INTELLIGENT FAULT-TOLERANT CONTROLLERS

Chien Y. Huang
Princeton University
Princeton, NJ

OVERVIEW

- ROLE OF (ARTIFICIAL) INTELLIGENCE IN FAULT-TOLERANT CONTROLS
- DESCRIPTION OF THE SYSTEM BEING DESIGNED
- AN EXAMPLE OF NEW CONTROL RECONFIGURATION APPROACHES
- SUMMARY AND FUTURE WORK

A system with fault-tolerant controls is one that can detect, isolate, and estimate failures and perform necessary control reconfiguration based on this new information. Artificial intelligence (AI) is concerned with semantic processing, and it has evolved to include the topics of expert systems and machine learning. Our research represents an attempt to apply AI to fault-tolerant controls, hence,the name intelligent fault-tolerant control (IFTC).

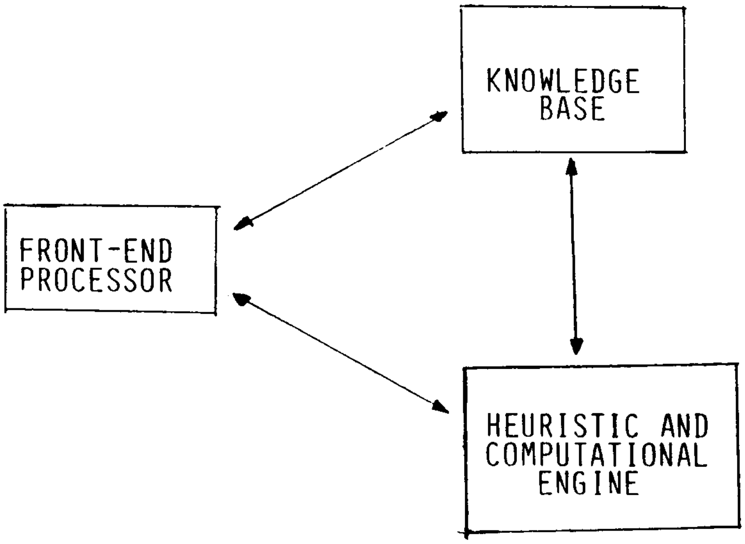
We are seeking generic solutions to problems, providing a system based on logic in addition to analytical tools, and offering machine learning capabilities. The advantages, to name a few, are that redundant system-specific algorithms are no longer needed, that "reasonableness" is used to quickly choose the correct control strategy (among many available ones), and that the system can adapt to new situations by learning about its effects on system dynamics.

INTELLIGENT FAULT-TOLERANT CONTROLS

- FAULT-TOLERANT CONTROL
  - FAULT DETECTION AND ISOLATION
  - FAULT ESTIMATION
  - CONTROL RECONFIGURATION
- ARTIFICIAL INTELLIGENCE (AI)
  - SEMANTIC PROCESSING
  - EXPERT SYSTEM (PROBLEM SOLVERS)
  - MACHINE LEARNING
- INTELLIGENT FAULT-TOLERANT CONTROLS (IFTC)
  - HYBRID OF ANALYTICAL METHODOLOGY AND AI
  - DIFFERENCES TO CONVENTIONAL APPROACH
  - ADVANTAGES

The Intelligent Fault-Tolerant Control (IFTC) System, which is currently being designed at the Laboratory of Control and Automation in Princeton University, consists of three main subsystems: front-end processor, which interfaces the system with the users; knowledge base, which houses the data base IFTC needs; and a heuristic and computational engine, which has both logic and numeric processing capabilities.

BLOCK DIAGRAM OF IFTC SYSTEM



The front-end processor is responsible for providing a friendly interface between the system and aircraft crew and designers of control strategies and heuristic rules. Through it the knowledge base can be updated or augmented. It will also guide a designer (usually an expert in some aspect) through the process of creating new rules and control strategies.

