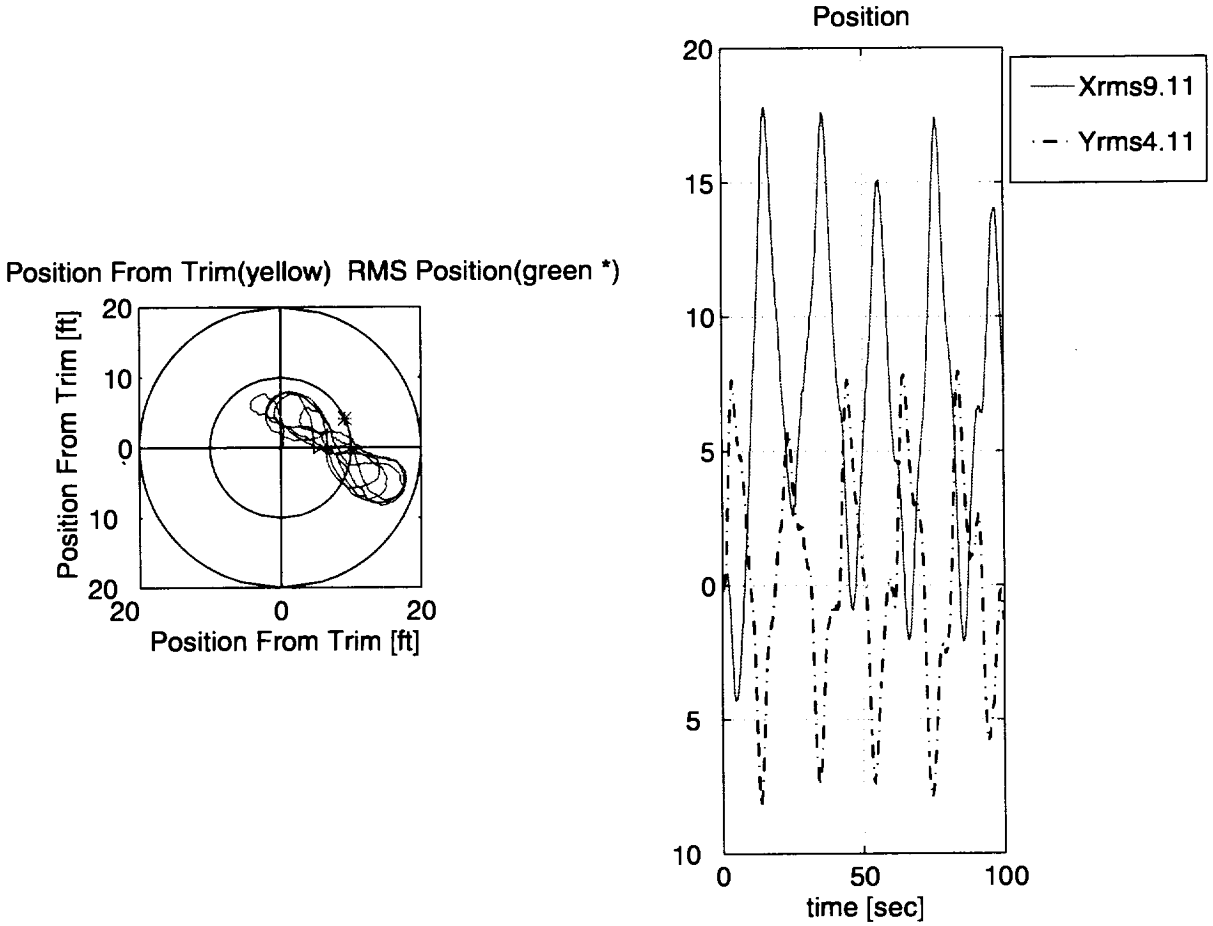


Figure 33. Time history plot of the helicopter's position for minimized RMS position.

