ABSTRACT: This paper describes the development of a workstation that integrates design, analysis, and simulation methods used for flight control system synthesis. Aerodynamic, propulsion, and structural models interface directly for analysis and synthesis work. The results transfer to a flight simulator and to the dynamic structural model. The workstation is implemented with an executive, which handles most input-output operations internally, so that data management by the user is minimized. A modular approach allows new techniques to be implemented easily as executives or as additional modules.

Introduction

As computers become more powerful, engineers are applying an expanding number of design approaches to increasingly more challenging controls design tasks. The continued development of more powerful design and analysis procedures motivates the creation of a number of new software tools. Digital fly-by-wire control systems require the ability to model systems as difference equations. Future vehicles employing features such as relaxed static stability and extensive use of composite materials, resulting in greater aeroservoelastic coupling, require comprehensive analysis and evaluation of these interacting phenomena. Structural dynamic models are processed using model reduction methods to generate low-order models appropriate for the synthesis and analysis of flight control laws.

Ideally, the controls engineer should be able to access a wide range of analytical tools from a single workstation. For developing new design techniques, the aerospace engineer should be provided with an environment where new techniques may be implemented and applied easily. Aerodynamic, propulsion, and structural models should be interfaced directly both to the nonlinear dynamic simulation and to the linear design and analysis methodologies. The results of the design effort should be easily transferable both to a flight simulator and into a mathematical model of the structural dynamics. The addition of an expert-aided design capability assists the novice through the various design approaches and allows the practiced user to take full advantage of the workstation.

Since the late 1970s, the Lockheed Flight Controls Research Department has been developing modern controls techniques specifically for aircraft control system design work [1]-[3]. Routines for the singular-value analysis of multivariable systems have also been developed and evaluated. Much work also has been done over the last 10 years in the area of flying-qualities prediction, for example, by equivalent systems methods [4].

Given these various developments, the need was seen for a single flight controls workstation that would include all of these design tools within one integrated package [5]. With this approach, time-consuming data management is reduced greatly. The workstation handles input-output operations internally and displays a list of the next logical steps in the analysis process. The user is directed toward the analysis methods of interest using a menu-driven hierarchy. Most results are displayed graphically to the user to maximize data inflow. The workstation is shown in Fig. 1. An expert system capability can be added to this system with only minor modifications to the basic workstation. To be implemented is an expert-aided approach by which predicted flying qualities are used to evaluate the performance of the closed-loop system. If the desired flying qualities are not attained, a rule base would advise the user on how to modify the flight control system.

System Design and Analysis

To properly assemble any interactive computer-aided design package, especially an application using an "expert system" approach, the design process for the application must be thoroughly understood. Flight controls design is a highly interdependent process, which eventually incorporates linear and nonlinear analysis, real-time simulation, and ground and flight testing. This dictates that any comprehensive flight control design package must be highly interactive in nature and must provide input-output facilities to exchange data and other information with associated disciplines. The autopilot design process given in Fig. 2 provides a tailored version of this design methodology. Design work progresses from the definition of the basic architecture using mission and functional requirements analysis to linear analysis, nonlinear simulation, logic simulation, piloted simulation, and finally to flight test. As the process goes from one progressively more complex step to another, few of the previous models are abandoned. Instead, the results of the new analysis are incorporated into the other models.

Analysis Tools

In support of the design methods outlined in the previous section, several analysis pro-

Presented at the 1988 American Control Conference, Atlanta, Georgia, June 15-17, 1988. Richard Dean Colgren is with the Lockheed Aeronautical Systems Company, P.O. Box 551, Burbank, CA 91520.
grams have been developed at Lockheed or have been obtained from outside sources. A complete list of these programs cross-referenced against their design function is given in the Table. The source for each program, whether it was developed or enhanced by Lockheed or purchased from a vendor, is also indicated on this table. Another package not listed here, the Flutter And Matrix Algebra System (FAMAS) [6], is a library of matrix algebra routines used by the structural dynamics group to generate the input model used by the MODEL R [7] state-space model reduction programs.

In the Table, the arrows indicate a process that converts the system from its original form into one covered by another function. For example, SIMULT [8] can be used to analyze the frequency response of continuous, discrete, and sampled-data systems. The equivalent systems method used generates a low-order continuous transfer function with a time delay from these frequency responses.

Arrows are used to show that discrete and sampled-data systems are converted into continuous systems, and nonlinear models into linear models, by the program.

**Flight Controls Workstation**

The workstation provides the user with classical and modern linear controls design techniques, as well as nonlinear simulation capabilities. Steady and unsteady aerodynamic data, engine models, and dynamic structural models are available to the nonlinear simulation. The dynamic structural model can be reduced using frequency-response techniques to provide a model more suited for controls augmentation work. The nonlinear simulation model is then linearized automatically for analysis and synthesis using classical and optimal control techniques, eigenstructure placement, and equivalent systems analysis. A matrix algebra tool set is also included. It provides an environment in which new analysis methods may be implemented and evaluated easily. The workstation provides an interactive, user-transparent interface between the various design tools. This greatly increases the accuracy of the design process by eliminating the manual transfer of data between analysis programs.

