CS684 – Embedded Systems (Software)

Introduction to Realtime Systems - I

Kavi Arya

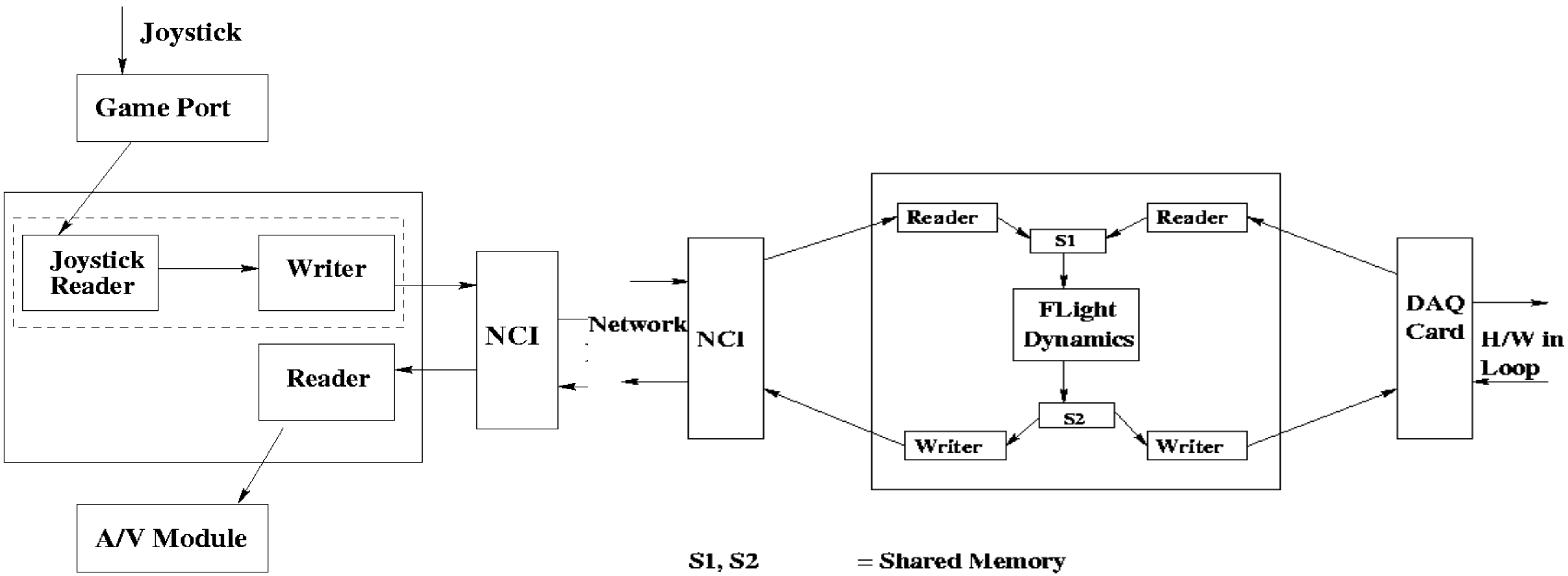
IIT Bombay

Embedded Systems?

Plan

- **Realtime Embedded Systems**
  - Introduction
  - Application Examples
- **Real-time support for ESW**

