Workstation-Based Simulation for Rapid Prototyping and Piloted Evaluation of Control System Designs

M. Hossein Mansur
Army/NASA Rotorcraft Division
Aeroflightdynamics Directorate (AVRDEC)
U.S. Army Aviation and Missile Command
Ames Research Center, Moffett Field, CA 94035-1000, USA

Jason D. Colbourne
Cal Poly Foundation
California Polytechnic State University,
San Luis Obispo, CA 93507

Yu-Kuang Chang
Raytheon STX
Ames Research Center
Moffett Field, CA 94035-1000

The development and optimization of flight control systems for modern fixed- and rotary-wing aircraft consume a significant portion of the overall time and cost of aircraft development. Substantial savings can be achieved if the time required to develop and flight-test the control system, and the cost, is reduced. To bring about such reductions, software tools such as Matlab/Simulink [ref. 1] are being used to readily implement block diagrams and rapidly evaluate the expected responses of the completed system. Moreover, tools such as CONDUIT (CONtrol Designer's Unified InTerface) [ref. 2] have been developed that enable the controls engineers to optimize their control laws and ensure that all the relevant quantitative criteria are satisfied, all within a fully interactive, user friendly, unified software environment.

Modern control systems regularly employ multiple selectable modes as well as gain scheduling, switching logic, and failure compensation schemes. Evaluation of these additional components constitutes an important and time-consuming aspect of the flight control system development process and cannot be conducted using tools such as Matlab/Simulink or CONDUIT. The reason is that credible dynamic transition between the modes and through the various schedules and logic can only be achieved through piloted simulation using a high-fidelity, non-linear, mathematical model of the vehicle.

Control design tools and non-linear mathematical models have historically not been designed to work together seamlessly. Therefore, piloted simulation evaluation of control law designs has always involved a complex and lengthy integration process. Rapid prototyping, whereby ideas are quickly implemented and tested without significantly lengthening the development cycle, has therefore not been possible. Engineers at the NASA/Army Rotorcraft Division at the Ames Research Center have developed a workstation-based simulation environment that significantly shortens the integration process and allows control law designs to be quickly tested in piloted simulation. This simulation environment is known as the Real-time Interactive Prototype Technology Integration/Development Environment, or RIPTIDE.

RIPTIDE offers the controls engineer the ability to readily evaluate each control law design in real time piloted simulation. It is not intended to be a substitute for very high-fidelity simulators, such as NASA Ames' Vertical Motion Simulator (VMS), with state of the art graphics and possibly motion. However, in most instances the state of the art fidelity of such simulators is not necessary and a good workstation-based simulator is all that is required. The advantages of a workstation-based simulator such as RIPTIDE are low cost, high availability, low overhead, and portability. Since RIPTIDE is hosted on a single graphics workstation, its hardware requirements are not enormous. Also, the use of a single workstation provides for better availability due to minimal overhead. Finally, the software can be used on similar workstations, allowing many organizations to obtain their own simulation capability.

RIPTIDE is a departure from the traditional methods of piloted simulation in that it employs an open software architecture. None of the components of the simulation are specified or fixed in advance and, therefore, any module can be added or modified without a need to modify the rest of the simulation environment. This modular architecture shortens the time required to evaluate multiple designs and significantly reduces development time. Each component of RIPTIDE is implemented as an independent process, which does not reply on any other component for its operation. To make this process independence possible, transfer of physical parameters, such as aircraft states and control inputs, is accomplished via shared areas of computer memory (figure 1), as opposed to passing of parameters between subroutines. Even timing and execution order is component independent and is controlled via a process scheduler that ensures that each component is executed at the appropriate time and that all processes are executed within the time allotted to insure real time operation.

