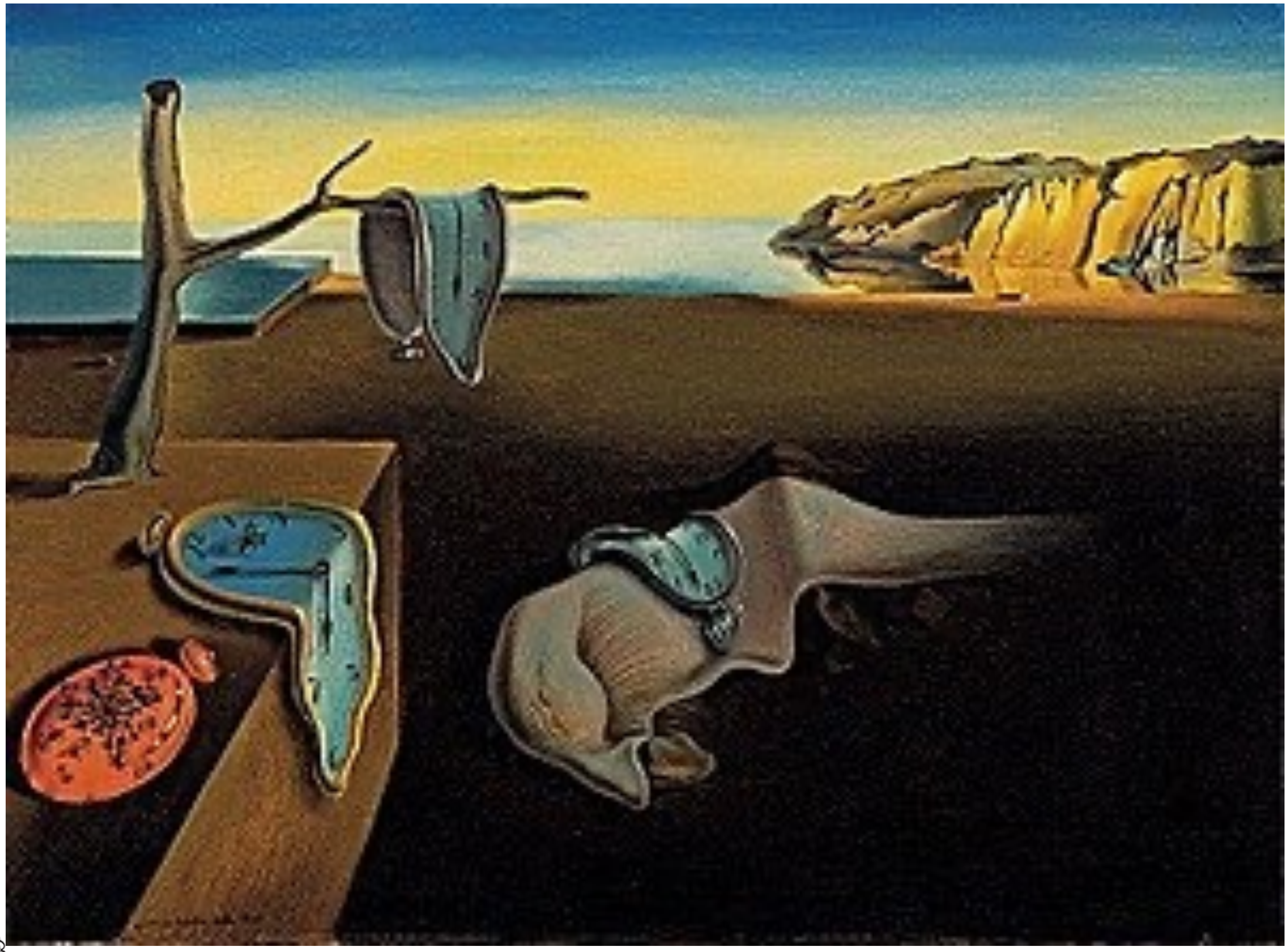
Embedded and Real Time Systems Theory Overview