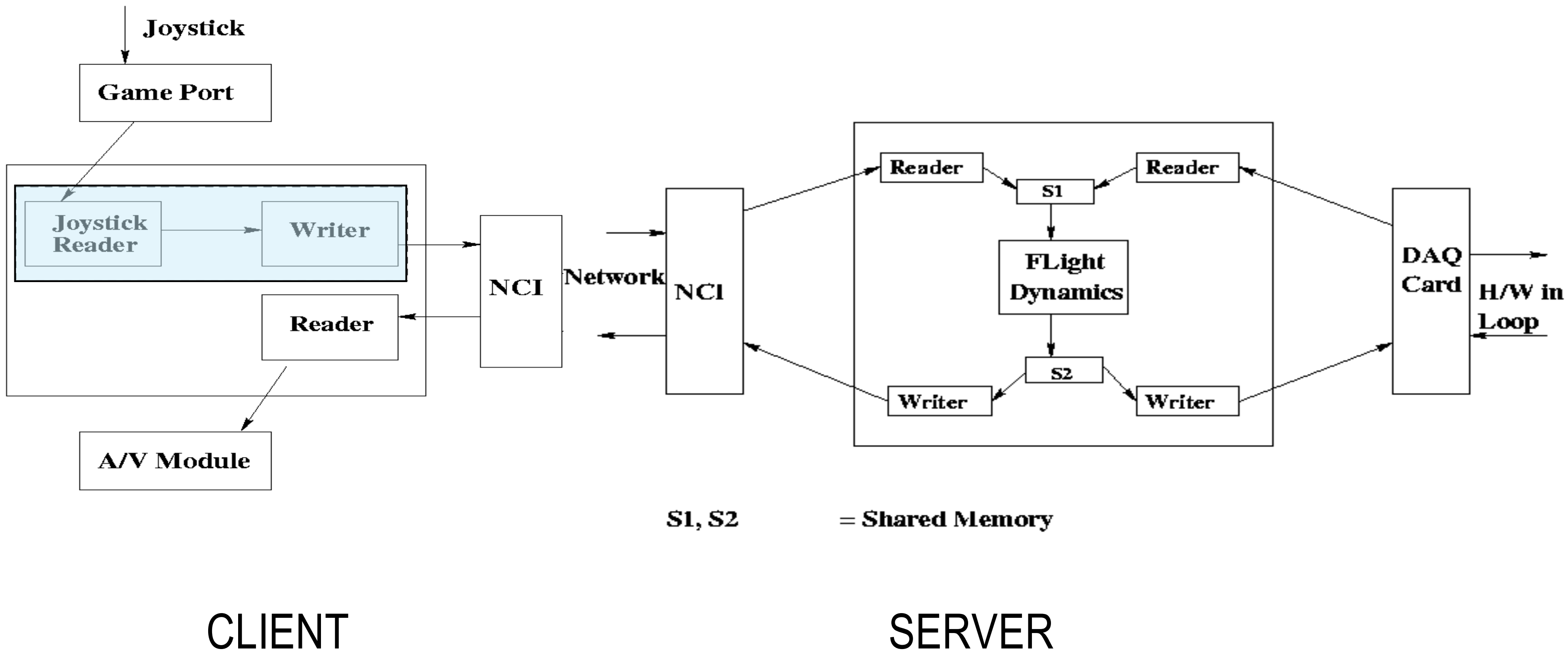
1. Flight Simulator



CLIENT - pilot SERVER - simulator

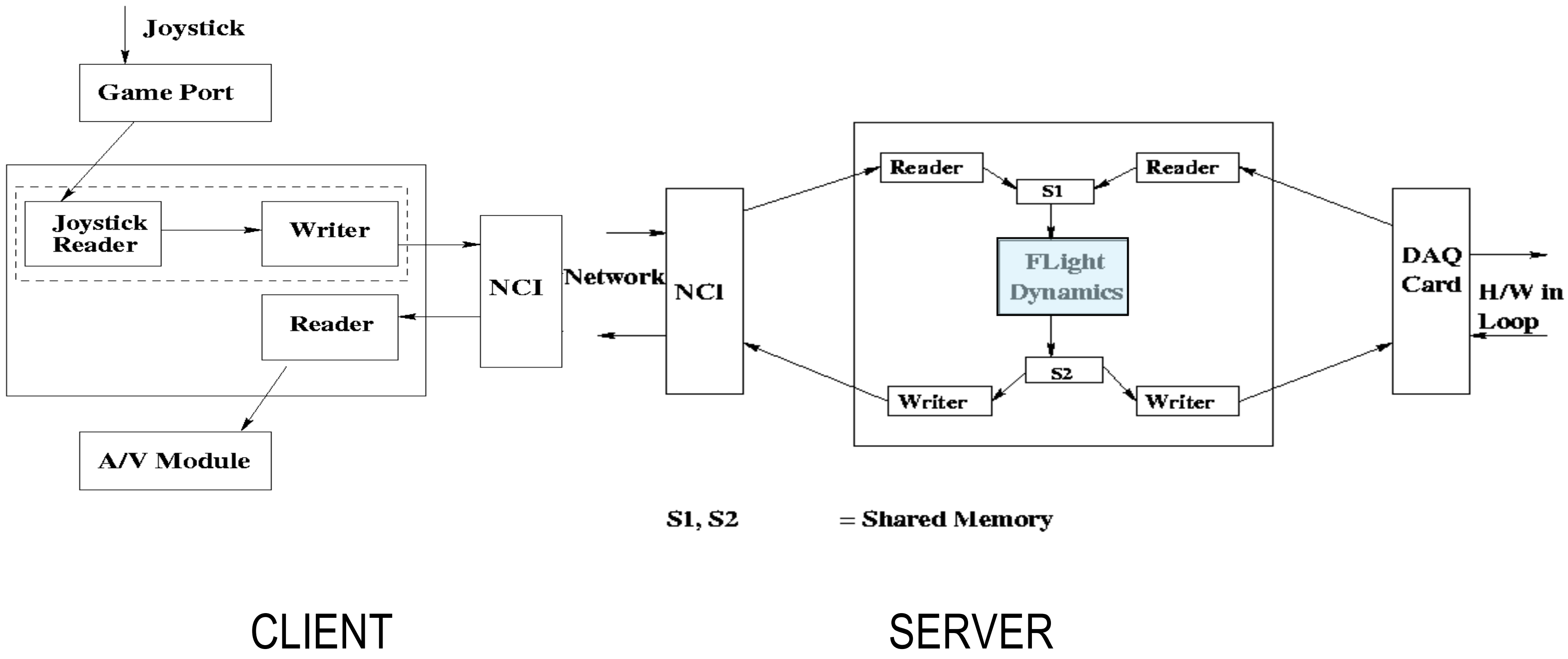
Constraints on responses to pilot inputs, aircraft state updates

Time Periods to meet Timing Requirements



Requirement	Choice Made	Rationale
Continuous pilot inputs should be polled at rates greater than **62.5** *Hz*	The time period of the writer on Client should be less than **16** *ms*	The writer thread on the Client polls for the pilot inputs from the joystick

Time Periods to meet Timing Requirements…



Requirement

The state of the aircraft is to be advanced at **12.5** *ms* time steps

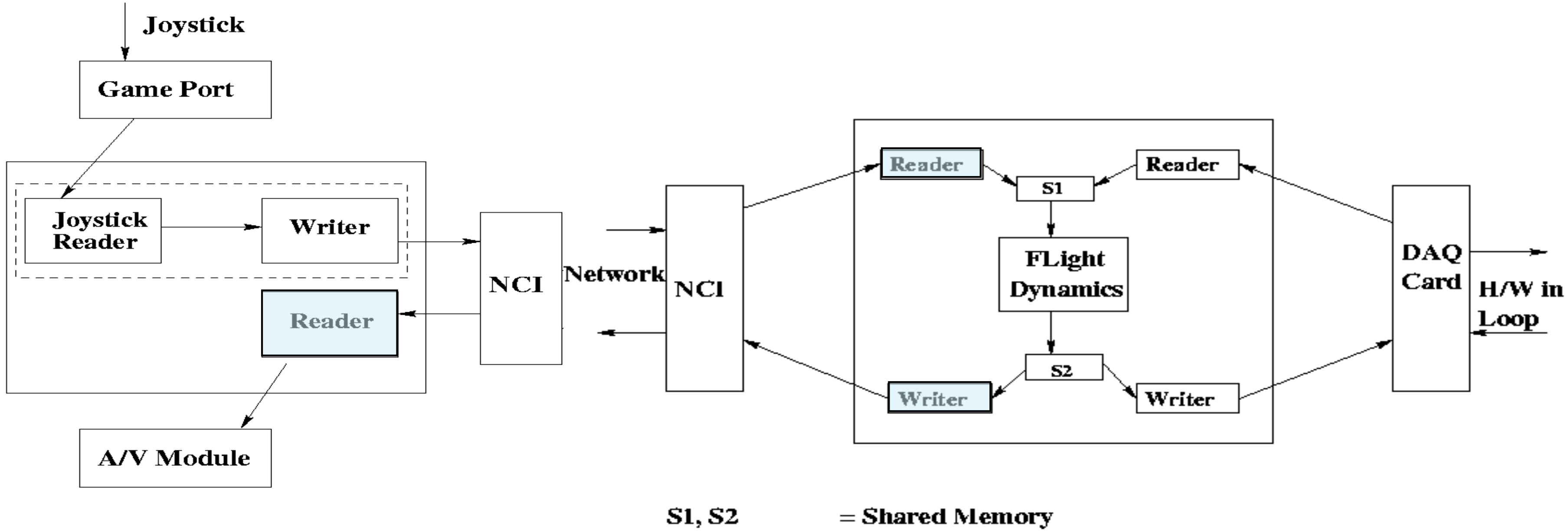
Choice Made

The time period of the Flight Dynamics thread on the Server is **12.5** *ms*

Rationale

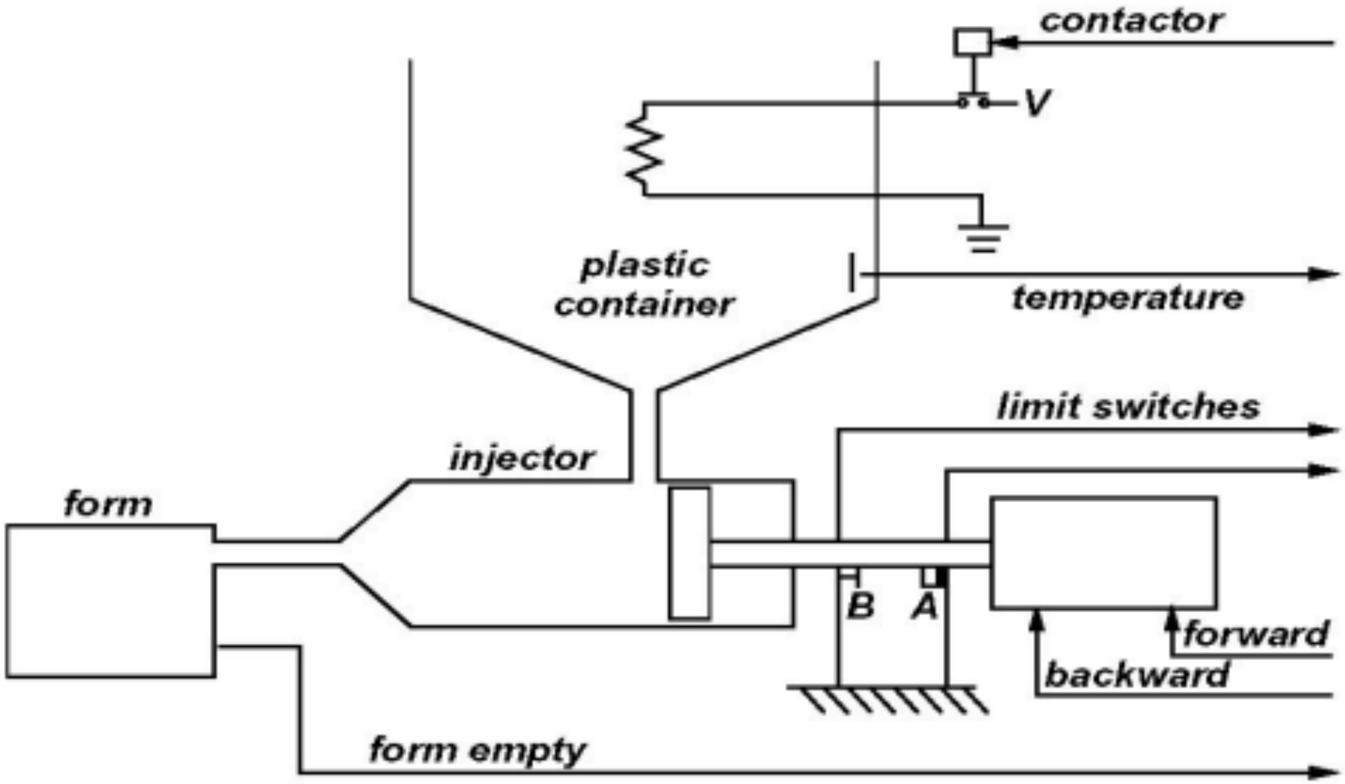
The flight dynamics thread on the Server advances the state of the system