FRONT-END PROCESSOR

- INTERFACES THE SYSTEM WITH AIRCRAFT CREW/DESIGNER
- UPDATES AND AUGMENTS THE KNOWLEDGE BASE
- GUIDES USER THROUGH THE PROCESS OF CREATING NEW RULES AND CONTROL STRATEGIES

The knowledge base stores the knowledge that the system has about itself and about the "WORLD" (i.e., the flight environment). It contains dynamic models of the aircraft in which the IFTC system is resident; control laws for different flight conditions (high, low Mach number, wind shear, etc.), the rules under which they apply, algorithms for failure detection, isolation and estimation, and an interconnection map (and status) of all devices present on the aircraft.

KNOWLEDGE BASE

STORAGE FOR

- DYNAMIC MODELS OF AIRCRAFT
- CONTROL LAWS FOR DIFFERENT FLIGHT CONDITIONS AS WELL AS RULES UNDER WHICH THEY APPLY
- FAULT DETECTION, ISOLATION, AND ESTIMATION ALGORITHMS
- INTERCONNECTION MAP FOR ONBOARD DEVICES

There are two distinct processors in the Heuristic and Computational Engine subsystem. The heuristic processor possesses the logic needed to prune branches in free-structure rule-space. It also has the power to make decisions when logic does not apply. The computational processor is endowed with sufficient numerical capability to find, for example, new gains for feedback and observer equations and to evaluate the effectiveness of a new control law by simulation.

HEURISTIC AND COMPUTATIONAL ENGINE

- HEURISTIC PROCESSOR
  - PRUNES BRANCHES IN TREE-STRUCTURE RULE-SPACE
  - PERFORMS HEURISTIC DECISIONS
- COMPUTATIONAL PROCESSOR
  - FINDS NEW CONTROL GAINS
  - EVALUATES EFFECTIVENESS OF THE CONTROL LAWS

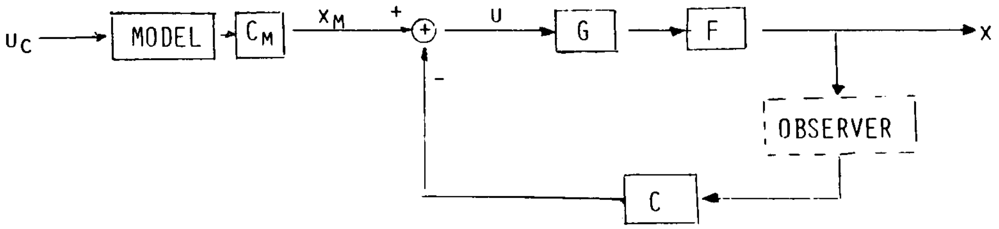
The lateral dynamics of an F-8C aircraft provides an example. The objective is to control its lateral motions by model-following. Suppose that the system is warned about an aileron-failure after a maneuver (it became biased at 1°). The question is what to do now.

"INTELLIGENT" CONTROL RECONFIGURATION - AN EXAMPLE

- MODEL: F-8C LATERAL DYNAMICS

$$x = Fx + Gu + w, \quad z = x, \quad u = -Cx + C_M x_M$$

$$x = (p \ R \ \beta \ \phi \ \omega \ \delta_A \ \delta_R \ \delta_{AC} \ \delta_{RC}), \quad u = (\delta_{AC} \ \delta_{RC})$$



- FAILURE: AILERON BIASED AT 1° DEGREE
  (FAULT DETECTION AND ESTIMATION ASSUMED)

A possible solution in conventional approaches is to set aside a value slot in software to represent the bias and add it to aileron commands. This slot normally holds a zero until a bias is detected at which time the slot is set to the estimated bias magnitude. Disadvantages of this method are that extra computation is necessary even when no bias is present and one such slot must be provided for every control on the airplane.

Faced with the same problem, the IFTC will instead change the dynamic model to reflect the new knowledge it acquired. This solution differs from the previous one in that the codes which control the aileron are being changed in real time. Therefore there is no need for reserved slots and no extra computation is carried out when there is no failure. Futhermore, the command is generic so that it will handle all cases with biases. We call this approach incremental learning because the IFTC uses its own tools to modify its "world" knowledge without outside intervention. It must also be pointed out that this example actually illustrates a restructurable control (as opposed to reconfigurable control) in the sense that the structure of the system has been changed.