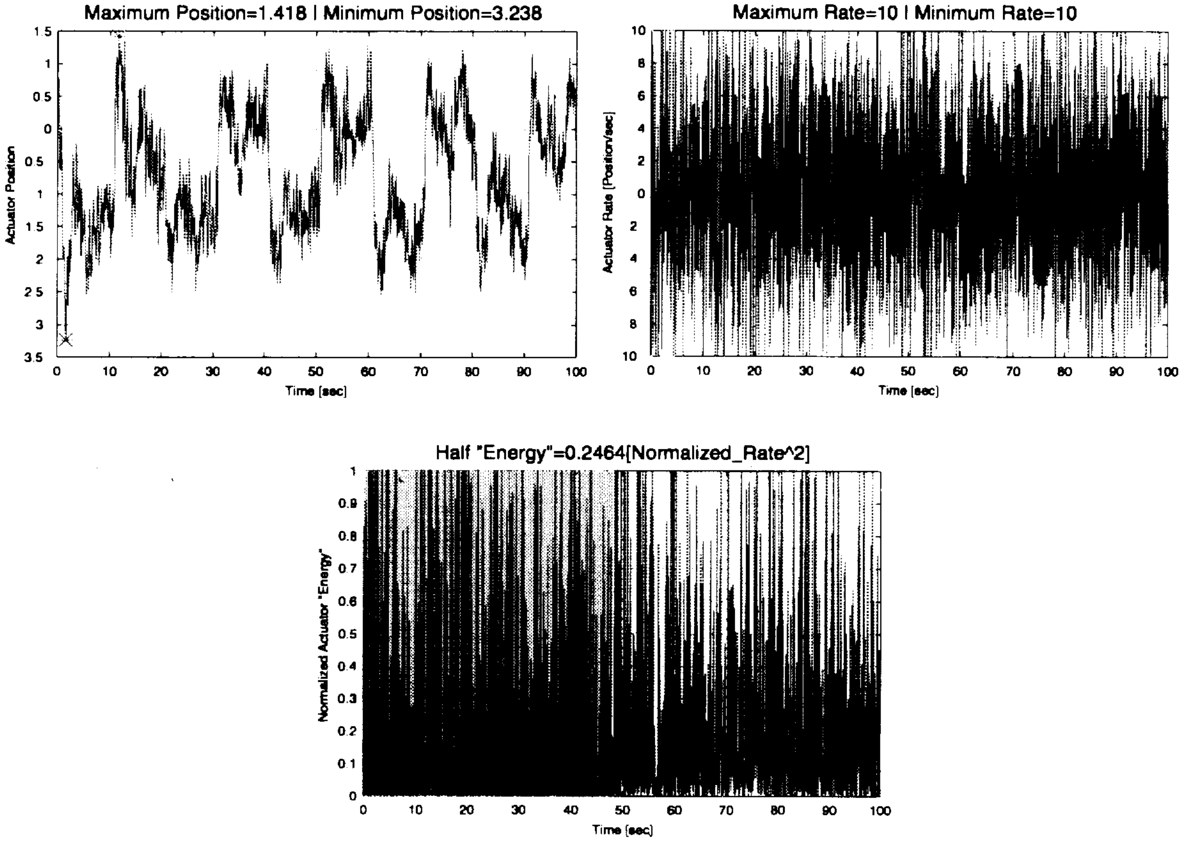


Figure 34. Supporting plot for actuator energy (roll) with saturation in the actuator rate.

The position hold specification is 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 (fig. 37). This increase in RMS value allows the actuator rate to reduce to a maximum value of 8.5 in/sec. The trade-off is becoming obvious: actuator energy for position hold. After reducing the actuator energy further (fig. 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 (fig. 39). The actuator rates again decrease, as seen in figure 40, to a level of 5.9 in/sec. The complementary nature between the actuator energy and the crossover frequency can be noticed as they both decrease during the optimizations. The lower crossover frequency acts as a filter, screening out higher frequencies of sensor noise, which drive the actuators excessively.