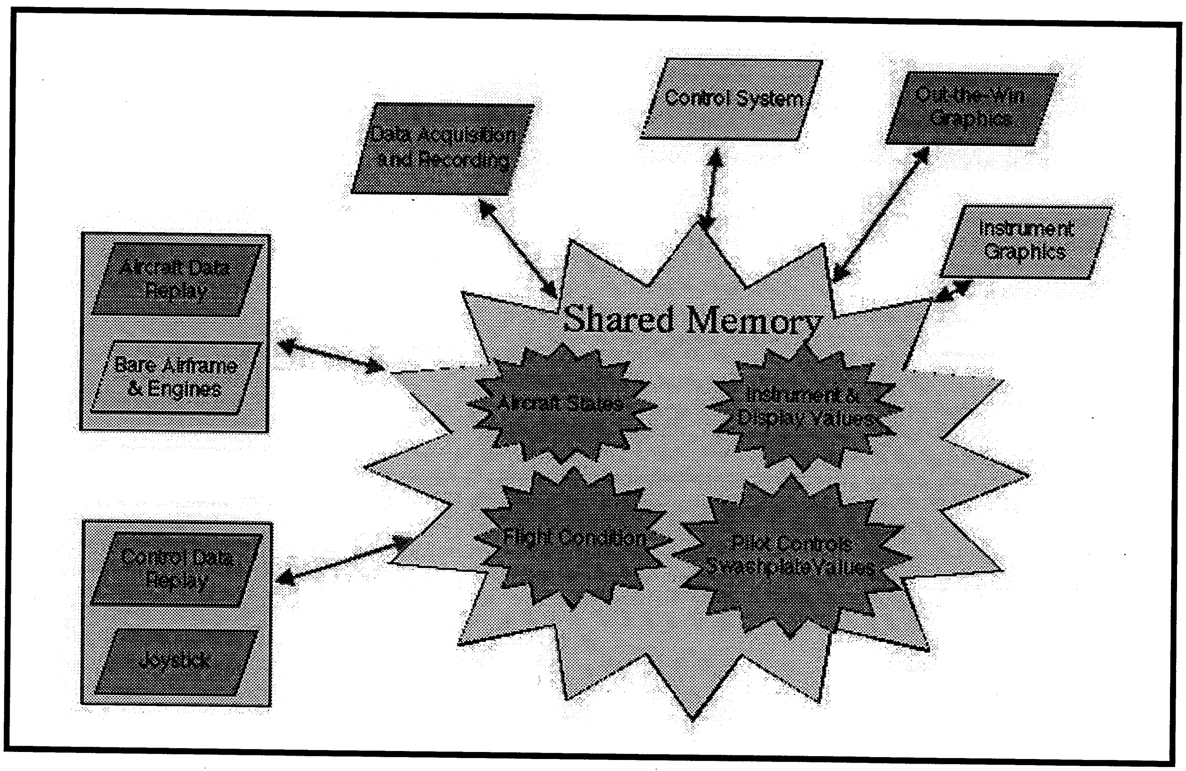

Figure 1: Shared memory used for inter-process communication

RIPTIDE can be easily coupled with tools such as Matlab/Simulink and CONDUIT. The code generation capabilities of Real Time Workshop can be employed to convert any set of block diagrams into code that can be executed within RIPTIDE. The requirement for input/output from/to shared memory is satisfied using specialized blocks, developed in Simulink itself, that accomplish these tasks. In this way, the integration process is all but