A functional schematic of the flight controls workstation, which incorporates the analysis tools described in the previous section, is given in Fig. 3. It is designed to support the flight control system evaluation and synthesis process. The engine model and low-frequency nonlinear aerodynamic data are accessed directly by the nonlinear simulation modeled using the Advanced Continuous Simulation Language (ACSL) [9]. The mathematical model of the structure is generated using FAMAS, a structural dynamics modeling program, to which the effects of unsteady aerodynamics are added (Fig. 4). A least-squares routine written for FAMAS generates a state-space model of very high

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>CONTROL</th>
<th>ACSL</th>
<th>SIMULT</th>
<th>MODEL R</th>
<th>EIGEN</th>
<th>ORACLS</th>
<th>MATLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINEAR ANALYSIS</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>NONLINEAR ANALYSIS</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>CONTINUOUS SYSTEMS</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>DISCRETE SYSTEMS</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>MIXED SYSTEMS</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>TRANSFER FUNCTION MODEL</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>STATE-SPACE MODELS</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>LINEARIZE SYSTEM</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>EIGENVALUES</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>ROOT LOCUS, FREQUENCY</td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>TRANSIENT RESPONSE</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>DERIVE TRANSFER FUNCTION</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>'EQUIVALENT' LOW-ORDER TRANSFER FUNCTION</td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'EQUIVALENT' LOW-ORDER STATE-SPACE MODEL</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIGENSTRUCTURE PLACEMENT</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>OPTIMAL FEEDBACK MATRIX</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>MATRIX ALGEBRA</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>STATISTICAL ANALYSIS</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

Note: SIMULT and EIGEN are programs developed by Lockheed, MODEL R has been enhanced by Lockheed, and the remainder are commercially available programs.
order. This model is then preprocessed and balanced using the MODEL R programs. It is reduced via truncation or residualization into a low-order model, which preserves the input-output behavior throughout the frequency range of interest [7], [10]. The linearized model, generated by either ACSL or MODEL R, can be used by the various linear analysis programs. Mathematical models of the structure can also be included within the nonlinear simulation.

The linearized model is made available to the linear analysis programs shown on the lower portion of Fig. 3. This state-space model can represent either the open-loop aircraft or the closed-loop system using control laws developed during previous design steps. During early phases of the design process, the aircraft and engine models can be entered directly as stability derivatives into CONTROL [11], [12]. Then an augmented state-space model including servos and filters is generated for use by the other linear analysis programs. This is done by dividing the input portion of CONTROL into several sections, as shown in Fig. 5. The first is a parameters section, which is used to enter all data constants into the program. The second is an auxiliary equations section, which follows standard FORTRAN expression evaluation rules. It can model nonlinear elements and gain schedules and assign the linear or trim values to those elements during a pre-processing "translation" phase. An equations section is used to model block diagram components as s-, z-, or w-plane transfer functions. A matrix section accepts either the linearized models or externally generated state-space models. By using variable names from the auxiliary equations or parameters section in coding the matrices, the state-space model can be generated during the translation phase.

Modern control design methods can be accomplished using MATLAB [13], EIGEN [1], or ORACLS (Optimal Regulator Algorithms for the Control of Linear Systems) [14]. MATLAB is an interactive program for computations involving matrices. EIGEN is a program for generating the output feedback matrix using eigenstructure placement. ORACLS is a library of subroutines for optimization problems using linear-quadratic methods. Much work is being done on ap-

---

**Fig. 3. Functional diagram of the flight controls workstation.**

**Fig. 4. Aeroservoelastic modeling.**

**Fig. 5. Control explicit program structure.**
plying these modern control techniques to flight controls design. The modern controls method emphasized at Lockheed is eigenstructure placement, as it provides for partial state feedback. Previously defined loop closures can be fixed while “optimizing” additional loop closures.

Classical analysis and nonlinear simulation remain the primary design and analysis procedures. Most flight control systems have been designed using “classical” methods, thus an extensive data base exists for this approach. Frequency responses for the augmented system are used to generate equivalent low-order transfer functions of the form given in MIL-STD-1797. They can be used either to show compliance with the military specifications or to predict flying qualities.

Expert-Aided Approach

As has been shown, a number of design methods and software packages are available to the engineer to solve increasingly complex design problems. As the number, capabilities, and complexity of these tools increase, it becomes difficult for the designer to effectively apply the available range of techniques and software. Also, the proficiency of the user varies widely. The addition of an expert-aided system to this workstation will assist the user in taking full advantage of these capabilities. It will also reduce man-hours and computer time, while improving flying qualities for enhanced aircraft performance, reduced pilot workload, and improved mission effectiveness.

For example, an approach to a flying-qualities-based system might be the following. The model of the vehicle and its flight control system are provided by the user. If equivalent system models are required, the matching rule base guides the user through the model generation process. Once completed, the relevant flying-qualities parameters are calculated. Criteria from MIL-STD-1797, Neal-Smith, bandwidth, or other methods then can be compared with these parameters, and flying-qualities levels predicted. The results are interpreted by the requirements rule base. If the desired flying qualities are not achieved, or further analysis is desired, the control synthesis rule base advises the user on how to proceed. The process is iterated until the user’s requirements are achieved. This feature is currently under development.

Conclusion

A workstation for the design of flight control systems has been presented. Classical control design techniques and nonlinear simulation are supported, along with modern linear controls design techniques. Equivalent systems methods are available for flying-qualities prediction and evaluation. Interfaces are provided to exchange information between related disciplines. The modular approach to the implementation of the workstation will allow an expert system to be added.

References