Other failures that will be addressed in the future are losses of actuator or sensor. We will also consider critical flight conditions like wind shear penetration, where we will provide IFTC with knowledge to carry out such a task. In all these cases, we see clearly the need for intelligent and generic controls.

CONTROL RECONFIGURATION

- CONVENTIONAL FIX: ADD BIAS TO COMMAND IN A RESERVED SLOT

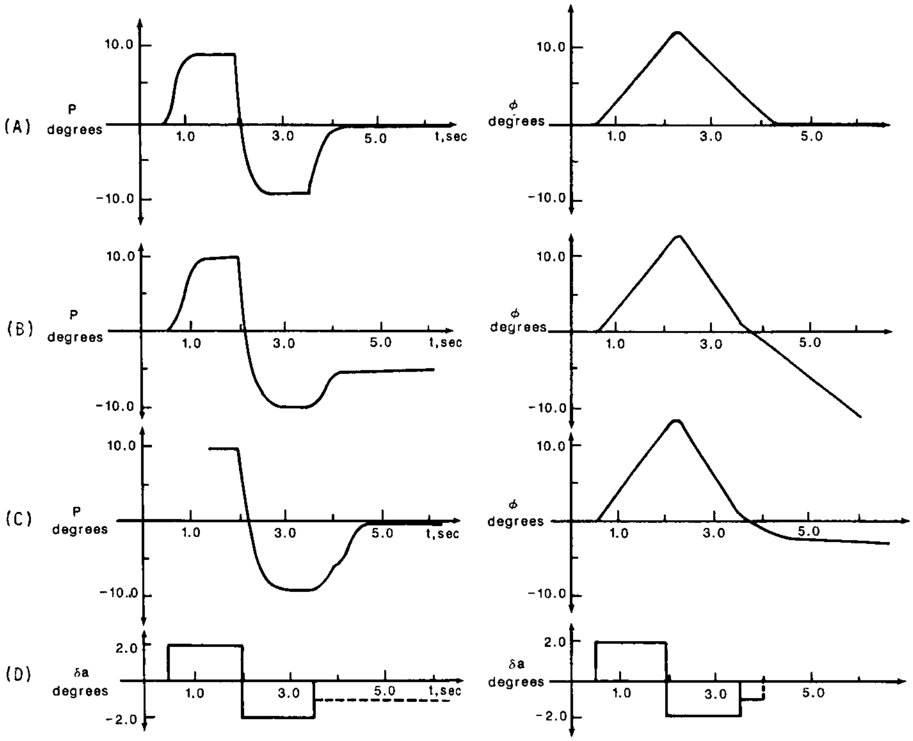
- NEW APPROACH: CHANGE DYNAMIC MODEL (USING A GENERIC COMMAND) TO REFLECT THE NEW KNOWLEDGE (INCREMENTAL LEARNING)

- OTHER "FAILURE" SITUATION: LOSS OF ACTUATOR OR SENSOR, WIND SHEAR

  - NEED OF INTELLIGENT GENERIC CONTROLS

A simulation was run with an aileron failure; (A) shows the normal roll response to a doublet aileron command, and (B) shows the failed case where at the end of the doublet command, the aileron became biased at 1°. (C) shows the control restructured a half second after the failure occurred; thus bringing the roll back to a desired state (at 0 deg/sec). (D) shows the aileron command, failure, and restructuring.

SIMULATION RUNS



SUMMARY

WE HAVE

- OUTLINED AN INTELLIGENT FAULT-TOLERANT CONTROLLER
- ILLUSTRATED THE NEW APPROACH BY AN EXAMPLE

FUTURE WORK

- EXPAND ON THE SUBSYSTEMS
- EXPERIMENT WITH ANALYTIC FAULT DETECTION ALGORITHMS
- STUDY THE MACHINE LEARNING