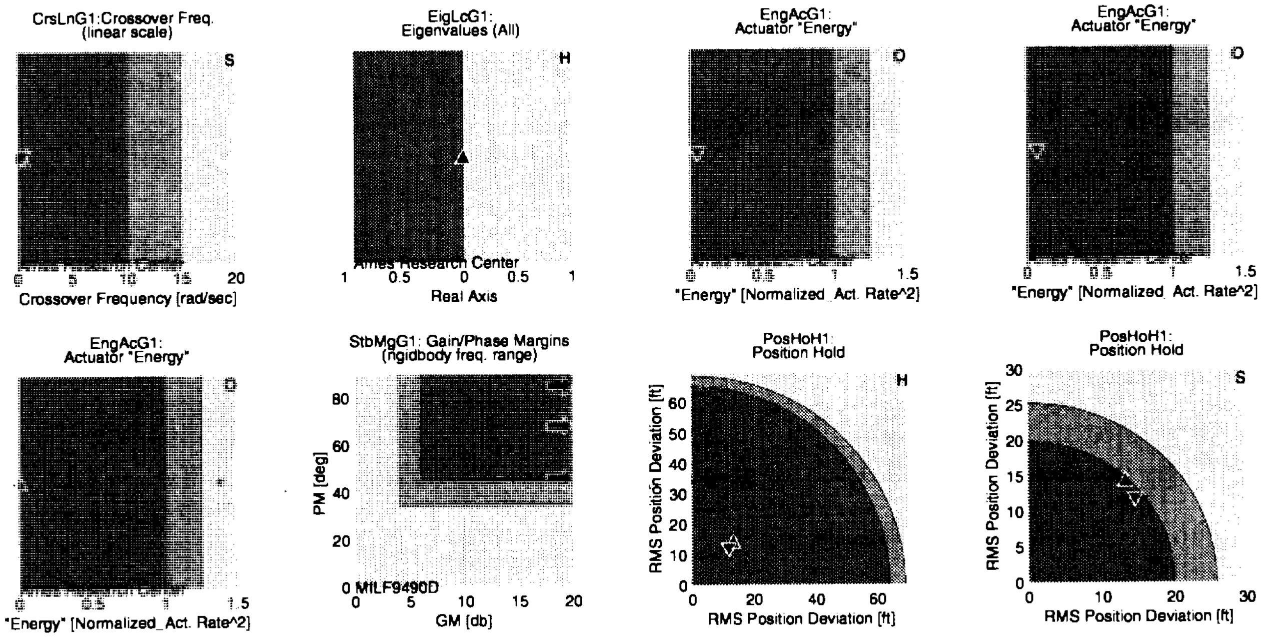


Figure 35. Medium RMS position hold solution.