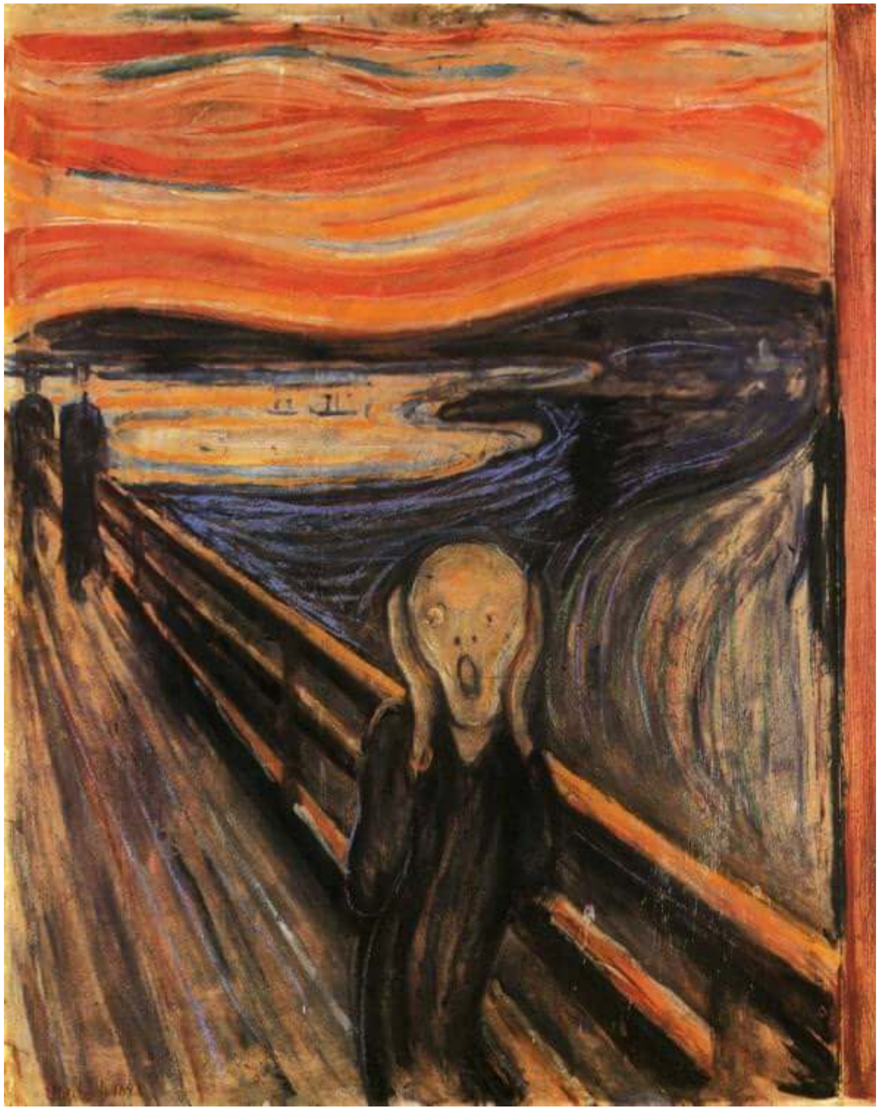
David Andrews
dandrews@eecs.ukans.edu

Its time for the land of Theory !



Salvador Dali The Persistence of Memory

Are You Ready ?



Edvard Munch The Scream

What We Will Cover Today

- Theoretical Basis
  - Signal Processing
  - Control Theory
  - Timekeeping
- Modeling/Requirements/System Architecture

Sci/Eng Disciplines Grounded in Theories

- Circuit Theory Very Satisfying
  - Physical Phenomena expressed by mathematical equations
    - V = IR
    - $I = C\,\frac{dv}{dt}$
    - $V = L\,\frac{di}{dt}$
  - Voltages, Current in a circuit solved by linear equations
    - A computer can do this !

Real Time Systems Theory

- Real Time Systems adopt theories to describe some physical attributes from:
  - Signal Processing
    - Sampling
    - Analog/Discrete Signal Processing
  - Control
    - Inputs, outputs, transfer functions, control functions
  - HOWEVER……

R.T. Sys Theory Shortcomings

- No Fundamental Laws for Design/Analysis of Real Time Systems…..why ?
  - No fundamental theory of embedded software
    - Programming languages have no inherent inclusion of time
    - Programming Models
  - No fundamental theory of embedded hardware
    - Based on Turing Machine
      - Says we should guarantee a stopping state……
  - Mismatch between computational and programming models
    - We represent application in a programming language without the concept of time, and implement it on a platform on a model that provably halts…….