Time Periods to meet Timing Requirements…



Requirement	**Choice Made**	**Rationale**
Response time for pilots should be less than **150** *ms* for commercial aircrafts and **100** *ms* for fighter aircrafts	Reader and Writer threads on Server, and the Reader thread on the Client should be as fast as the system permits. (Time period of **4** *ms* in our case)	• Delay in data transfer at these threads increases the response time • These threads should be interrupt driven in order to minimize the response time

Example 2: Injection Molding



–Keep plastic at proper temperature (liquid, not boiling)

–Control injector solenoid (make sure that the motion of the solenoid terminates before the piston reaches the end of its travel.

Source: "Laboratory for Perceptual Robotics, UMass" Copyright 1996 by Roderic A. Grupen

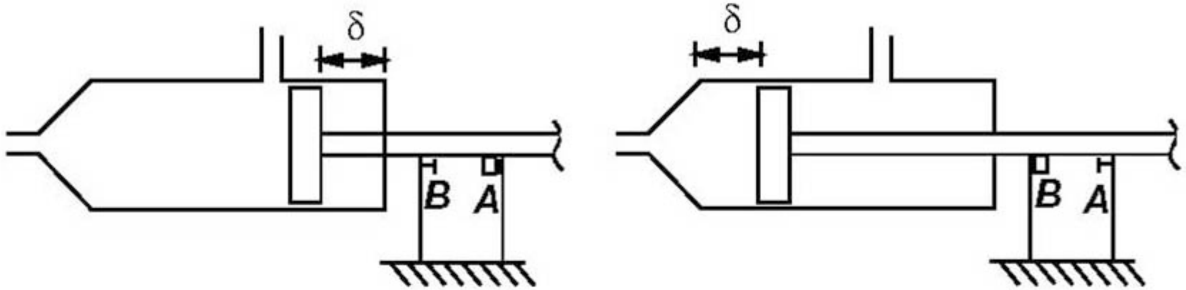
Controlling a reaction

- **we know:**
  - if temperature too high, it explodes
  - maximum rate of temperature increase
  - rate of cooling
- **events:**
  - temperature change
  - temperature > safe threshold
- **we can derive:**
  - how often we have to check temperature
  - when we have to finish cooling

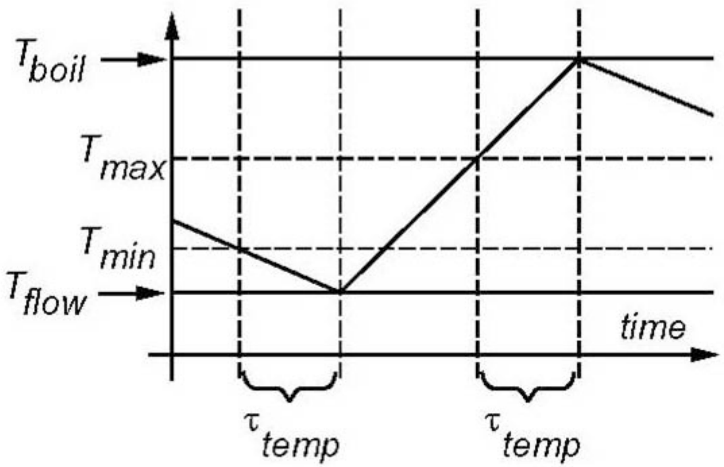
Example - Injection Molding (cont.)

– Timing constraints

- the injector must be off within τ_{inj} seconds after A or B limit signals, so that $v_{inj}\tau_{inj} < \delta$



- the temperature control contactor must be activated within τ_{temp} seconds of a temperature event, so that $T_{boil} < T < T_{flow}$



Example - Injection Molding (cont.)

– Concurrent control tasks

injector control:

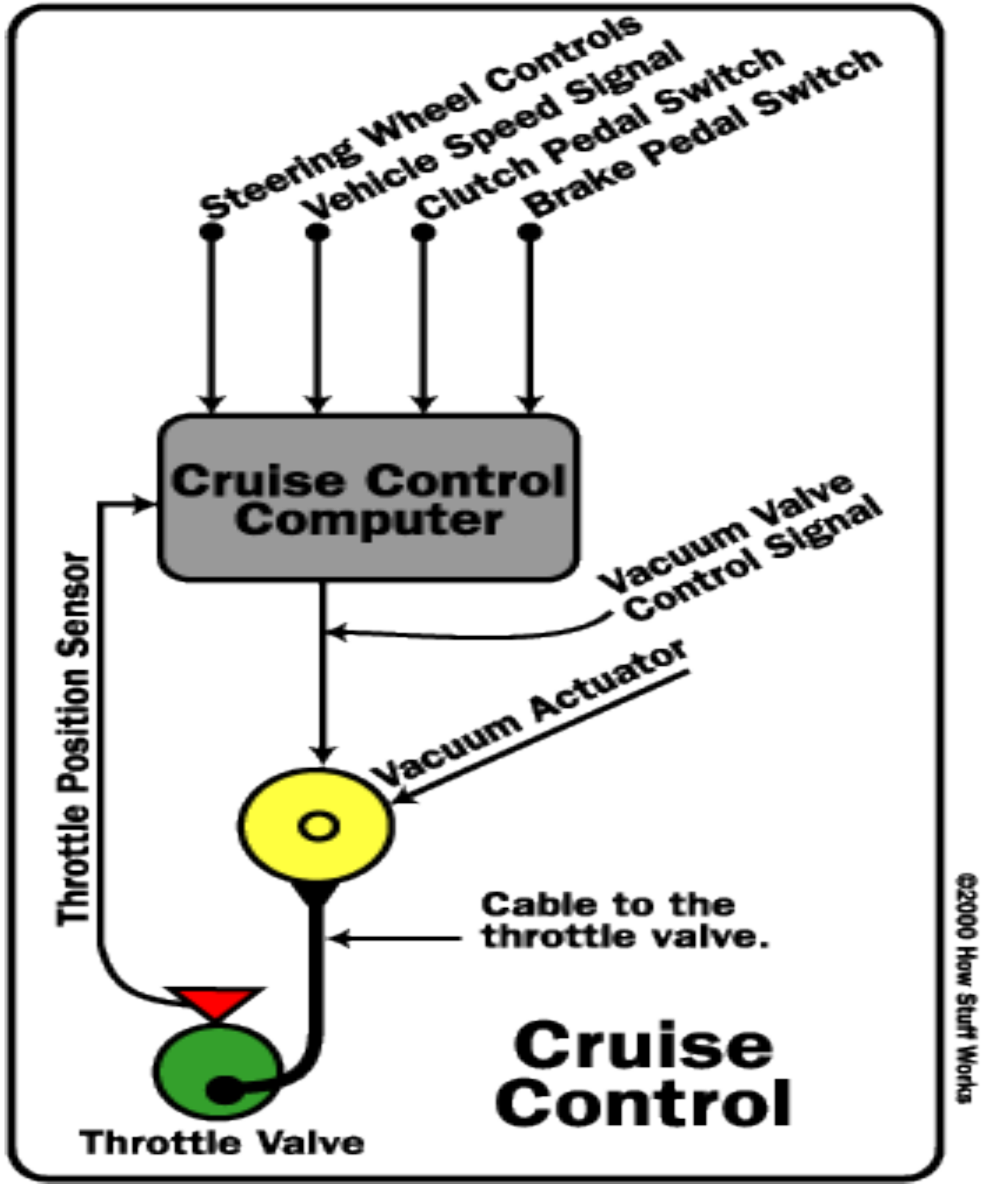
```
in position A;
while(1) {
    wait_until(form_empty);
    on(forward);
    wait_until(B);
    off(forward);
    on(backward);
    wait_until(A);
    off(backward);
}
```

temperature control:

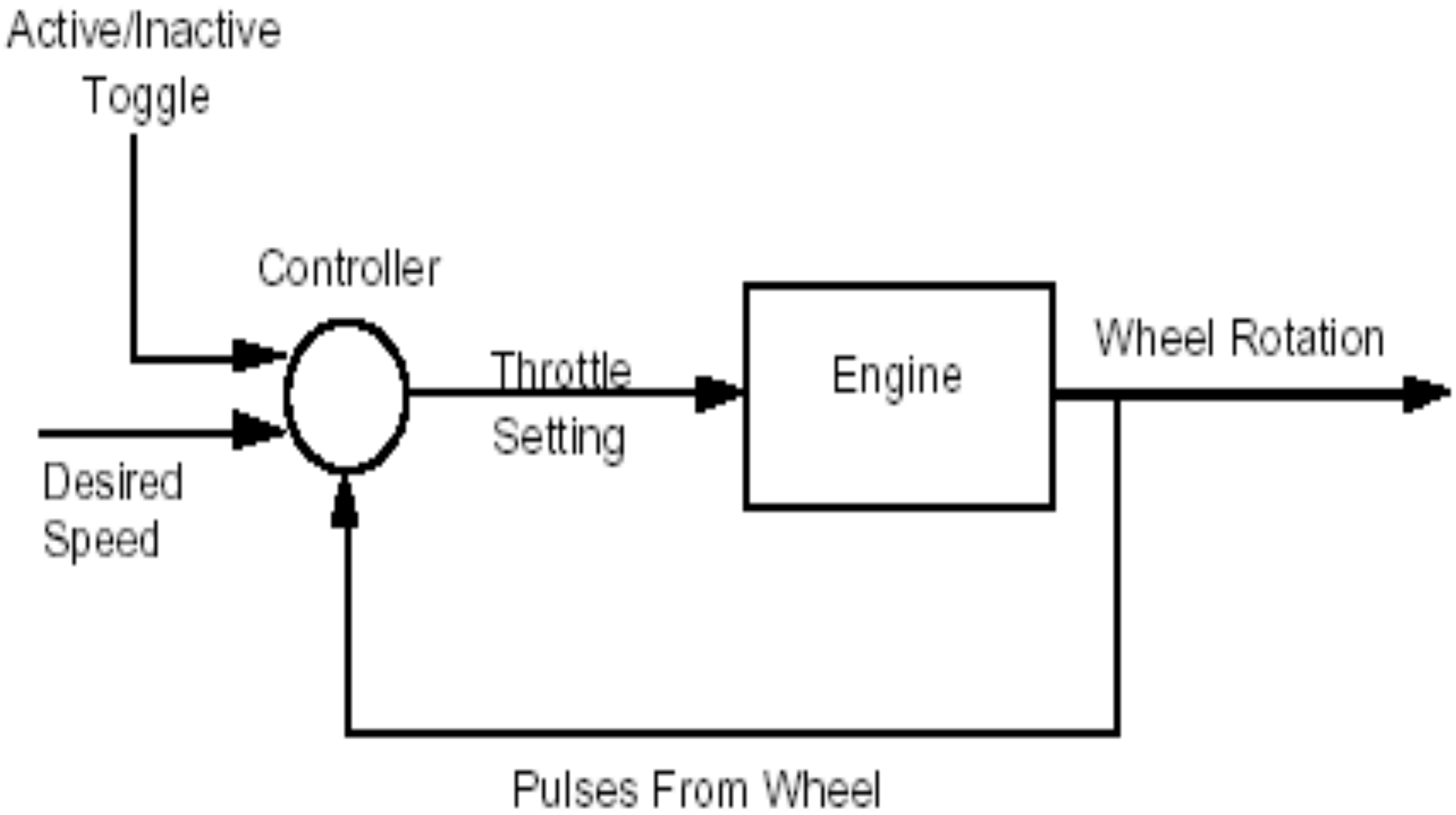
```
while(1) {
    analog_in(Temp);
    if (Temp > Tmax) {
        off(contactor);
    }
    else if (Temp < Tmin) {
        on(contactor);
    }
}
```


Automotive
Electronics

Example 3: Cruise Control

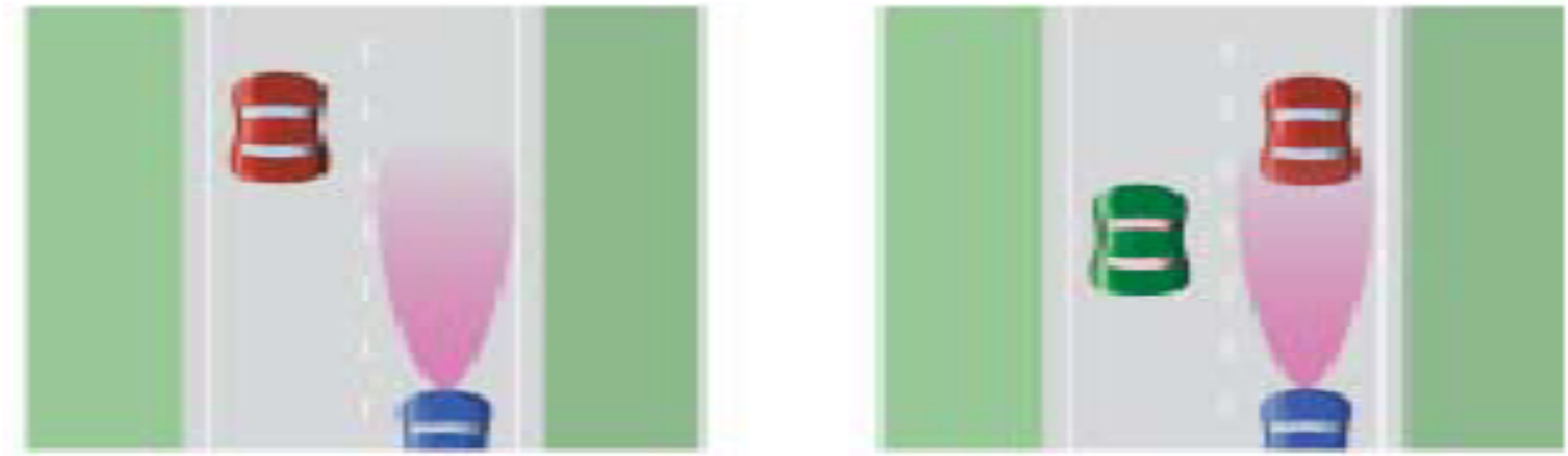


- Controls car speed
- Actuates the throttle valve by a cable connected to an actuator, instead of by pressing a pedal.
- The throttle valve controls the power and speed of the engine by limiting how much air the engine takes in .



Control Architecture for Cruise Control

Adaptive Cruise Control with Driver Alert



- Helps to reduce the need for drivers to manually adjust speed or disengage cruise control when encountering Slower traffic.
- Automatically manages vehicle speed to maintain a distance set by driver.
- Alerts drivers when slower traffic is detected in the path.
- Audible and visual alerts warn the driver when braking is necessary to avoid slower moving vehicles ahead.
- Drivers can adjust system sensitivity to their preferred driving style.

