Experimental Test Beds for Embedded & Hybrid Systems

Gautam Biswas and Ken Frampton
Vanderbilt University/ISIS

NSF

UC Berkeley: Chess
Vanderbilt University: ISIS
University of Memphis: MSI

Foundations of Hybrid and Embedded Software Systems

Experimental Research

Goal: Train new generation of engineers for designing, analyzing, and developing complex, distributed heterogeneous systems

Synergy
Research ↔ Instructional Labs

At three levels:

1. **Researchers: Experimental Platform for Embedded and Hybrid System Design research, Tool Development, Applications**
2. **Graduate Education: Task driven research and experiments, Tool Development Research**
3. **Undergraduate Education: Lab experience for learning and understanding basic embedded and hybrid system characteristics, hands on learning by experimentation**

Experimental Platforms at ISIS

- Hardware test beds
  - Complex physical processes
  - Networks of embedded sensors
  - Multiple mobile assets
- Software platform for embedded system design and analysis
  - Model building tools
  - Heterogeneous models of computation
  - Analysis and verification tools (model transformations)
  - Real-time constraint-bound execution environments
  - Code generation tools
  - Application- and task- specific tools (e.g., fault-adaptive control, distributed control, layered fault management)

Test Bed 1: Control of Complex Embedded Systems

- ❑ Test bed: sufficiently complex physical process
- ❑ Modeling Environment: physical processes + controllers
- ❑ Design Tools
- ❑ Analysis tools: simulation, symbolic checking, verification
- ❑ Run time systems: code generation + experimental tasks

Physical Test Beds

- Three Tank System

- Mobile Robot systems

Six wheel all terrain truck

Pekee open platform

Soccer robot:USC/ISI

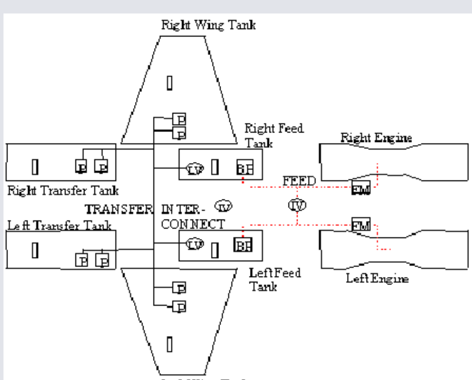
Software Platforms
(From Model-based Design research)

- Modeling languages for complex hybrid systems + embedded controllers
  - GME: visual modeling tool
  - Ptolemy: models of computation
- Analysis tools
  - Model Transformations
  - Reachability and safety analysis, symbolic model checkers for verification
- Real-time environments for embedded systems
  - QNX or VxWorks based
  - Giotto + e-machines
  - Real-time CORBA based environments
- Automated code generation
  - For various platforms

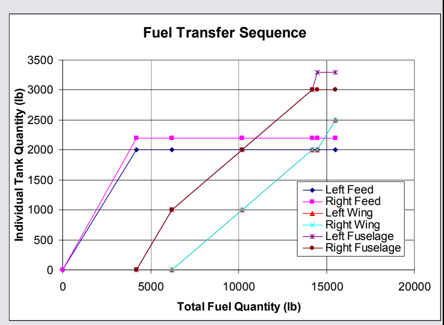
Example Task (1): Mobile Robot Control

- Robot must navigate and reach target while moving through unknown terrain
- Issues:
  - Terrain not uniform: multi-modal control; autonomous mode switching
  - Fault Detection and Isolation: process, sensor, and actuator faults
  - Fault Adaptive Control: fault identification, controller tuning, and reconfiguration

Example Task (2) Fault-Adaptive Control of Aircraft Fuel System



P Transfer Pump
LV Level Control Valve
IV Interconnect Valve
BP Boost Pump
FM Flow Meter
Fuel Quantity Sensor



Hybrid Control based on Tank Levels:
Supply sufficient fuel
Maintain aircraft CG

Possible Faults:

1. Degraded or Failed Wing and Fuselage Tank Pumps
2. Feed Tank and other Valve Degradations
3. Leaks in Pipes

Test Bed 2: Distributed Sensor Networks

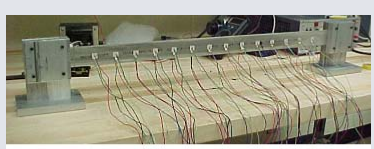
Task: Control of Distributed Parameter Systems