R.T. Theory

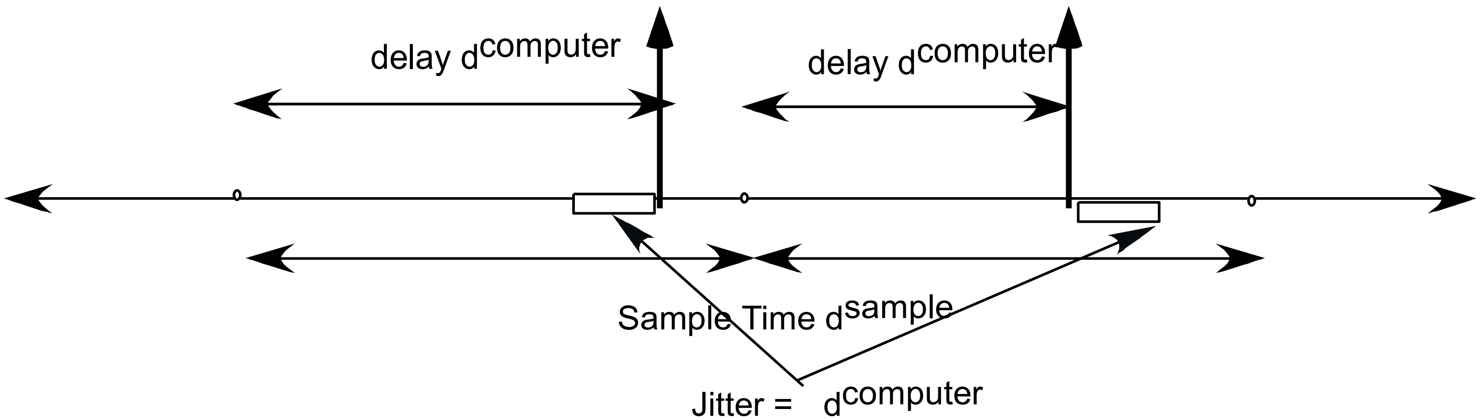
- Be that as it may……
  - Use the theories that we have
    - Signal Processing
    - Control
    - Discrete Time Keeping
  - Then adopt some key models
    - Concurrency
    - Finite State Machines
- Mix and shake to form an approach to understand how to relate a set of requirements into a computable solution…..

Start with Some Temporal Requirements

- Were do temporal requirements come from ?
  1. The controlled object
  2. The controlling computer system (aka embedded system itself)…

- Controlled Object (From Control Theory)
  - delays associated with the time the system requires to
    - Response delay to initiated change d^{object}
    - Achieve desired change d^{rise}
  - Other Variations/Types Depending On Actual System
- Controlling Computer
  - Delays associated with
    - Sampling Times d^{sample}
    - Calculation Times: computer delay $d^{computer}$
    - Variance in Calculation Times (Jitter) $\Delta d^{computer}$

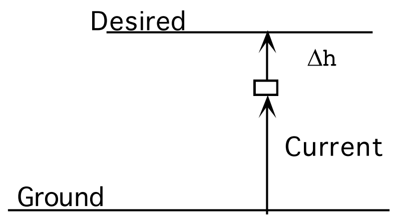
Computer Controller Delays

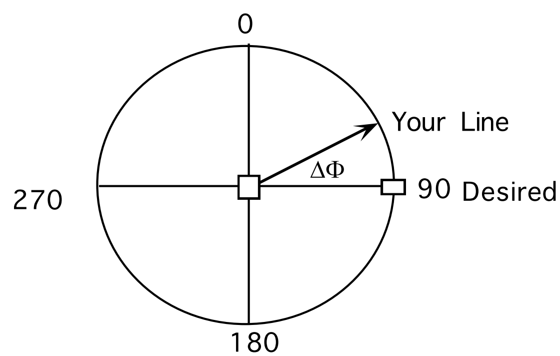


Controlled Object

- Much information borrowed from control theory….
- Exact information will be based on system under consideration.

