

# CS4220 Embedded Systems CS6235 Real-Time Systems

## 5B: Feedback-Based EDF

**Instructor: Calton Pu**  
**calton.pu@cc**

TA: Younggyun Koh (young@cc)

1

## Earliest Deadline First

- Assumptions
  - The schedule is feasible
  - Tasks are independent
- Dynamic priority scheduler
  - Look at all tasks in the queue
  - Compare their deadlines
  - Earliest Deadline First (EDF)
- Minimizes deadline failures when feasible
  - Another story when schedule is infeasible

2

## Discussion of EDF

- Optimal dynamic scheduler
  - Up to 100% WCAU
- Several practical problems
  - Overhead of dynamic scheduling
  - Instability under overload
  - “Priority inversion”

3

## Minimum Laxity First

- Similar to EDF, improved
  - Laxity = time to latest feasible start time (when the task can still complete)
  - Run the task closest to failing first
  - Optimal in minimizing deadline failures
  - More graceful degradation
- Comparison with EDF
  - MLF = EDF when all tasks have same length

4

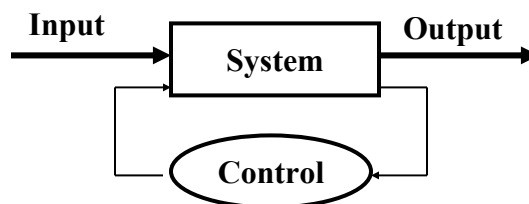
# Control Systems

- Systems with changes
  - Human-controlled: bicycle, cars, airplane
  - Inherently unstable airplane wings
  - Chemical reactions, nuclear power plants
- Basic assumptions
  - Changes are within a certain range
  - Changes are “continuous” (physical world)

5

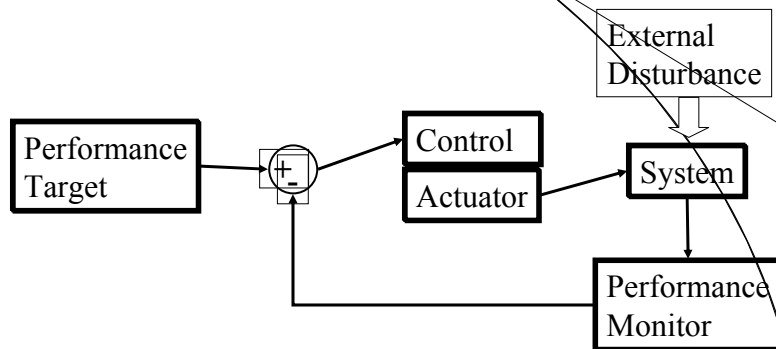
# Feedback Approach

- Basic control loop
  - Observe the system being controlled
  - Compare current behavior with expected
  - If different, make adjustments



6

# Control Components



7

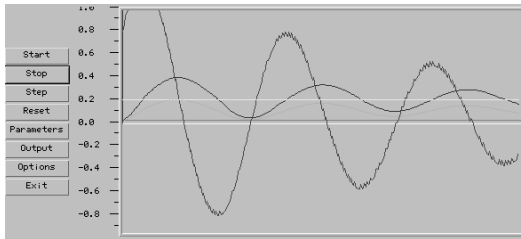
# Phase Locked Loop

- Simple example: FM Radio
  - Frequency modulation over a channel
  - Channel is a range of spectrum: 90.1MHz
  - Broadcast frequency moves over range
- Simple loop: PLL
  - Knows about the range, frequency move
  - Moves along with the broadcast

8

# Predictability of Control

- Control system properties
  - Stability: does the steady state converge?
  - Maximum error: how far from target?
  - Responsiveness: how fast does it react?



9

# PID Control

- Proportional-Integral-Derivative
- See equation (1), p. 57
- Linear combination of 3 components
  - Proportional:  $C_p$
  - Integral:  $C_i$
  - Derivative:  $C_D$

10

# Advantages of PID

- Systematic design method
  - Given predictable input parameters
  - Design control components by composition of filters (Superposition Theorem)
- Predictable performance
  - Responsiveness, maximum error, stability
  - Sampling Theorem: need to sample at twice the frequency of phenomenon

11

# PID Design Tradeoffs

- Example: minimize response time
  - React immediately to input change
  - Leads to instability during spikes
- Integral filter adds stability
  - Average the input signals over a window (smooth out the spikes)
  - Reduces instability, delays response

12

## FC-EDF

- Goal of the scheduler
  - Maximize utilization & minimize MissRatio
  - EDF maximizes utilization anyway
  - Feedback controls maximum allowed utilization level to minimize MissRatio
- Control knobs (figure 2, p. 59)
  - Admission control: rejection of tasks
  - Service level: multiple versions of tasks

13

## PID Controller

- Observes the MissRatio of system
  - Calculates missed deadlines in a window
  - Sampling rate: every  $SP$  seconds
- Control signal ( $\Delta CPU$ ) on utilization
  - Calls Service Level for “quick” reaction (small adjustment of accepted tasks)
  - Admission Control called if needed (slower impact on utilization)

14

## Service Level Adjustment

- Task service levels
  - Logical versions of tasks that require more or less CPU
  - Concrete example: imprecise computations
- Adjustments that change utilization
  - Higher service level uses more CPU
  - Typical scenario: starts at high level and gets reduced during overload

15

## Admission Control

- Stops tasks from entering queue
  - Does not change accepted task queue
  - Reduces CPU utilization slowly
- Can only reduce task queue length
  - Service level is bi-directional control
- Potential trade-off with service level
  - Admit tasks at low level service


16

## Discussion

- Disadvantages
  - Coarse-grained adaptation (seconds)
  - Detects missed deadlines and then react
  - Applicable to stable workloads
- Advantages
  - Close enough to classic feedback model for a classic analysis (section 3.6)

17

## Stability Analysis

- BIBO stability
  - Bounded Input  Bounded Output
- Assumptions
  - Utilization tracking is good (bounded error)
  - Tasks are independent
- Stability condition (Theorem 1, p. 61)
  - Coefficients defined well

18

## Other System Parameters

- Maximum error
  - How far can the utilization tracking drift?
- Responsiveness
  - How quickly will the controller respond to input changes?
- Tradeoffs between parameters

19

## Performance Evaluation

- Simulation-based (figures on p. 63)
  - MRA: miss ratio among admitted tasks
  - UTIL: actual CPU utilization
  - HRS: hit ratio among submitted tasks (successful execution ratio)
  - VCR: value completion ratio (higher service level contributes higher value)
  - etf: estimated time factor

20

# Scheduler Adaptation

- FC-EDF (figure 9, p. 65)
  - Watch the adaptation of FC-EDF
  - etf spike (300 SP) causes misses
  - etf loss (600 SP) causes low utilization
- EDF and EDF-AC
  - EDF (figure 11): high miss ratio
  - EDF-AC (figure 10): improves on EDF

21

# Summary

- EDF, EDF-AC
  - Earliest Deadline First (naïve)
  - EDF with admission control
- FC-EDF
  - Feedback control of service level & AC
  - Stability analysis
  - Performance evaluation

22