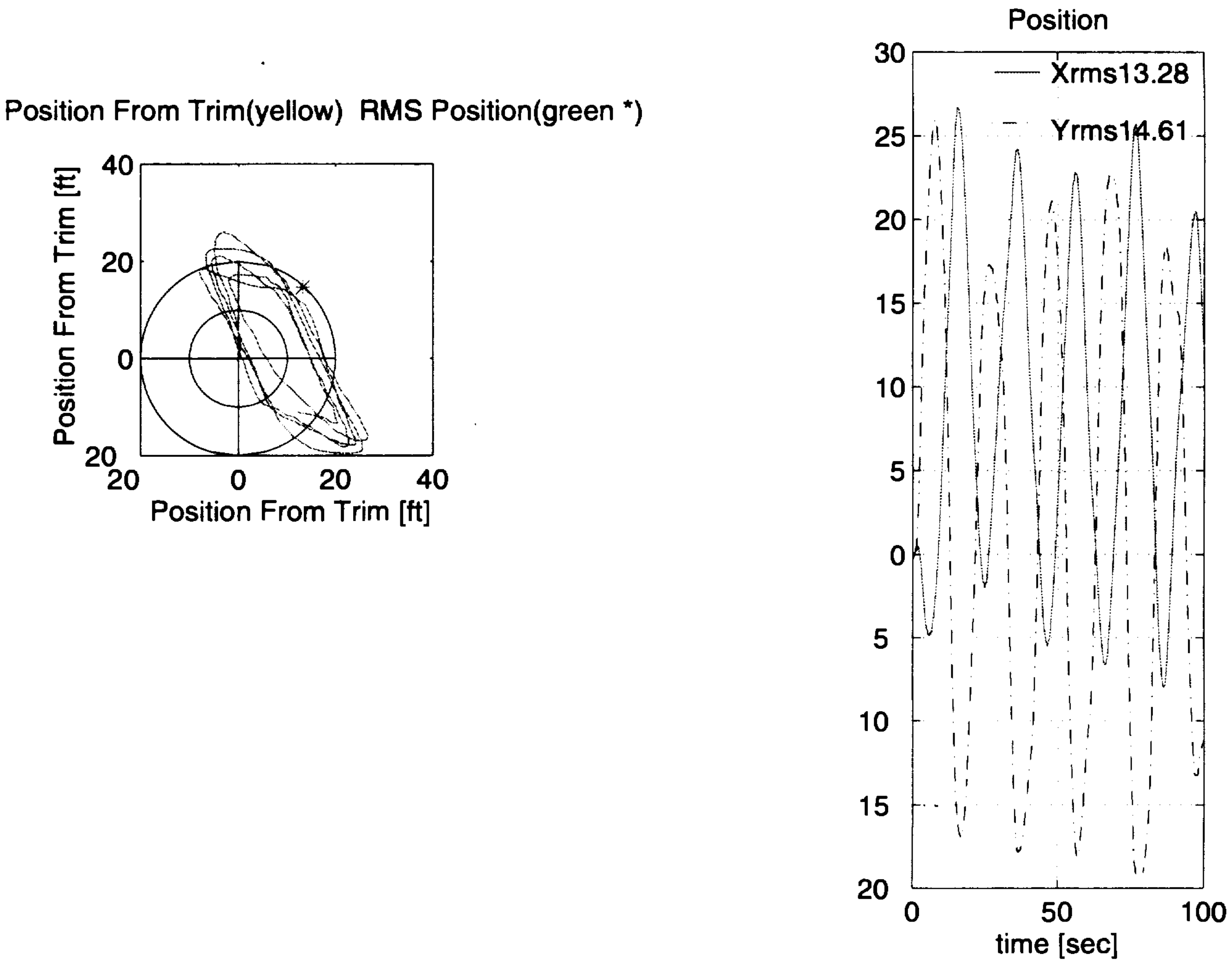


Figure 36. Time history plot of the helicopter's position for medium RMS position.

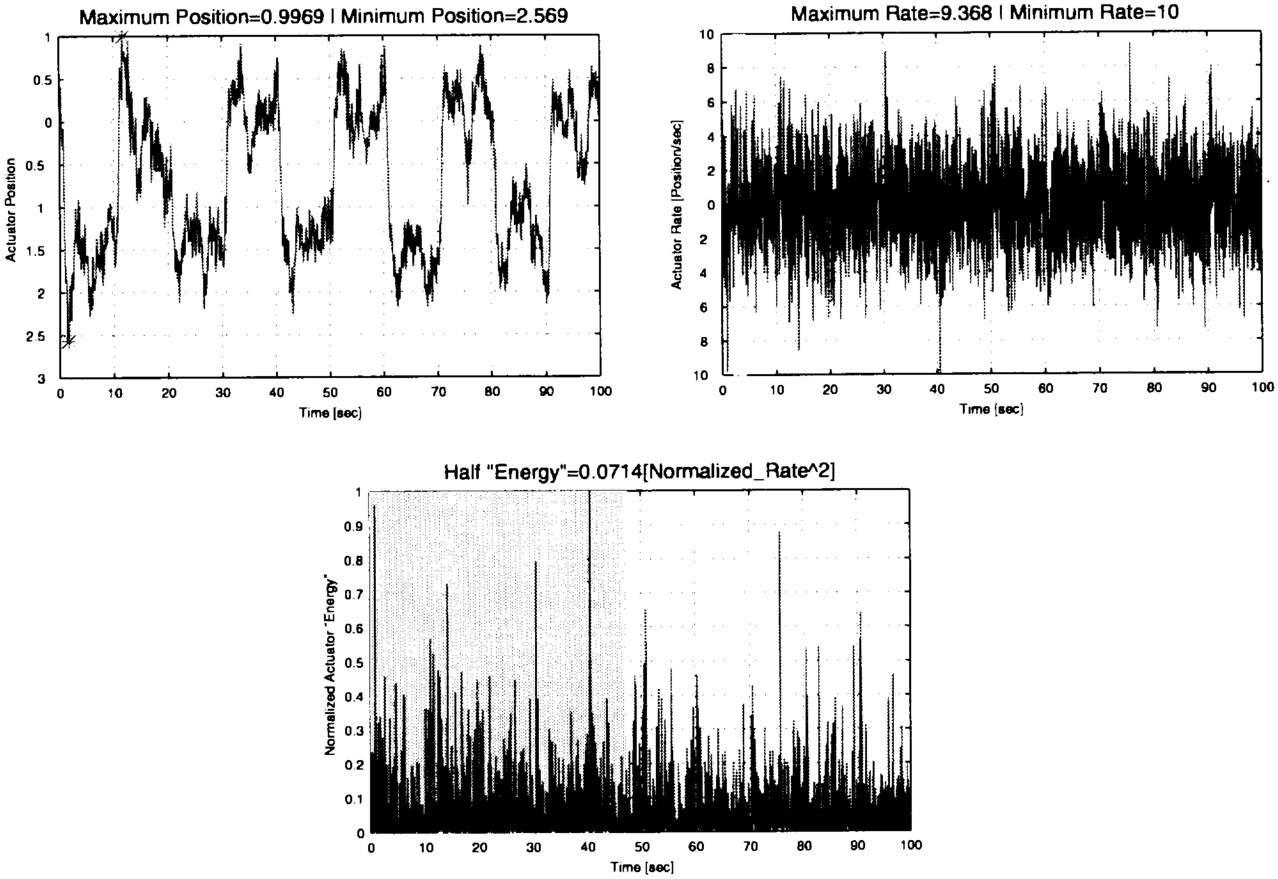


Figure 37. Supporting plot for actuator energy (roll) with saturation in the actuator rate.

