CSCI1600: Embedded and Real Time Software

Lecture 8: Modeling I: Continuous Systems
Steven Reiss, Fall 2017
Readings

- What did you get from the reading?
Why Modeling

- Specifying system behavior
  - Understanding what the system should do
    - In response to external events
    - In response to internal events
- Designing the system
  - Requirements, specifications
  - Understanding odd conditions
  - Understanding time and other constraints
Why Modeling

- Proving properties of the system
  - Proving a system is “safe”
    - What does safety mean
    - Show the system fits this meaning
    - Can be done in terms of models
    - Also can be done in terms of code
  - Performance properties
    - Timing requirements
Proving Program Properties

- Prove heat and cold never on within 5 minutes of each other
  - Can do this by looking at the code
    - How complex is the code, how difficult
    - What are all the things that can happen
    - What if there are multiple processes/threads
  - Easier to prove a high-level model satisfies the property
  - And then show the code meets the model
The Real World

- Embedded systems exist in context
  - That context is the real world
- Is the real world continuous or discrete?
Physical Systems

- If we embed into a physical system
  - What is the effect
  - What do we want to happen
  - Will we achieve the effect
  - What can go wrong

- Need to model the physical system
Modeling for Design

- Why bother with modeling
  - Why not just implement what we want
  - And see what happens
- What do you want
- How do you know what will work
- What happens if things go wrong
Consider a RC Car

- What are you concerned with
  - Position of the car (dimension?)
    - Location, wheel positions, velocity
  - Relative or absolute

- What are your controls
  - Accelerate or not
    - Binary or can you control
    - How does this affect the position
  - Turn left/right
    - Binary or can you control
    - How does this affect the position
RC Car

- What accelerate means
  - How much acceleration
  - Maximum speed
  - Slowdown when not accelerating

- What does turn mean
  - How fast does the wheel turn
  - Maximum amount of turn
  - Does car follow wheel?
RC Car

- Suppose you want to track a line
  - Can tell left/right by amount
- What should you do?
Modeling

- This is one reason to model
  - Understanding what your options might be and what your system should think about doing
  - Making it easy to measure the effects
- Are there others?
Simulation

- Suppose our RC car were breakable
  - E.g. helicopter, drone, ...
- Suppose it had real effects
  - E.g. real car, X-ray machine
  - Pin ball machine
- Suppose it were expensive
  - Or not yet built
- How would you create the embedded program
Simulation Environment

- Create a simulation environment
  - Simulate the physical world
  - Let your program run as part of the simulation
- This requires modeling the real world
  - Physical details
  - Gravity, motion, etc.
  - Can be chemical, heat, ...
- How many have written a simulation?
Simulation

- Small time step
  - Fixed or variable
- Track position, velocity, orientation, state
  - Of each object
  - Or subobject (e.g. wheels of RC car)
- Compute next state
  - Next position, velocity, orientation
- Consider a bouncing ball
Simulation

- Effectively doing calculus
  - Small step = delta
  - Smaller step => more accurate
  - This is why you develop the formulas as in the text

- Different ways of computing next
  - Runge-Kutta, ...
  - Different amount of current and next deltas
  - For our purposes, simplest should work
Time Step

- Larger time step is faster
- Smaller time step is more accurate
  - For simple motion, not that important
  - Complex motions, collisions
- Need a time step that ensure events
  - Pinball machine
  - Pool table
  - RC Car
Simulation

- Can be expressed in modeling language
  - Matlab (Simulink), LabView provide a graphical interface
  - Actor model described in text

- Various systems exist
  - Special-purpose systems (e.g. flight simulators)
Simulation Concerns

- **Fidelity**
  - How accurately does it model the real world
  - How important is this accuracy

- **Speed**
  - Does it operate in real time
  - Does it operate on the time scale of your system

- **Debugging**
  - Can you debug your program using it
  - Can you simulate real-world faults
Embedded Systems

- You can model the outside world
  - Within your program
  - But you probably don’t want to simulate it
  - WHY?

- What do you do instead
  - Get outside state, compare to expected
  - Compensate
Feedback

- In general, your system doesn’t try to compute the next step based purely on the model
- Instead it uses feedback to control the system
- We’ll get into this in two weeks
We’ve talked about modeling the outside world

Suppose we want to model our embedded system

- What’s different
- How might we do this
RT/Embedded SW Architecture

- Break the problem into tasks
- Each task has its own requirements
  - How often to run, when to run
  - How much time it takes
  - How critical it is
- What are the tasks for a problem?
Task Types

- Control-oriented tasks
  - Managing state, handling sequential actions

- Timer-oriented tasks
  - Something needs to be done periodically

- Data-oriented tasks
  - If multiple tasks access a data structure
  - Have a task in charge of that structure

- Device-oriented tasks
  - Single task to handle a device
Modeling the System

- First modeling the individual tasks
  - Interactions with the physical world
  - Interactions based on time
- Then modeling their interaction
  - With each other
Homework

- Read Chapter 3.1-3.5
- Exercises 3.2 and 3.6