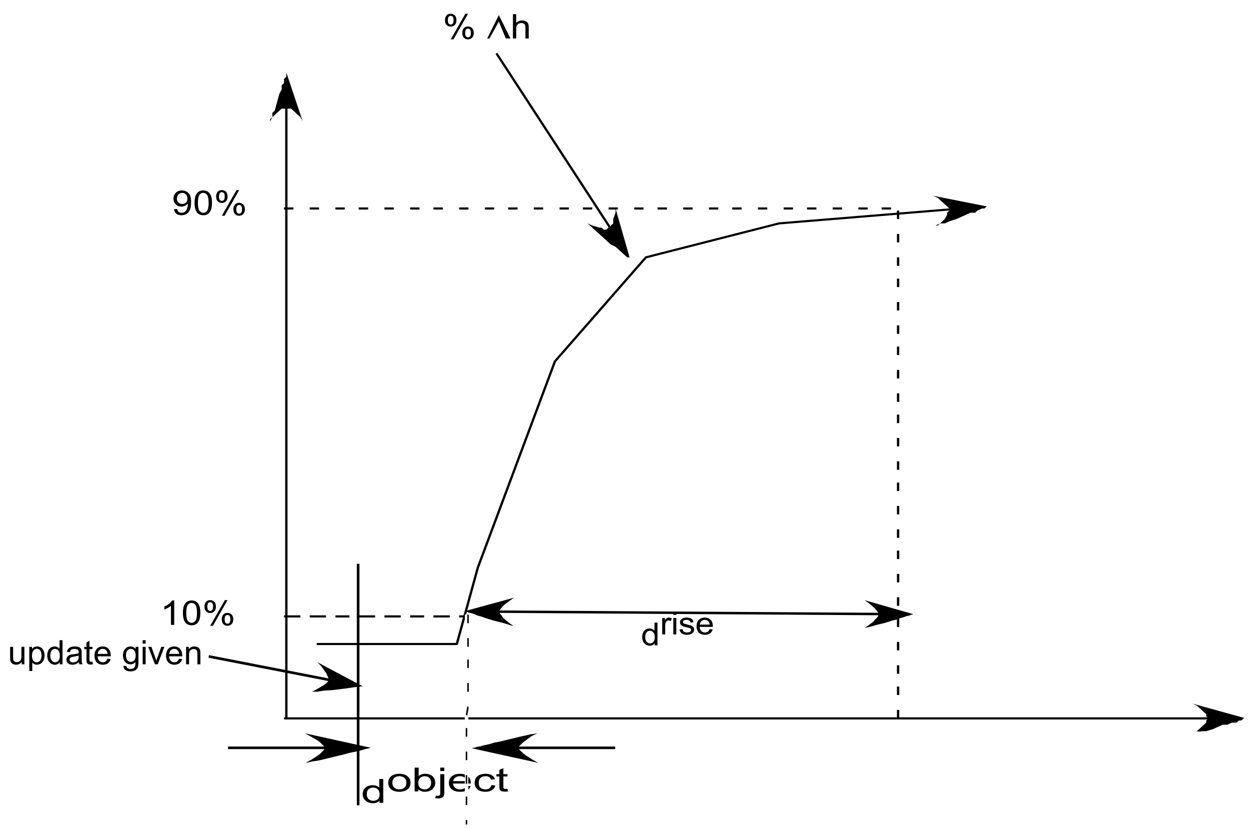
Example: Flight controller that will adjust Azimuth and Elevation





System Timings

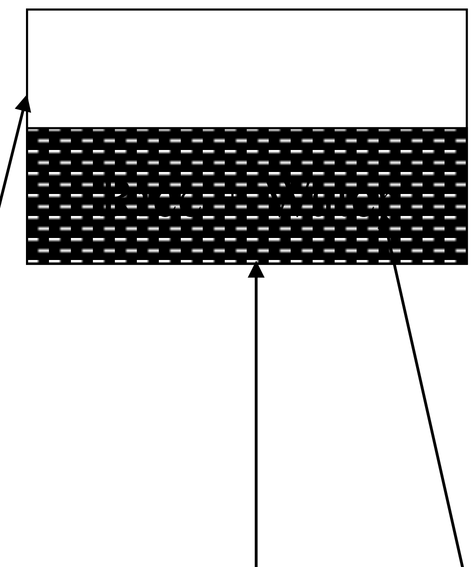
- Δh is given to actuators for adjustment