Plan

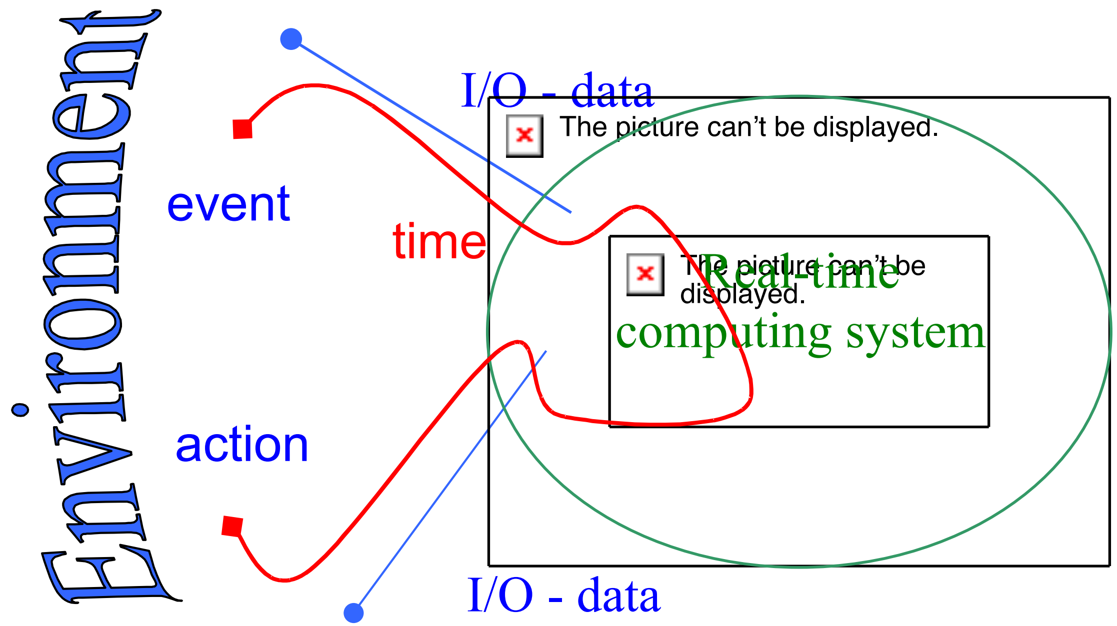
Real-Time Support

- Special Characteristics of Real-Time Systems
- Real-Time Constraints
- Canonical Real-Time Applications
- Scheduling in Real-time systems
- Operating System Approaches

What is "real" about real-time?

computer world	*real world*
e.g., PC	industrial system, airplane
average response for user, interactive	events occur in environment at own speed
occasionally longer	reaction too slow: deadline miss
reaction: user annoyed	*reaction:* damage, pot. loss of human life
computer controls speed of user	*computer must follow speed of environment*
"computer time"	**"real-time"**

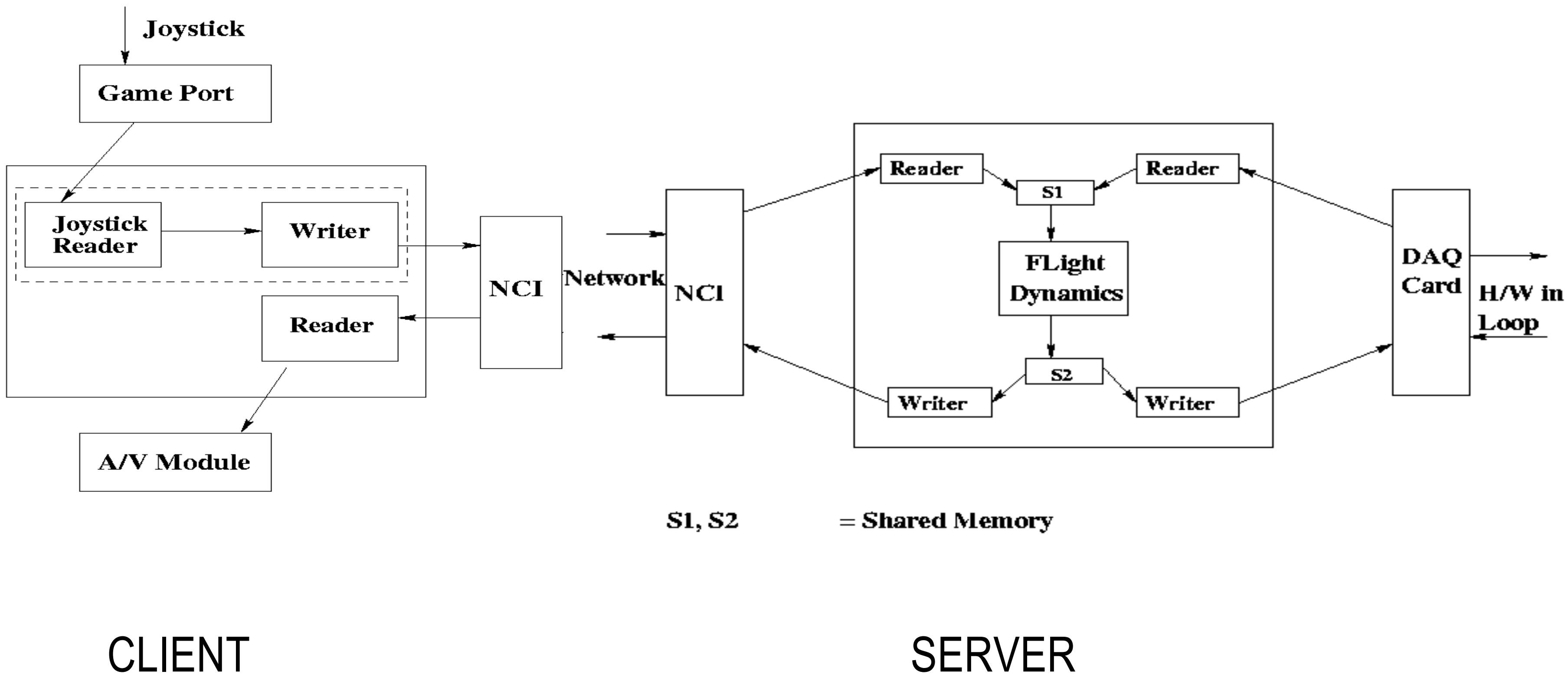
Real-Time Systems



A real-time system is a system that reacts to events in the environment by performing predefined actions