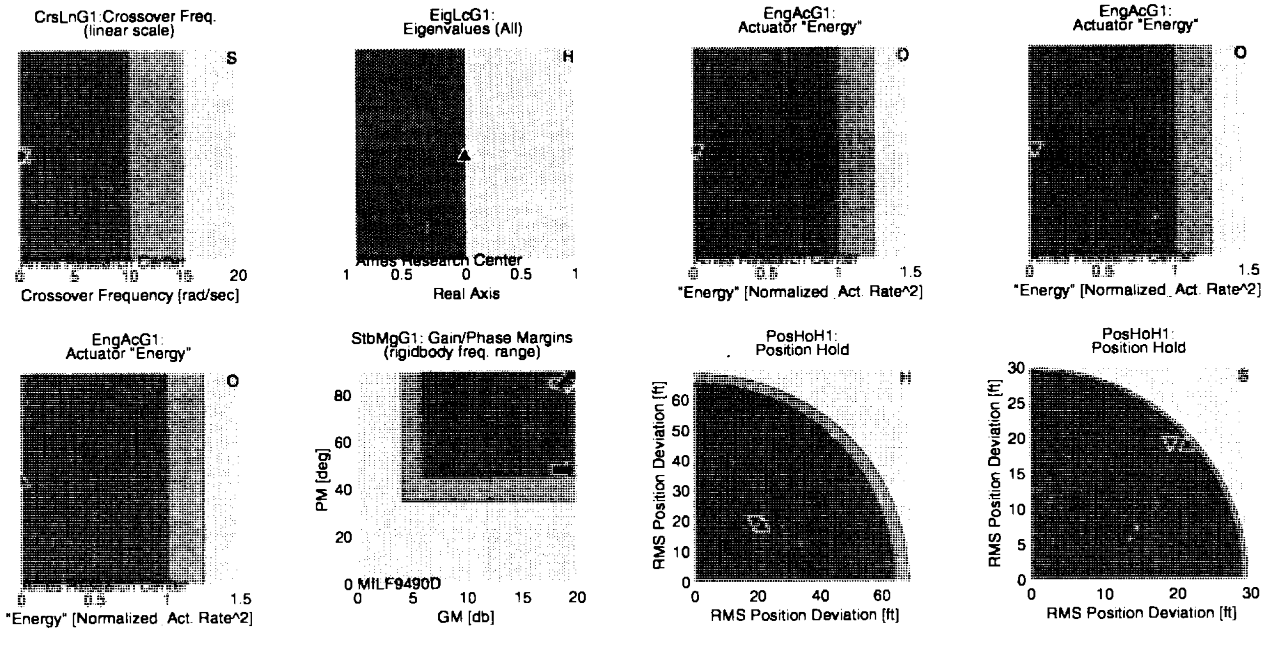


Figure 38. Large RMS position (minimized roll actuator energy).