Summary of Control Loop Parameters

Symbol	Parameter	Sphere of Control	Relationships
d^{object}	Controlled object delay	Controlled object	Physical process
d^{rise}	Rise time of step response	Controlled object	Physical process
d^{sample}	Sampling period	Computer	$d^{sample} << d^{rise}$
$d^{computer}$	Computer delay	Computer	$d^{computer} << d^{sample}$
$\Delta d^{computer}$	Jitter of delay	Computer	$\Delta d^{computer} << d^{computer}$
$d^{deadtime}$	Dead time	Computer and controlled object	$d^{computer} + d^{object}$

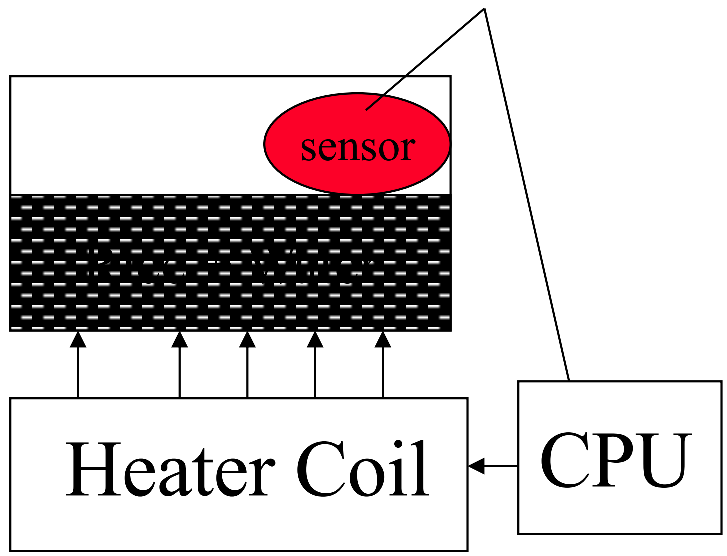
Example Simple Control System



In example:

Object = metal pot + water + rice

Example Simple Control System

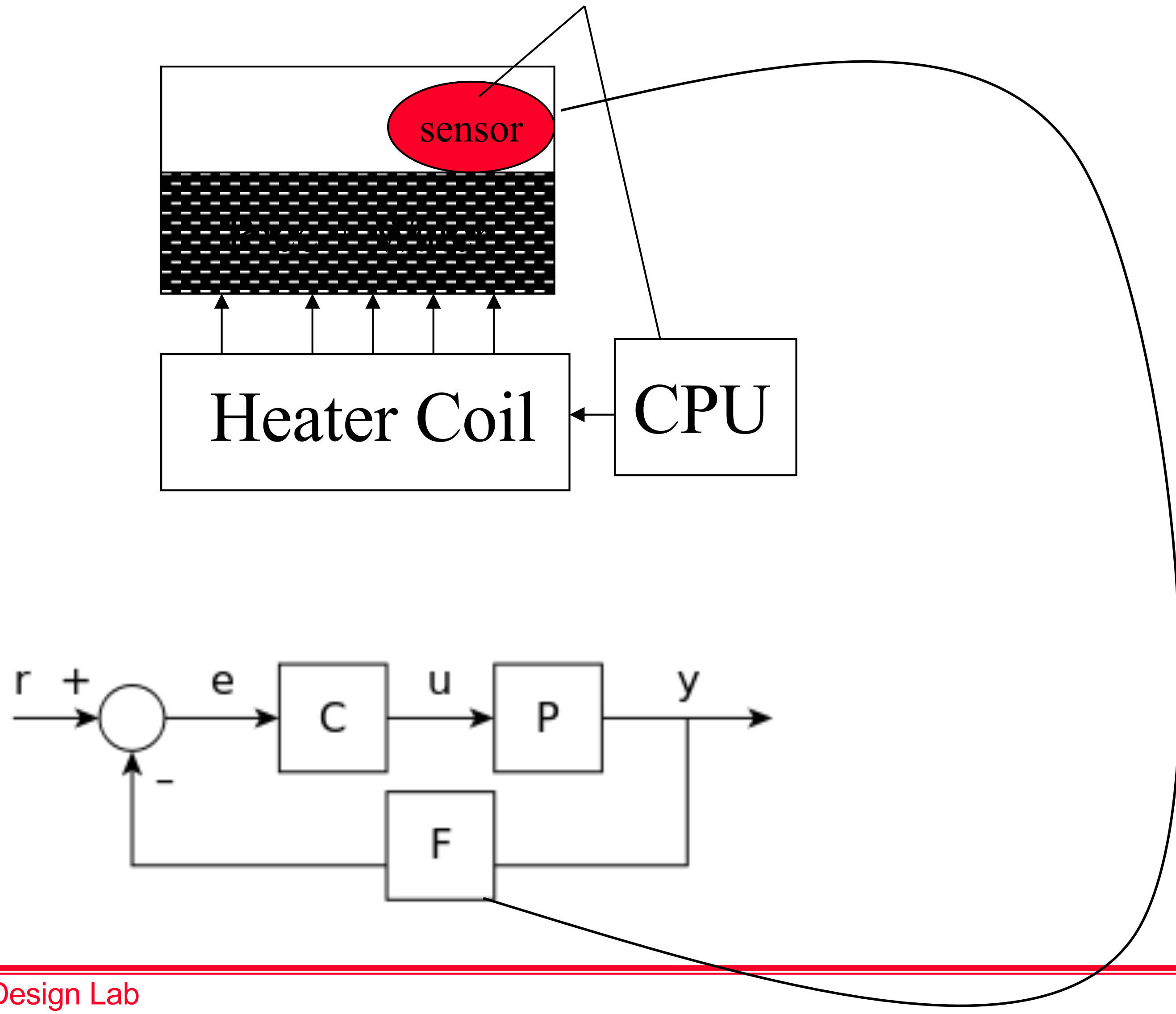


In example:

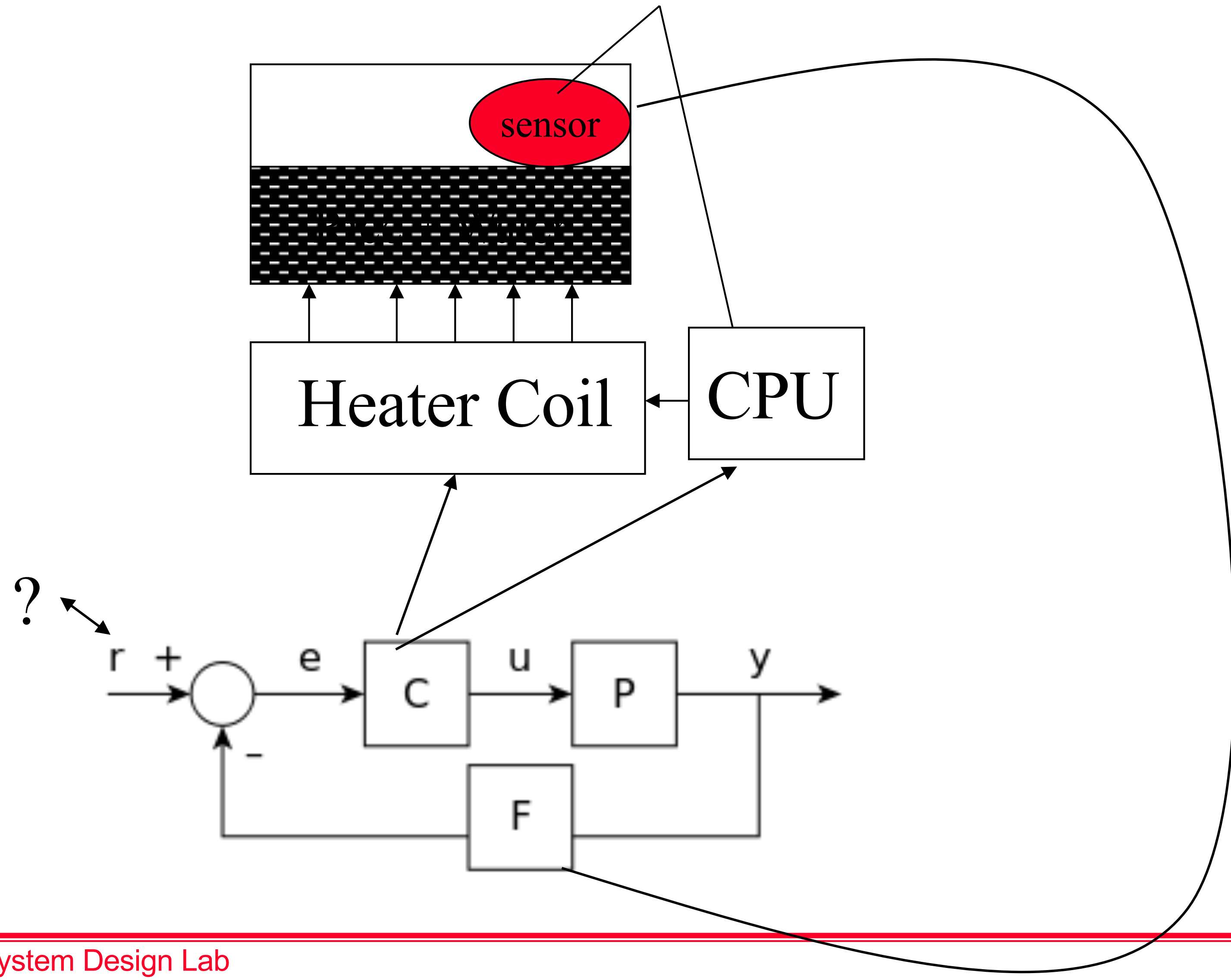
Object = metal pot + water + rice

Control system is temperature sensor + heating elements

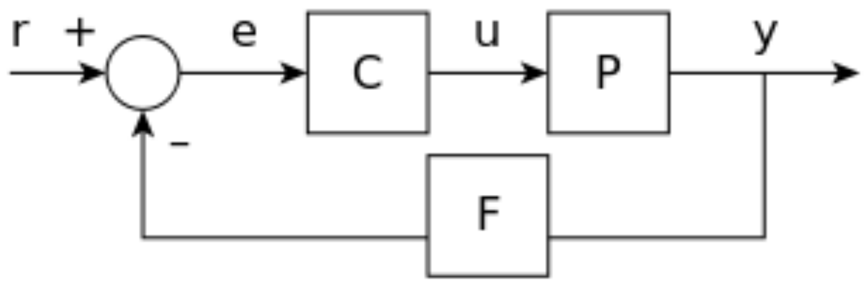
Model as closed loop control system



Model as closed loop control system



Model as a closed loop control system



- Classic Closed Loop Transfer Function

$$H(s) = \frac{P(s)C(s)}{1 + F(s)P(s)C(s)}$$

Element Definitions

d^{object} controlled object delay

- Delay from applying control force to first observed response
- Due to inertial lag of physical plant (speed of thermal wavefront in rice cooker)

d^{rise} rise time of step response

- Physical time constant of system (thermal mass of rice+ water+ pot+ heaters)

d^{sample} sampling period

- How often temperature sensor is read (should be **> 10x rise time**)

$d^{computer}$ computer delay

- Time to compute new actuator command point (sensor-heater on or off)

$\Delta d^{computer}$ jitter of computer delay

- Variations in computer delay (*e. g.* , cache misses, competing tasks)

$d^{deadtime}$ dead time

- End- to- end latency from observation to action (lower = more stable)

Example Values For Rice Cooker

d^{object} = 90 seconds
- Time from electricity to coil to temperature change at sensor
- Varies depending on coil size (amps) and metal

d^{rise} = 10 minutes
- Time to boil water; varies with amount of water

d^{sample} = 10 seconds to 1 minute sampling frequency
- No point sampling temperature at 10 MHz !

$d^{computer}$ = 10- 100 msec compute time
- Hard to buy a computer slower than that

$\Delta d^{computer}$ = .1 - 1 msec jitter
- Conditional branches in software
- A/ D timeout loops, early- out multiplication, etc.

$d^{deadtime}$ = 90 sec + 10 msec ~= 90 sec
- Computer isn't a limitation in this case

General Control System Issues

Latency is bad; it can create unstable systems

- If control loop is 180 degrees off from an oscillation, it will amplify problems
- Variability in latency reduces control effectiveness
- Communication network can be a large part of this latency
- And, you're usually stuck with a given latency in the physical system

Make sure control loops run faster than plant time constants

- Generally **10x faster** gives smooth control and a safety margin
- Generally, want to set control loop deadlines faster than control loop frequency
- (each answer computed before next reading is taken)

Computational Components

- Real Time System Interfaces With World in a Timely Fashion
- Look at a Simple Computational Node's Requirements
  - Computational Element (CPU, FPGA, etc)
    - Lets assume CPU for now. What decides this ?
  - Input/Output Capabilities
    - Standard Serial/Parallel Communications devices
  - Timer Chip
    - Resolution
  - Memory
    - Program and Data Storage