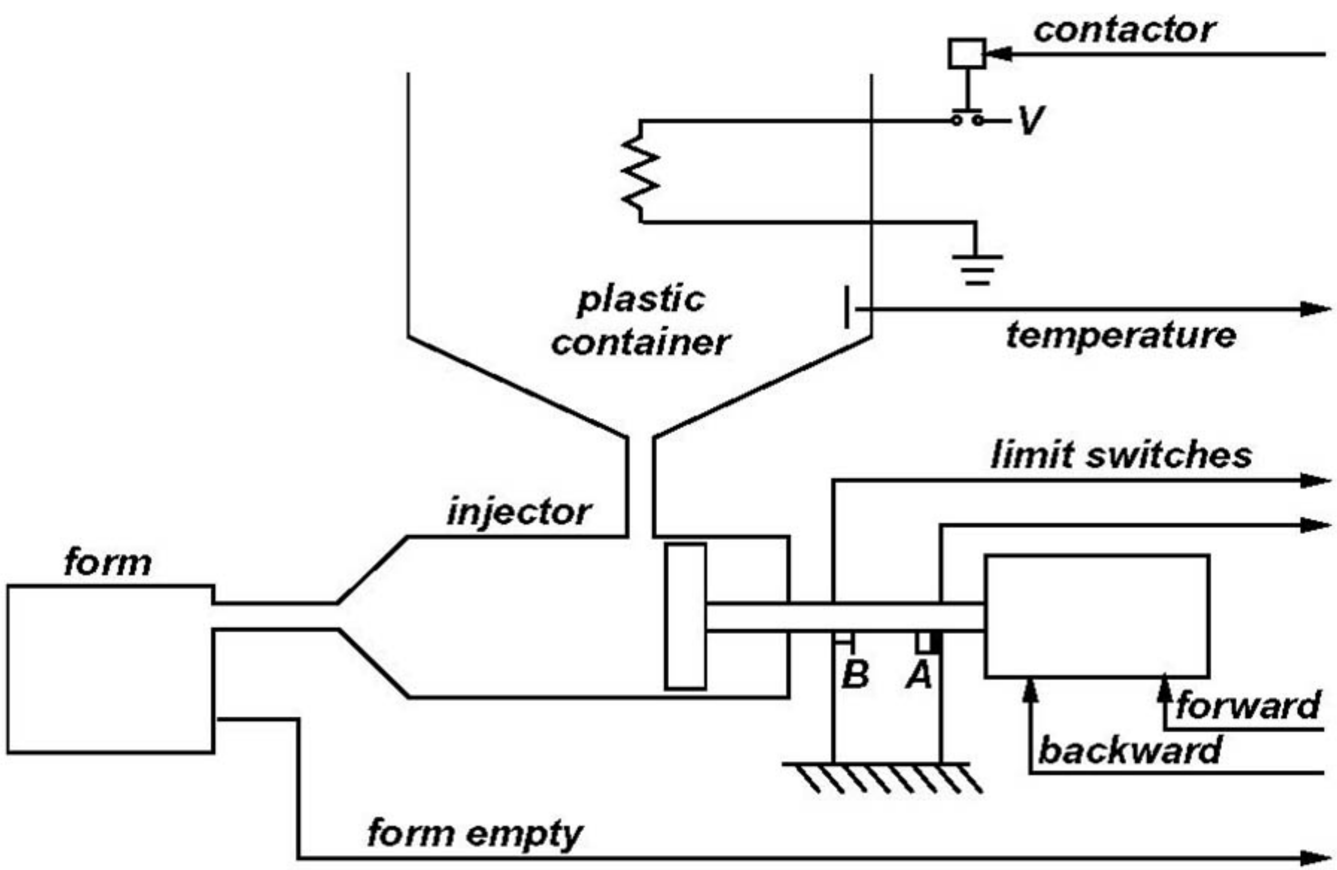
<u>within specified time intervals.</u>

Flight Avionics



Constraints on responses to pilot inputs, aircraft state updates

Example: injection molding



Constraints:

– **Keep plastic at proper temperature** (liquid, but not boiling)

– **Control injector solenoid**
(make sure motion of piston reaches end of its travel)

Real-Time Systems: Properties of Interest

- ***Safety***: Nothing bad will happen
- ***Liveness***: Something good will happen
- ***Timeliness***:
  Things will happen on time -- by their deadlines, periodically, ...

In a Real-Time System....

Correctness of results depends on value
*and its **time** of delivery*

correct value **delivered** ***too late*** **is** ***incorrect***

e.g., traffic light: light must be green ***when crossing*****, not enough before**

Real**-time:**

(Timely) reactions to events ***as they occur*****, at their pace:**

(real-time) system (internal) time same time scale as environment (external) time

Performance Metrics in Real-Time Systems

- Beyond minimizing response times and increasing the throughput:
  - *achieve timeliness*.
- More precisely, how well can we predict that deadlines will be met?

Types of RT Systems

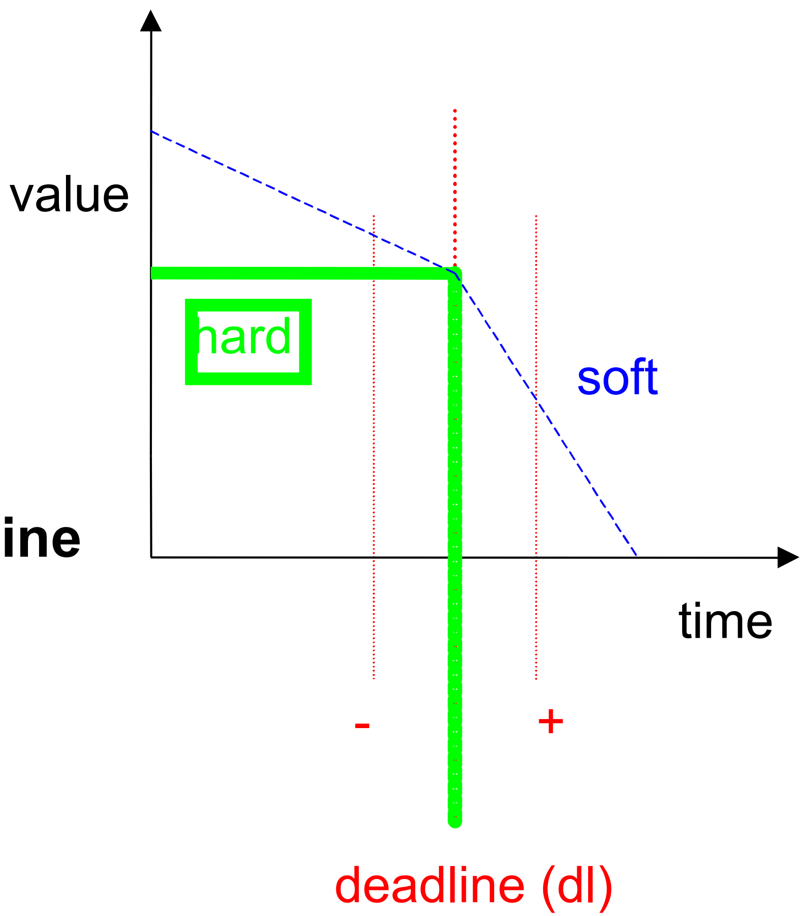
Dimensions along which real-time activities can be categorized:

- *How tight are the deadlines?*
  --deadlines are tight when **laxity** (deadline -- computation time) is small.
- *How strict are the deadlines?*
  --what is the value of executing an activity after its deadline?
- *What are the characteristics of the environment?*
  --how static or dynamic must the system be?

Designers want their real-time system to be *fast*, *predictable*, *reliable*, *flexible*.

Hard, soft, firm

- **Hard**
  result useless or dangerous if deadline exceeded
- **Soft**
  result of some - lower - value if deadline exceeded
- **Firm**
  If value drops to zero at deadline



Deadline intervals:
result required not later *and not before*

Examples

- **Hard real time systems**
  - Aircraft
  - Airport landing services
  - Nuclear Power Stations
  - Chemical Plants
  - Life support systems
  - …
- **Soft real time systems**
  - Multimedia
  - Interactive video games
  - ATM response
  - …

Real-Time: Items and Terms

Task

- program, perform service, functionality
- requires resources, e.g., execution time

Deadline

- specified time for completion of, e.g., task
- time interval or absolute point in time
- value of result may depend on completion time

Plan

- Special Characteristics of Real-Time Systems
- Real-Time Constraints
- Canonical Real-Time Applications
- Scheduling in Real-time systems
- Operating System Approaches

Timing Constraints

Real-time means to be in time ---

How do we know something is "in time"?

How do we express that?