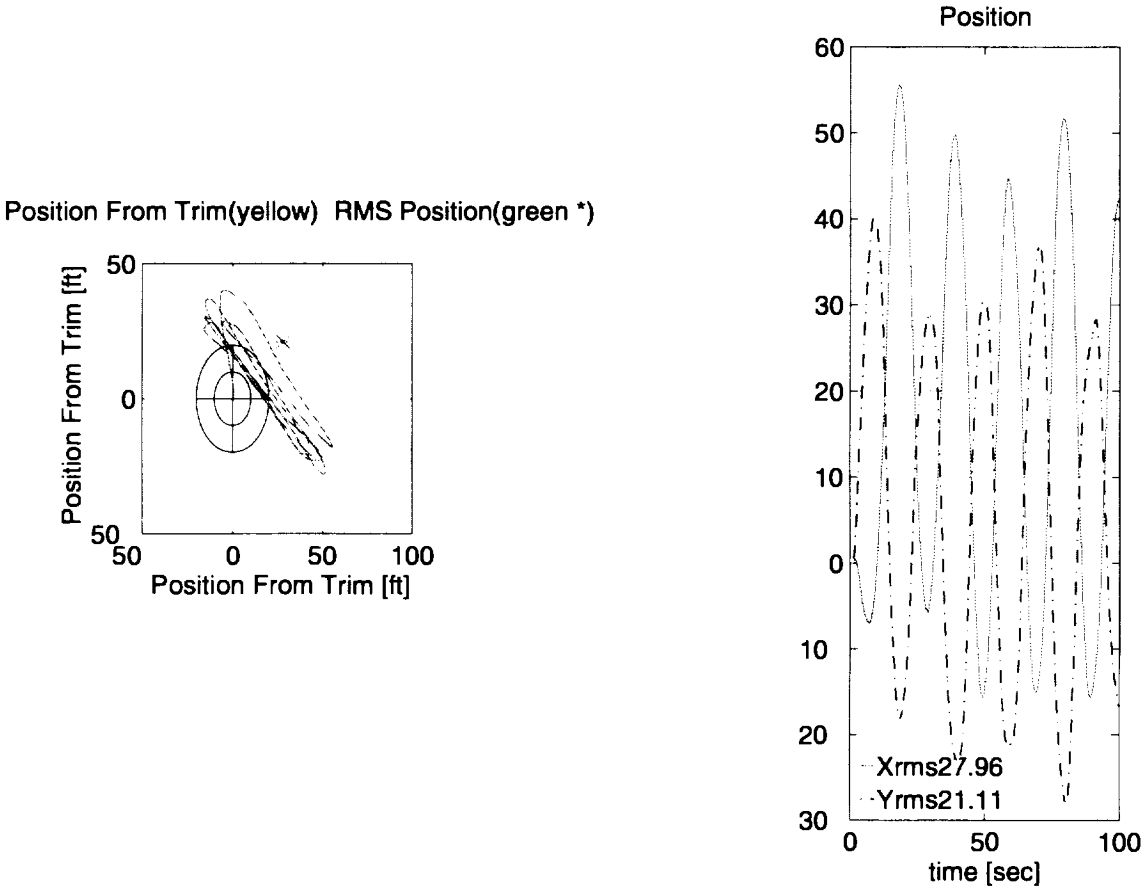


Figure 39. Time history plot of the helicopter's position for large RMS position.