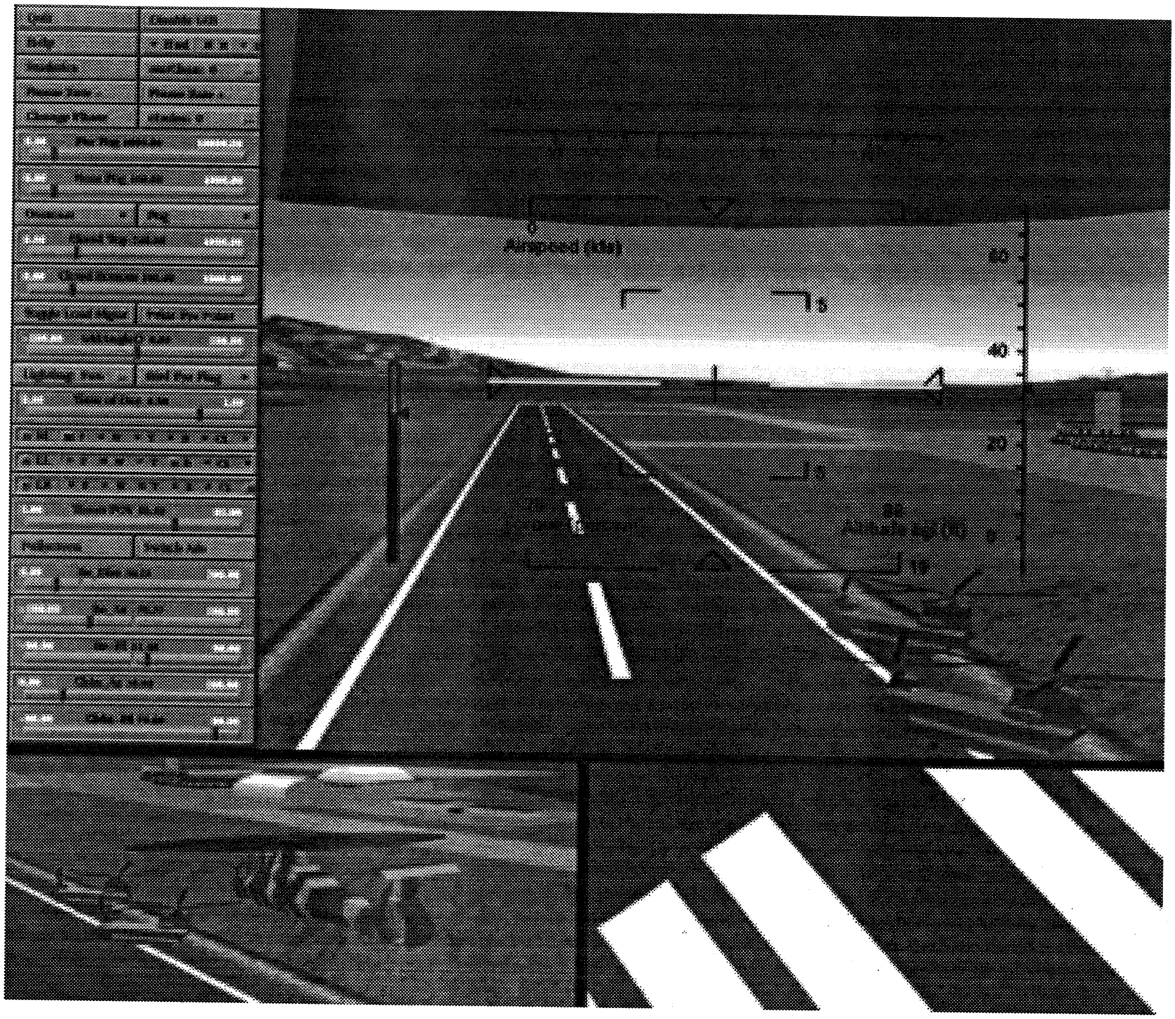

Figure 2: RIPTIDE out-the-window graphics and GUI

eliminated and the time between the availability of a control law version and its evaluation in piloted simulation is reduced to minutes.

RIPTIDE is currently being used to implement and evaluate the baseline control laws for the Rotorcraft Aircrew Systems Concepts Airborne Laboratory (RASCAL) UH-60 helicopter. The control laws are implemented as Simulink block diagrams and converted to executable code using Real Time Workshop. Specialized Simulink blocks are used to enable the resulting code to communicate with the RIPTIDE shared memory. A high-fidelity mathematical model of the UH-60 aircraft, based on the Gen Hel model developed by Sikorsky [ref. 3] and modified/enhanced at Ames [ref. 4], is implemented and provides a nonlinear representation of the RASCAL aircraft.

The proposed paper will describe the RIPTIDE simulation environment in detail, including the scene and instrument display capabilities (figure 2), pilot controller, and math model/control law integration process. The major uses of the environment will be discussed, including:

1. Piloted Simulation
2. Rapid prototyping and evaluation of control laws
3. Evaluation of vehicle mathematical models
4. Evaluation of glass cockpit displays
5. Flight data visualization using high-fidelity graphics
6. Graphical comparison of flight and simulation data

The applicability of RIPTIDE to the rapid evaluation of control system designs in piloted simulation will be demonstrated. As an example, results of piloted evaluations of the RASCAL baseline control laws in the RIPTIDE environment will be presented. Finally, a short video will be shown to better demonstrate the graphical capabilities of the system and the level of realism achievable.

References

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

2. Tischler, M. B., Colbourne, J. D., Morel M. R., Biezad, D. J., Levine, W. S., and Moldoveanu, V., "CONDUIT – Anew Multidisciplinary Integration Environment for Flight Control Development," *AIAA Paper 97-3773*, August 1997.

3. Howlett, J. J., "UH-60A Black Hawk Engineering Simulation Program: Volume I -- Mathematical Model", NASA CR-166309, Dec. 1981.

4. Ballin, M. G., and Dalang-Secretan, M. A., "Validation of the Dynamic Response of a Blade-Element UH-60 Simulation Model in Hovering Flight," American Helicopter Society 46th Annual Forum, Washington D.C., May 1990.