- *Timing constraints* **are used to specify temporal correctness**

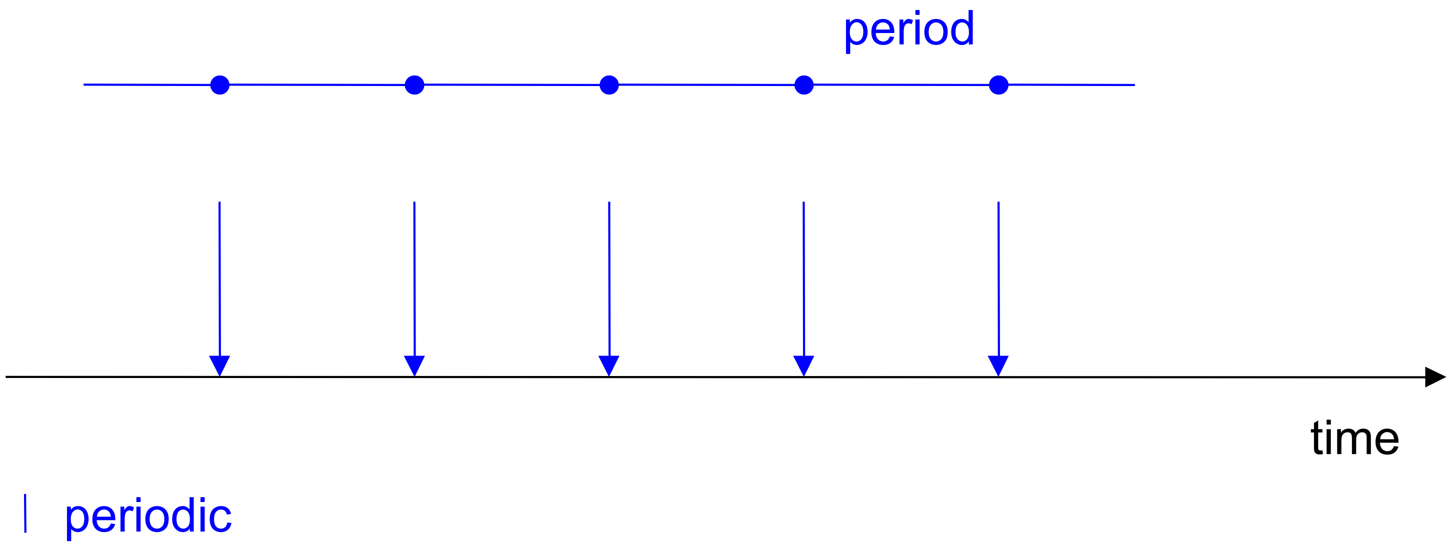
  e.g., "finish assignment by 2pm",

  "be at station before train departs".

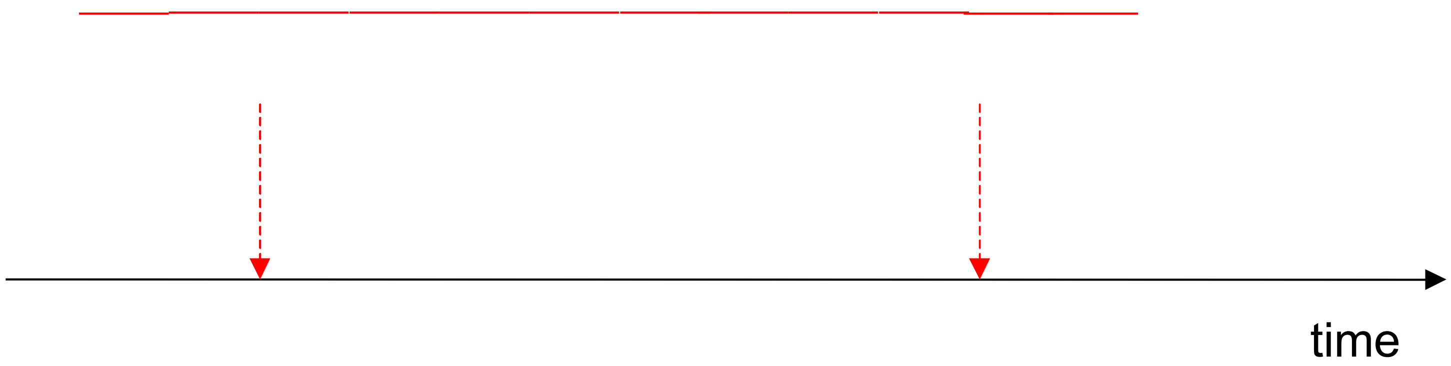
- **A system said to be *(temporally) feasible*, if it meets all specified timing constraints.**

- **Timing constraints do not come out of thin air: *design* process identifies *events*, derives models, and finally *specifies* timing constraints**

- **Periodic**
  - activity occurs repeatedly
  - e.g., to monitor environment values, temperature, etc.

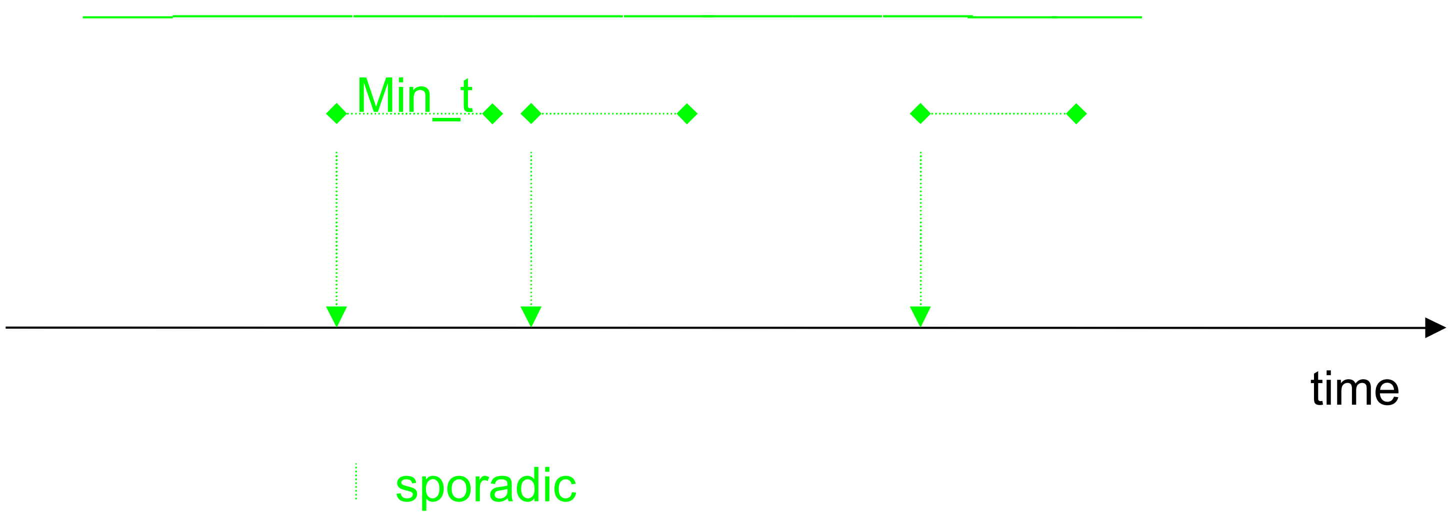


- **Aperiodic**
  - can occur any time
  - no arrival pattern given



aperiodic

- **Sporadic**
  - can occur any time, but
  - minimum time between arrivals



Who initiates (triggers) actions?