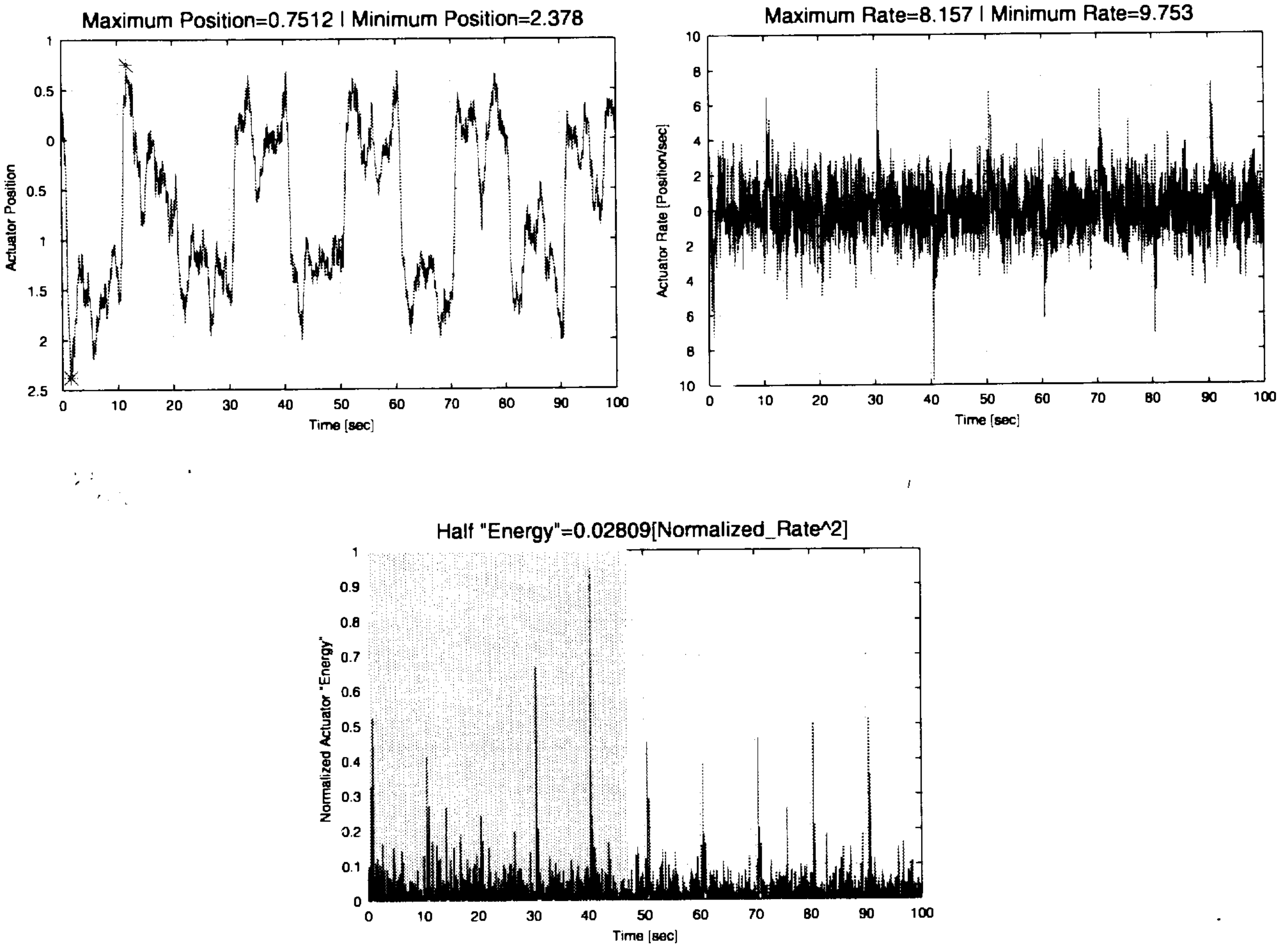


Figure 40. Supporting plot for actuator energy (roll) with saturation in the actuator rate.

The large difference between the RMS value and the maximum value seen in the last case needs 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 is demonstrated to show the trade-off between the actuator energy (and rate) and the position held by the helicopter (fig. 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. Recently, results indicating that the increase of actuator activity could be related to increased failure rate (ref. 15). Rozak was able to show a correlation between an increase in control horn fatigue damage on simulation of a Black Hawk and increased actuator activity. This study stresses the importance of evaluating and minimizing actuator energy when designing within CONDUIT.