- Test bed: physical systems with distributed parameters – e.g. vibrations, acoustics, fluidics, environmental
- Critical issues at the interface of mechanical/computational systems:
  - Control system design in an embedded computational environment
  - Effects of embedded system limitations on control
  - Leveraging embedded software technologies for control

Experimental Platforms

Experimental test beds of increasing scalability and complexity will be developed

Simple beam vibration control



Acoustic target detection

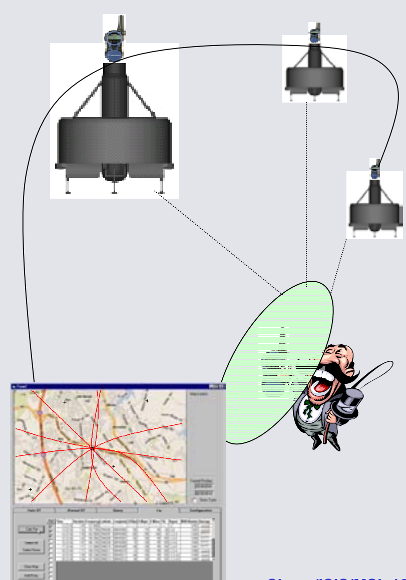
Complex structural acoustic control (launch vehicle payload fairing)

Test Bed 3: UAV-Based Radio Location

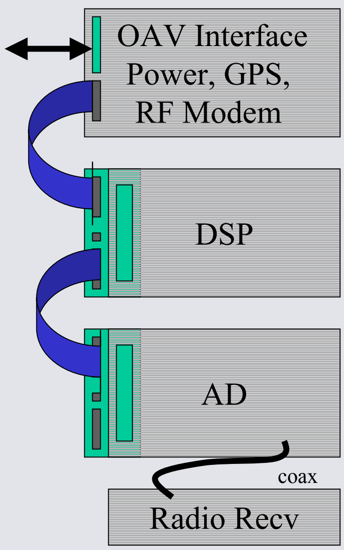
- Multiple Organic Air Vehicles (OAVs)
- Time Difference of Analysis (TDOA) for
- geo location of objects
- Tracking as object(s) of interest move
- Issues:
  - Model-based design and integration of OAV payload

Example Task: Find a radio source

- Family Radio Systems
  - Detect call button
  - Utilize GPS clock to find relative time of detection of call button
- DSP processing
  - A/D Sample baseband FRS data
  - Detect call signal
  - Communicate TOI (based on GPS clock – perhaps refined) via serial RF to base
- Base Calculation
  - intersection ellipse
  - location
  - coordination redirection



Hardware Platform: Form Factored Payload



From OAV:
- GPS Location
- Clock
- 5V Power

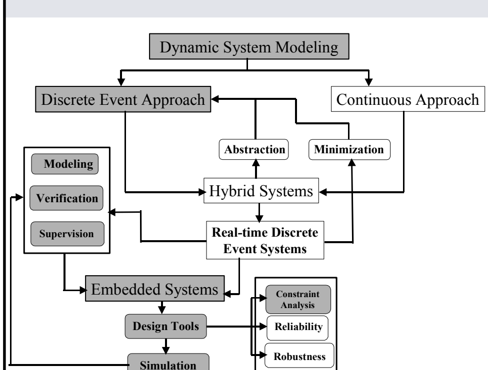
To OAV:
- Position Control
- Network: RF Modem

TMS320C611
- 500-900 MFLOPS
- 64 MB RAM
- 100K Gate FPGA

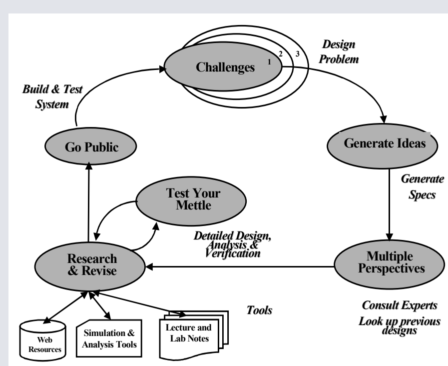
A/D Converter
- Dual Channel, 12 bit
- 50 MSample/Sec

Receiver
- First-Cut: FRS

Application to Undergraduate Education



The Curriculum



Challenge-Based Instruction