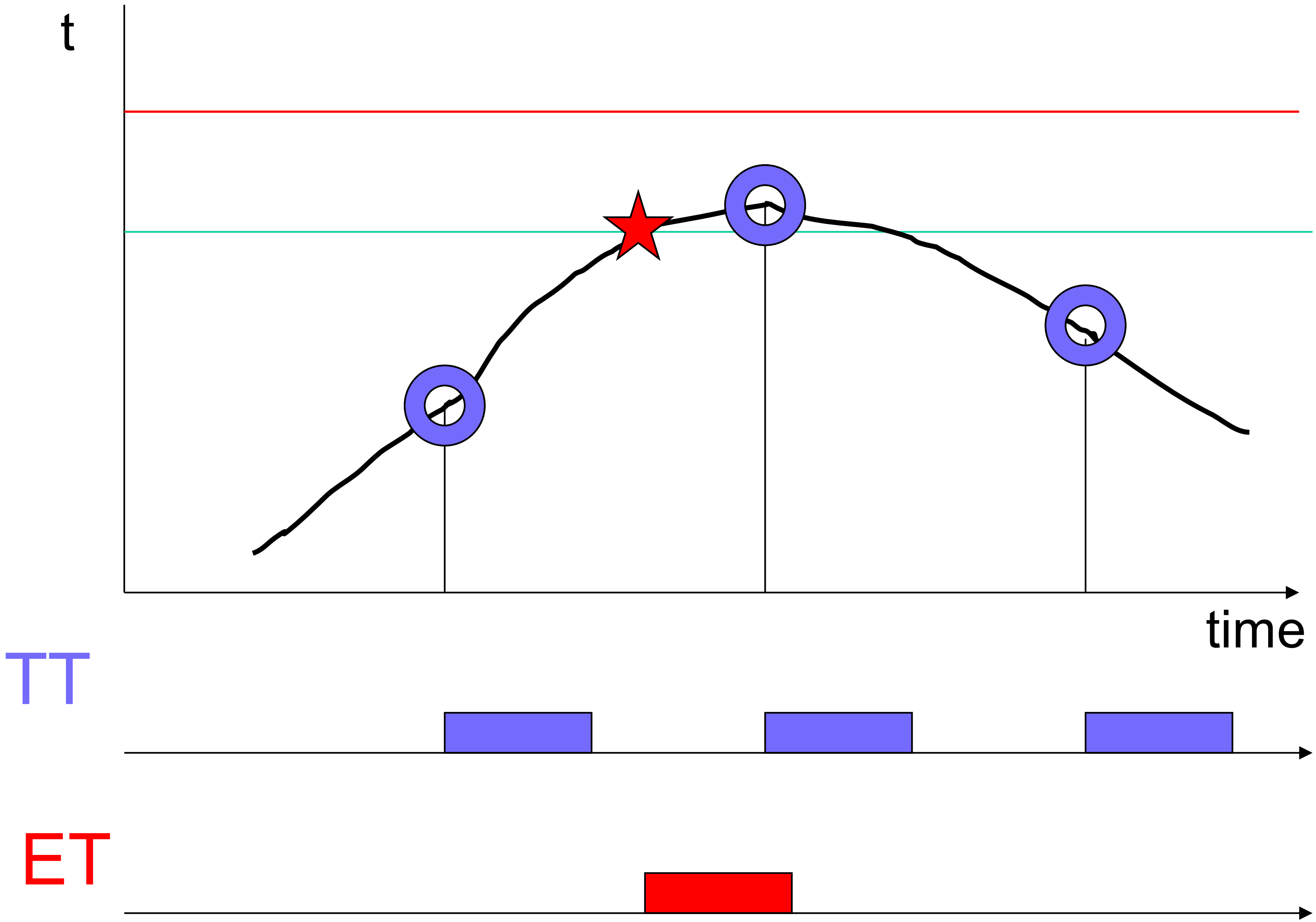
Example: Chemical process

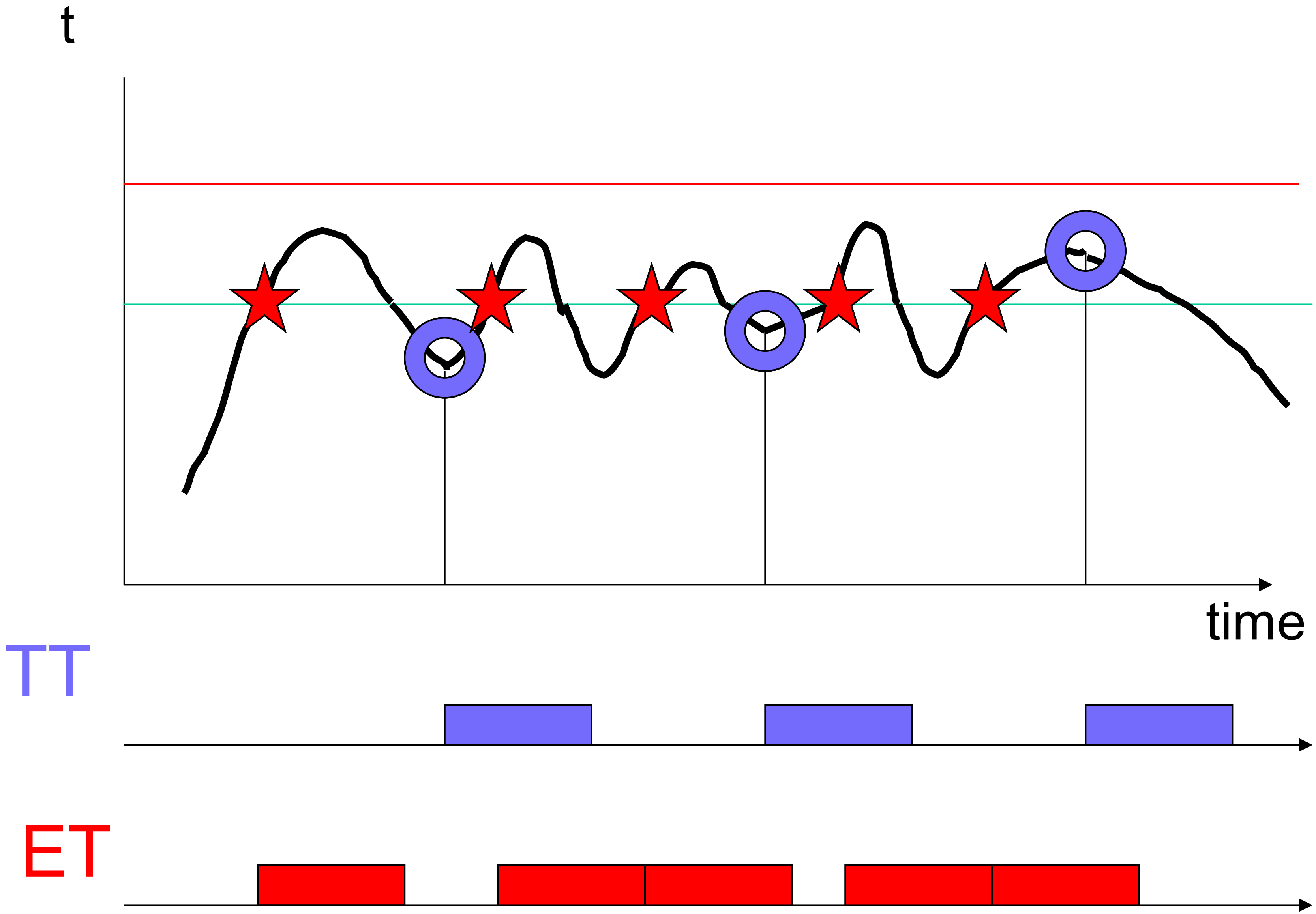
- Controlled so temperature stays below *danger* level
- Warning triggered before danger point

  …… so cooling can still occur

Two possibilities:

- Action whenever temp raises above *warn;*
  event triggered
- Look every *int* time intervals;
  action when temp if measures above *warn*
  time triggered


t
time
TT
ET


t
time
TT
ET

ET vs TT

- **Time triggered**
  – Stable number of invocations

- **Event triggered**
  – Only invoked when needed
  – High number of invocation and computation demands if value changes frequently

Slow down the environment?

- **Importance**
  - Which parts of the system are important?
  - Importance can change over time
    e.g., fuel efficiency during emergency landing
- **Flow control**
  Who has control over speed of processing?
  Who can slow partner down?
  - Environment
  - Computer system

  RT: environment cannot be slowed down

Other Issues to worry about

- **Meet requirements -- some activities may run only:**
  - After others have completed - *precedence constraints*
  - While others are not running - *mutual exclusion*
  - Within certain times - *temporal constraints*

- **Scheduling**
  - Planning of activities, such that required timing is kept

- **Allocation**
  - Where should a task execute?

In Summary

- **Examples:**
  - Flight simulator
  - Injection molding
  - Automobile Cruise Control
- **Definitions:**
  - What is “real” about realtime systems
  - Performance metrics in realtime systems: “timeliness”
  - Types of RT systems: nature of deadlines, hard, soft, firm
- **Timing constraints**
  - Periodic, aperiodic, sporadic
  - Event driven vs time-triggered systems
  - Other issues: requirements, scheduling, resource allocation