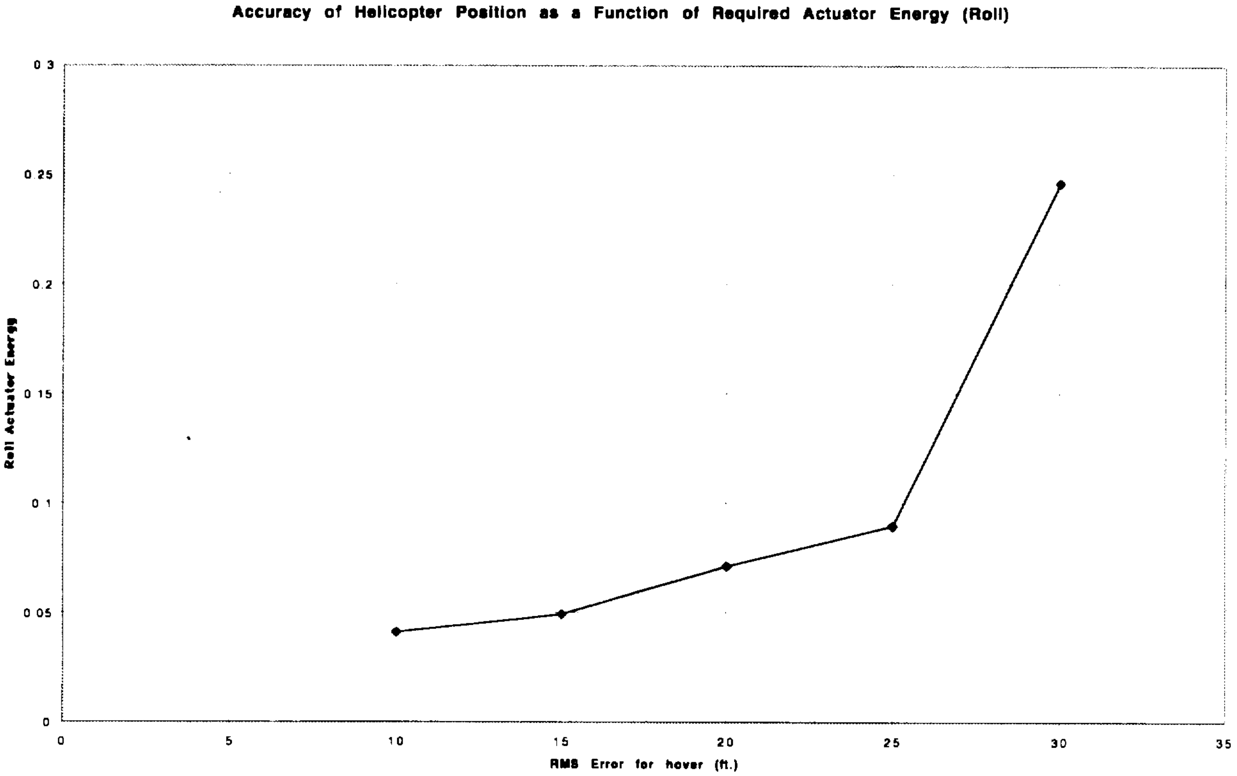


Figure 41. The trade-off between hover hold and actuator energy.

6.0 CONCLUSIONS

CONDUIT has proved to be 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, which 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 succeeds in being flexible to use any control architecture that the engineer desires. CONDUIT features 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 using a model of the Black Hawk UH-60A with the RASCAL control laws implemented were used to demonstrate the application of CONDUIT. CONDUIT showed the ability to evaluate a control system against a series of handling quality specifications in a relatively short period of time. The three problems featured in this report showed that CONDUIT can work with a sophisticated model of an aircraft and control law and optimize for 28 specifications based in time and frequency domain using closed and broken loop responses. It is hoped that industry takes advantage of this tool in designing aircraft of the future.

REFERENCES

1. United States Army Aviation Systems Command: Handling Qualities Requirements for Military Rotorcraft. ADS-33C, St. Louis, MO, Aug. 1989.

2. United States Department of Defense: Flying Qualities of Piloted Vehicles. MIL-STD-1797, Mar. 1987.

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

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

5. 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.

6. Yudelvitch, G. et. al.: Rotorcraft Flight Control System Design Based on Multi-criterion Parametric Optimization. AIAA G.N.C. Conference, Baltimore, MD, Aug. 1995.

7. 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.

8. 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.

9. Fletcher, J. W.: Identification of UH-60 Stability Derivative Models in Hover from Flight Test Data. J. Am. Helicopter Soc., vol. 40, no. 1, 1995, pp. 32–46.

10. Tischler, M. B.: Comprehensive Identification from Frequency Responses. NASA CP-10150, Sept. 1994.

11. 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.

12. Tischler, M. B.; Fletcher, J. W.; Patrick, P. M.; Morris, M.; and Tucker, G. E.: Flying Qualities Analysis and Flight Evaluation of a Highly Augmented Combat Rotorcraft. J. Guid., Control, and Dyn., vol. 14, no. 5, 1991, pp. 954–963.

13. Takahashi, M. D.: Flight Control Design Using Rotor-State Feedback for an Articulated Rotor Helicopter in Hover. NASA TM-103967, Jan. 1993.

14. 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.

15. Rozak J. N., Ray A., “Technical Note Robust Multivariable Control of Rotorcraft in Forward Flight: Impact of Bandwidth on Fatigue Life.” J. Am. Helicopter Soc., vol. 43, no. 3, 1998, pp. 195-201.

REPORT DOCUMENTATION PAGE

OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1999

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
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

5. FUNDING NUMBERS
581-30-22

6. AUTHOR(S)
Jason Colbourne*

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Aeroflightdynamics Directorate, U.S. Army Aviation and Missile Command, Ames Research Center, Moffett Field, CA 94035-1000

8. PERFORMING ORGANIZATION REPORT NUMBER
A-99V-001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001 and U.S. Army Aviation and Missile Command, St. Louis, MO 63120-1798

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA/TM-1999-208763
AFDD/TR-99-A-005

11. SUPPLEMENTARY NOTES
Point of Contact: Jason Colbourne, Ames Research Center, MS 210-5, Moffett Field, CA 94035-1000
(650) 604-6194
*California Polytechnic University, San Luis Obispo, CA 93407

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified — Unlimited
Subject Category 08 Distribution: Standard
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report details the development and use of CONDUIT (Control Designer's Unified Interface). CONDUIT is a design tool created at Ames Research Center for the purpose of evaluating and optimizing aircraft control systems against handling qualities. Three detailed design problems addressing the RASCAL UH-60A Black Hawk are included in this report to show the application of CONDUIT to helicopter control system design.

14. SUBJECT TERMS
CONDUIT, Control systems, Handling qualities, Aircraft dynamics, Computer aided control systems design, Stability and control

15. NUMBER OF PAGES
67

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT