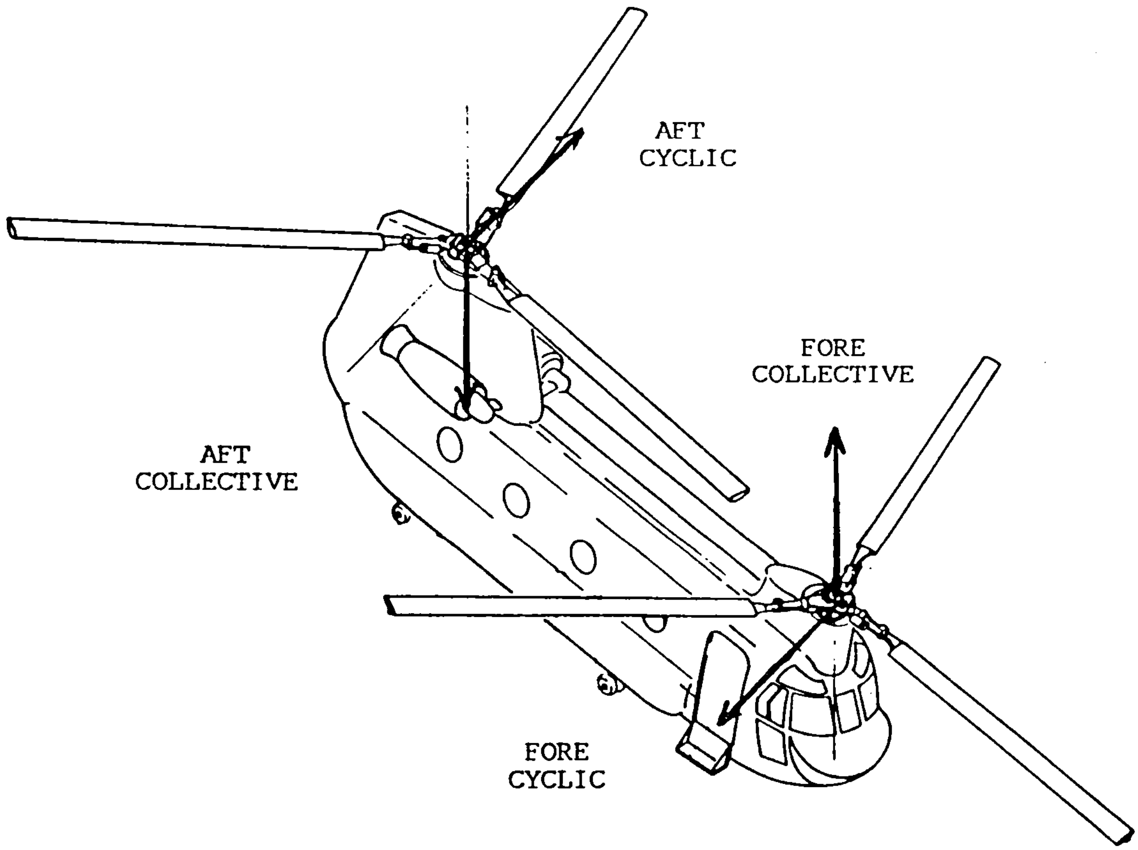
RULE-BASED FAULT-TOLERANT FLIGHT CONTROL

Dave Handelman
Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey

APPLICATION VEHICLE: CH-47

Fault tolerance has always been a desirable characteristic of aircraft. The ability to withstand unexpected changes in aircraft configuration has a direct impact on the ability to complete a mission effectively and safely. The objective of this research was to investigate possible synergistic effects of combining techniques of modern control theory, statistical hypothesis testing, and artificial intelligence in the attempt to provide failure accommodation for aircraft. This effort has resulted in the definition of a theory for rule-based control and a system for development of such a rule-based controller. Although presented here in response to the goal of aircraft fault tolerance, the rule-based control technique is applicable to a wide range of complex control problems.



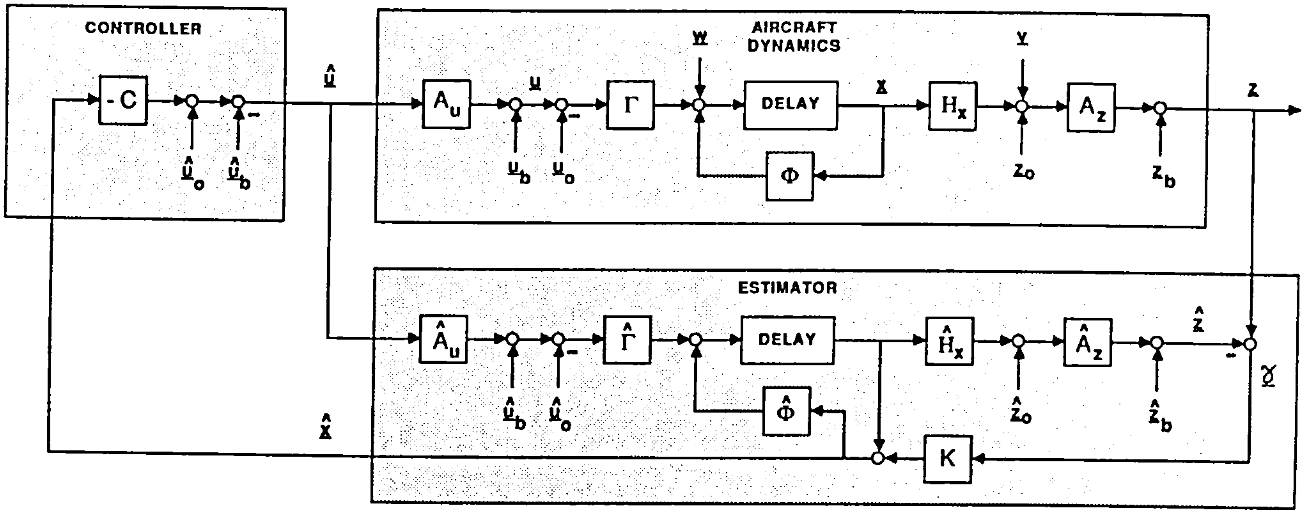
FAILURE MODES

The flight control system is expected to handle single abrupt and significant failures in aircraft sensors, controls, and structure. Biased, attenuated, and stuck sensors and controls, as well as structural failures in the form of center-of-gravity shifts, are to be detected, diagnosed, and accommodated automatically by the control system.

COMPONENT TYPE	ELEMENT	MODE
SENSOR	LONGITUDINAL VELOCITY	BIASED
	LATERAL VELOCITY	ATTENUATED
	VERTICAL VELOCITY	STUCK
	ROLL RATE	
	PITCH RATE	
	YAW RATE	
	PITCH ANGLE	
	ROLL ANGLE	
CONTROL	FORWARD CYCLIC PITCH	BIASED
	FORWARD COLLECTIVE PITCH	ATTENUATED
	AFT CYCLIC PITCH	STUCK
	AFT COLLECTIVE PITCH	
STRUCTURE	CENTER OF GRAVITY	SHIFTED

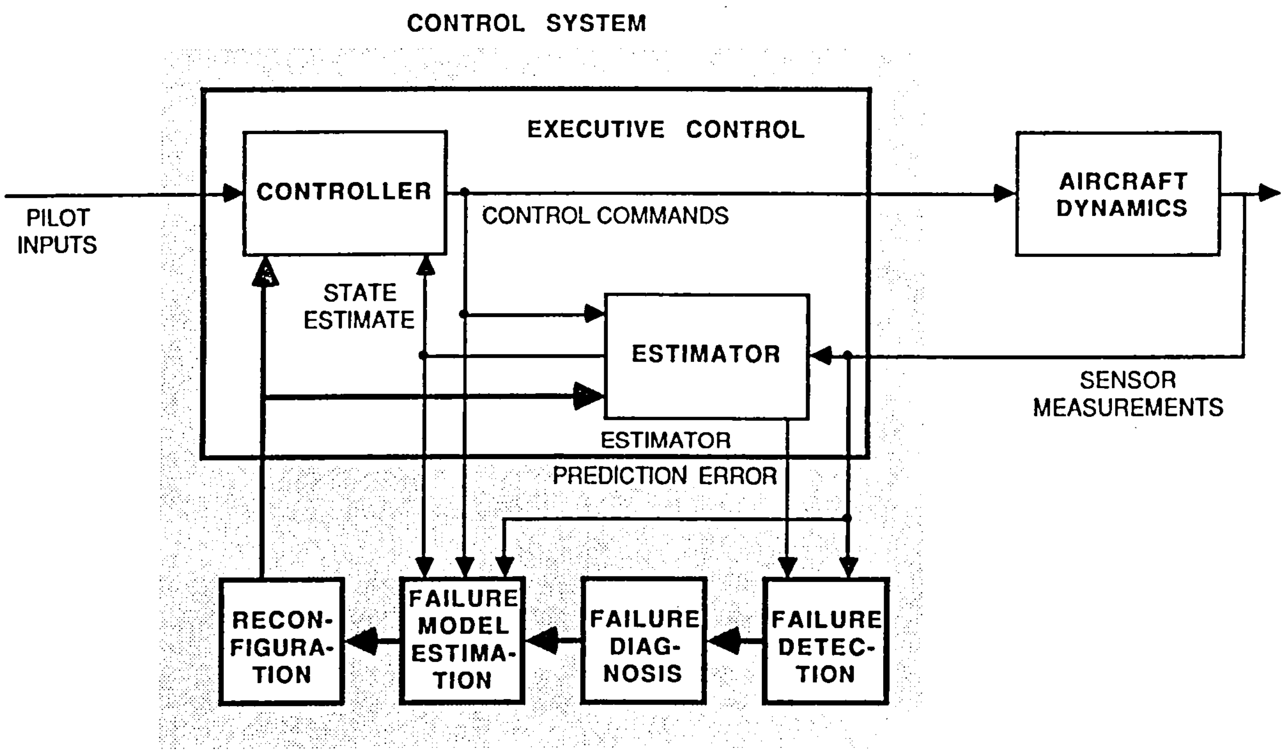
LINEAR MODEL OF AIRCRAFT DYNAMICS, ESTIMATOR, AND CONTROLLER WITH FAILURE-MODEL VECTORS AND MATRICES

The fault-tolerant controller is designed to regulate aircraft motion about a constant flight condition. Assuming linearity within a neighborhood of this nominal operating point, a state-space mathematical model approximates the aircraft dynamics. Based on this no-failure model, a Kalman Filter is used for state estimation, and a Linear-Quadratic Regulator is used for feedback control calculations. A failure is assumed to change significantly the mathematical model representing the actual aircraft dynamics, prompting the need for estimator and regulator reconfiguration.



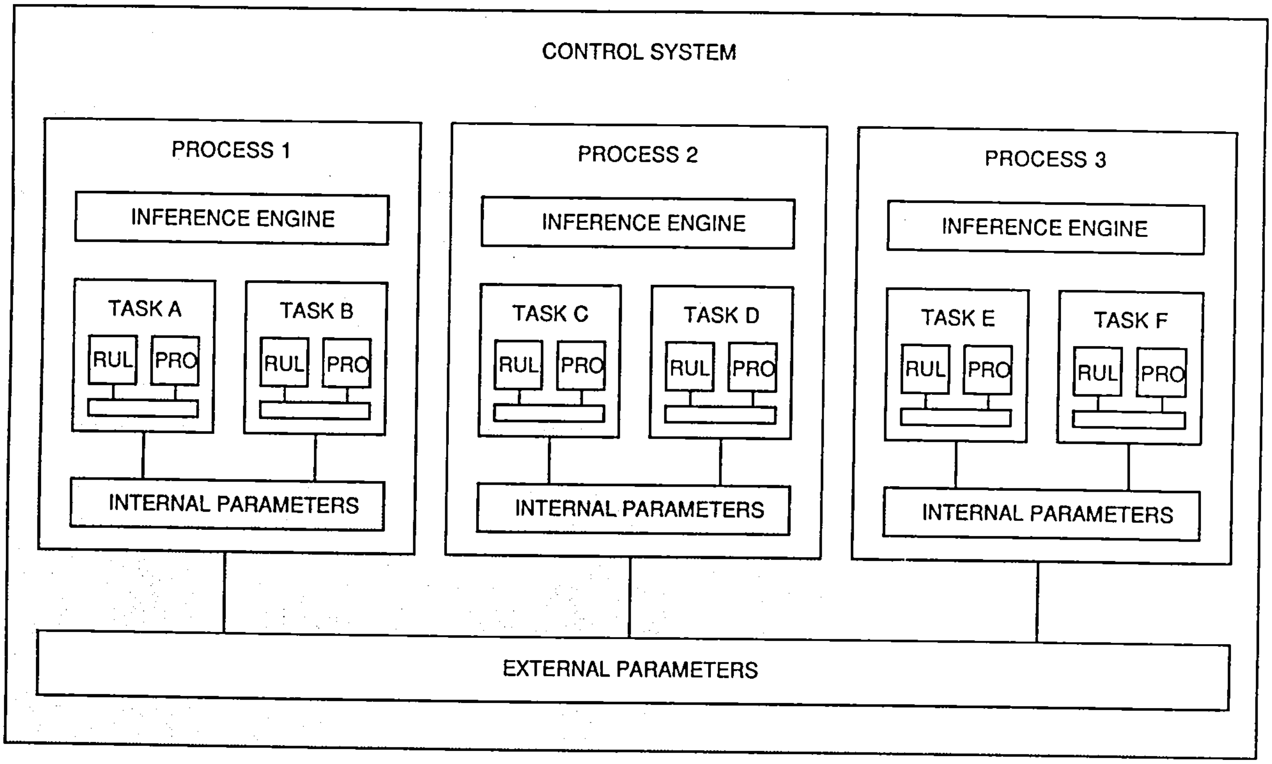
ORGANIZATION OF THE FAULT-TOLERANT FLIGHT CONTROL SYSTEM

The fault-tolerant flight controller breaks the overall job of failure accommodation into five main tasks. The executive control task provides continual dynamic state estimation, feedback control calculations, and synchronization of the remaining tasks. The failure detection task monitors aircraft behavior and detects significant abnormalities. The failure diagnosis task finds a set of probable causes and effects of the problem, while the failure model estimation task generates a mathematical model of the aircraft dynamics considered to reflect changes due to the failure. Finally, the reconfiguration task determines what action should be taken to correct the situation.



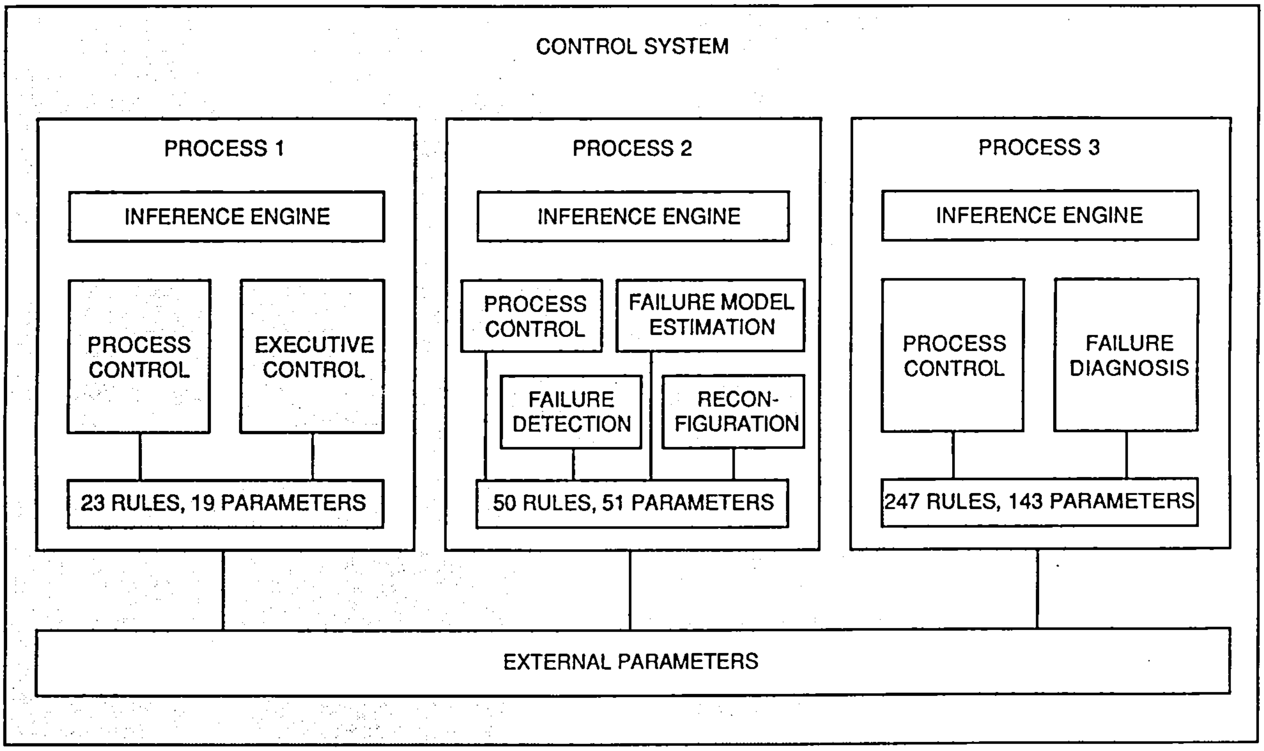
RULE-BASED CONTROL SYSTEM SOFTWARE ARCHITECTURE

The control system uses a rule-based search mechanism to perform control task scheduling and selection. Controller software development begins with the creation of a knowledge base for each task. Containing parameters, rules, and procedures, these knowledge bases perform intended task actions when properly searched. Tasks are grouped into processes, and each process knowledge base is translated automatically from LISP to Pascal. This code optimization, coupled with parallel processing, enables eventual real-time performance. The reduction of the overall control problem into tasks, followed by the grouping of tasks into processes, results in an organized hierarchical software structure resembling a set of cooperating expert systems using blackboard communications.



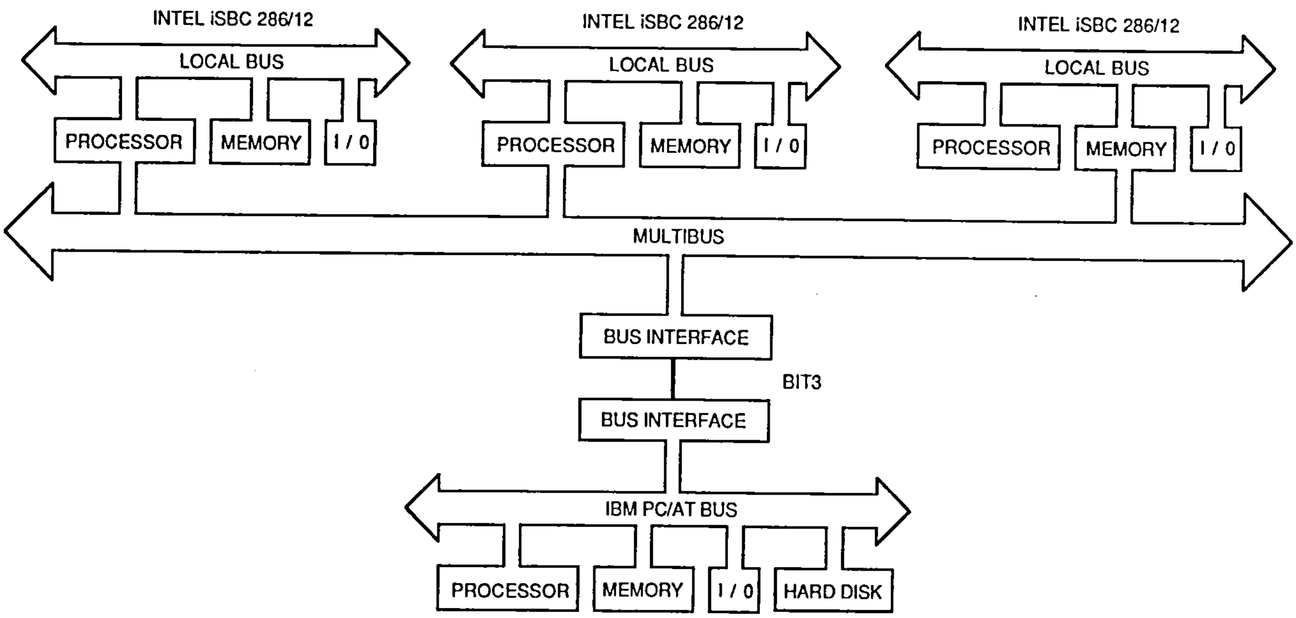
RULE-BASED FAULT-TOLERANT FLIGHT CONTROL SYSTEM SOFTWARE ARCHITECTURE

The fault-tolerant flight control system, presently capable of accommodating a significant bias or stuck failure in a sensor or control, implements the rule-based software architecture using 8 major tasks distributed among 3 processes. Each process contains a process control task responsible for intra-process task coordination and initialization. Process 1 contains the executive control task and a total of 19 parameters and 23 rules. Process 2 performs failure detection, failure model estimation, and reconfiguration using 51 parameters and 50 rules. Within process 3, 143 parameters and 247 rules perform failure diagnosis.



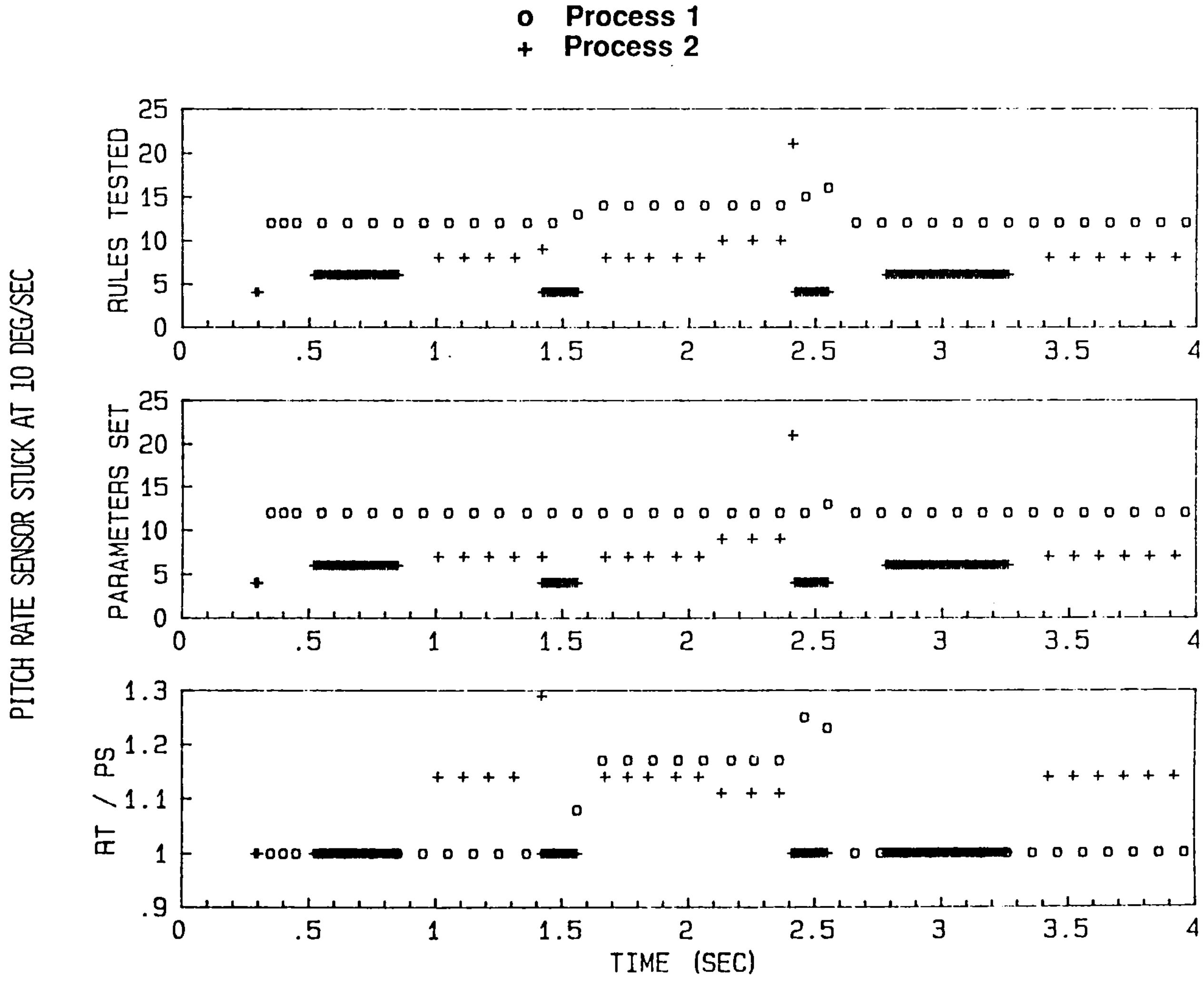
RULE-BASED CONTROL SYSTEM HARDWARE ARCHITECTURE

The controller is composed of single board computers embedded in a tightly coupled multi-microprocessor system. Each process is assigned to a processor, with inter-processor communications occurring through shared memory. During development, controller software is downloaded from a personal computer linked to the controller hardware through a memory-mapped bus interface.



RULE-BASED FAULT-TOLERANT FLIGHT CONTROL SYSTEM
SEARCH EFFORT TIME HISTORIES

The rule-based control system uses search to schedule and select actions. During a search, rules are tested in the attempt to infer parameter values. The amount of effort expended by a control system processor during a search can be measured in an absolute sense by the total number of rules tested, and in a relative sense by the ratio of rules tested to parameters set. Performance measures such as search effort can be used to verify adequate balancing of work load between processors.



CONCLUSIONS

Based on experience gained through the design and implementation of a fault-tolerant flight control system, the proposed rule-based approach appears applicable to a large class of complex control problems. Its beneficial characteristics include a hierarchical system organization, high-level rule description, a search-based decision-making mechanism, smooth integration of numeric and symbolic computation, an incremental growth capability, inherent parallelism and automatic code optimization for real-time execution speed, simplified coordination of processor communications, simplified modification of code, and powerful debugging facilities. Additionally, the rule-based technique can deliver real-time performance using conventional, economical hardware.

- **RULE-BASED APPROACH FACILITATES COMPLEX CONTROL SYSTEM DESIGN**

- **REAL-TIME PERFORMANCE ACHIEVABLE USING CONVENTIONAL HARDWARE**