Abstraction of Cyber-Physical Interplays and Its Application to CPS Design

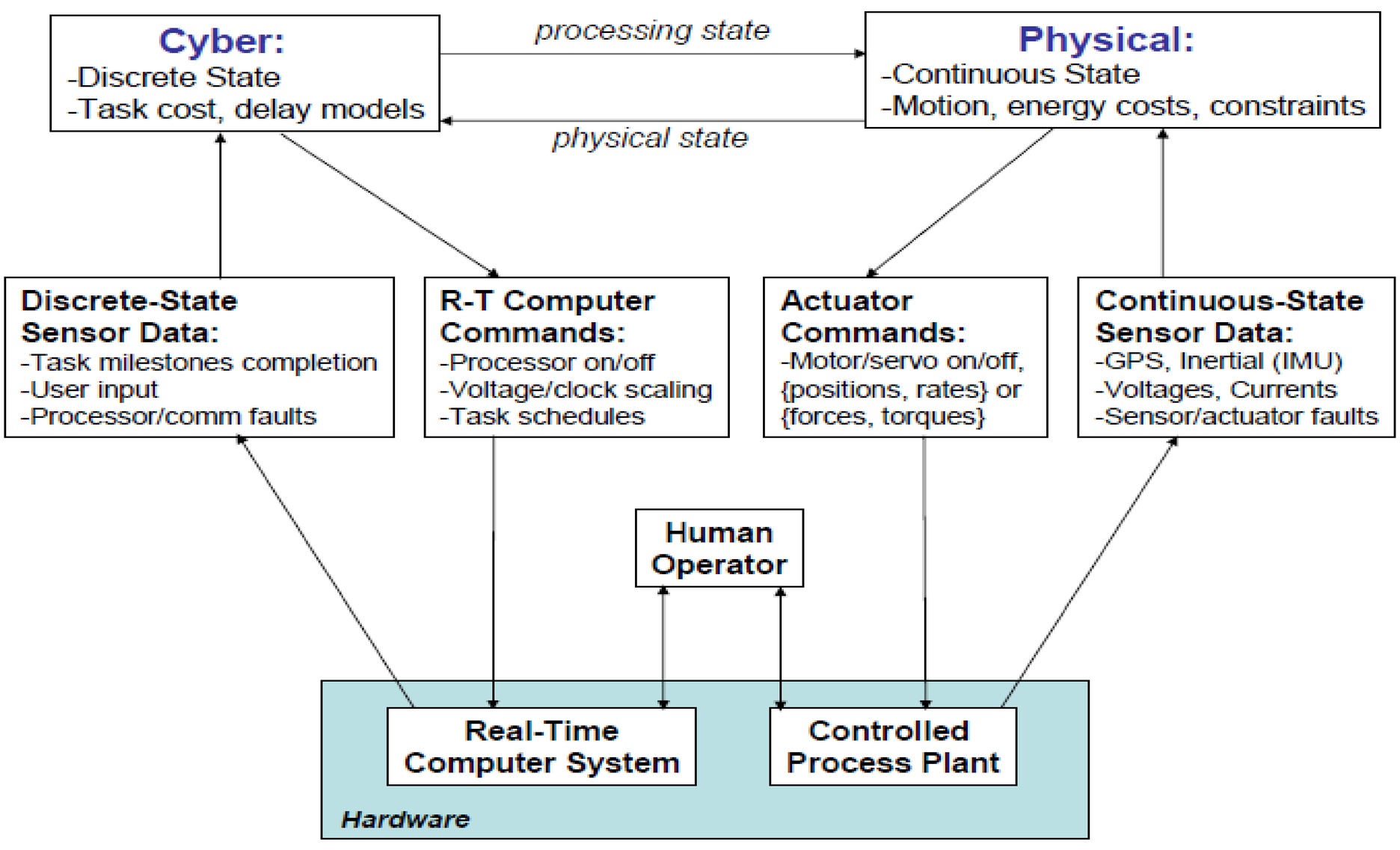
K. G. Shin, E. M. Atkins
University of Michigan

I. Koren, C. M. Krishna
University of Massachusetts

Objective: To facilitate

- Effective codesign of real-time computer and controlled plant (offline)
- Efficient online use of computational and plant resources to ensure
  - Fault Tolerance
  - Power/energy efficiency
  - Superior controlled plant performance

Abstraction Duality in CPS



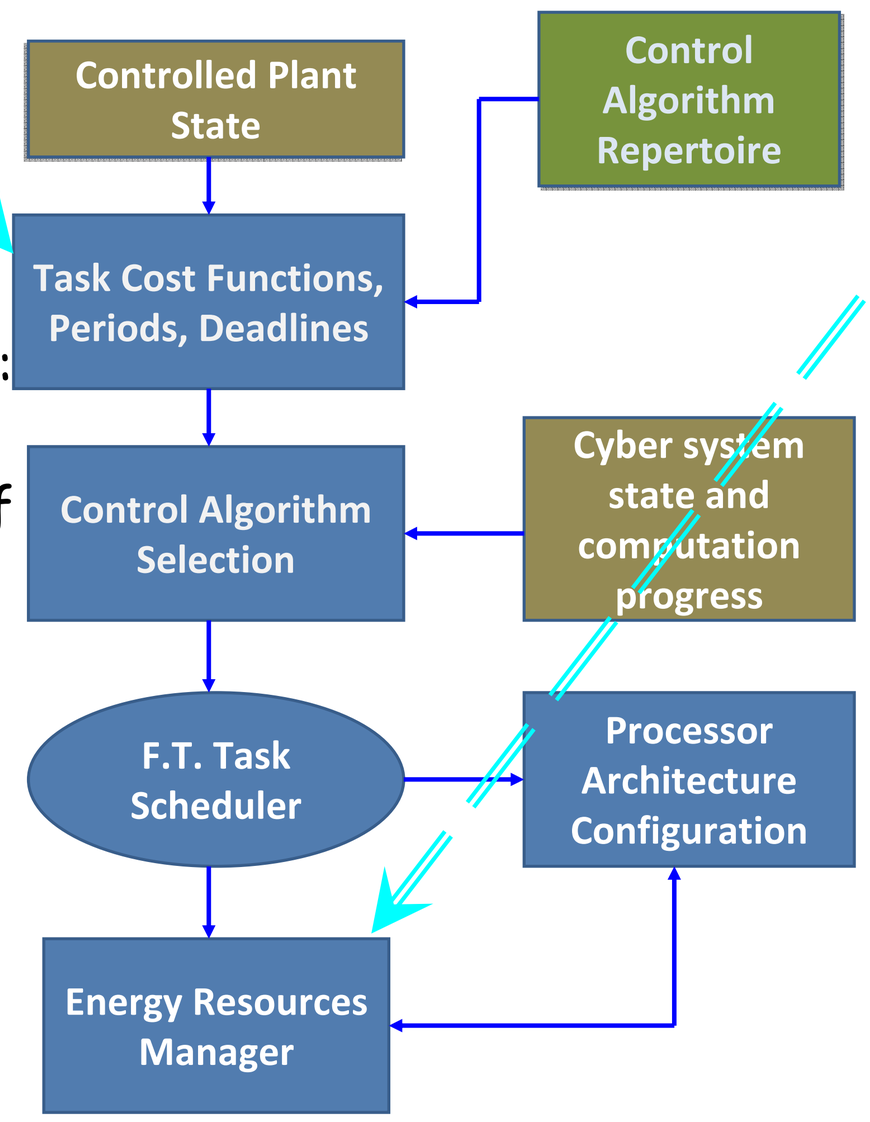
Interfaces

- **Application-to-Computer Interface**: Specifies how the application's QoS degrades as a function of
  - Current application and environmental state
  - Computational delays
  - Computational imprecision
  - Residual mission lifetime
- **Computer-to-Application Interface**: Specifies
  - Task periods, precision levels, and priority rules
  - Use of energy reserves
  - Task milestones completion
  - Computational health

Abstraction of Cyber-Physical Interplays and Its Application to CPS Design

Capturing the Cost of Computational Delay

- Cost functions link controlled process & controlling computer:
  - Quantify degradation in control quality as a result of computational (feedback) delays
  - For an independent task, cost = difference in performance functional with, and without, delay
  - When tasks are dependent, cost function becomes multidimensional

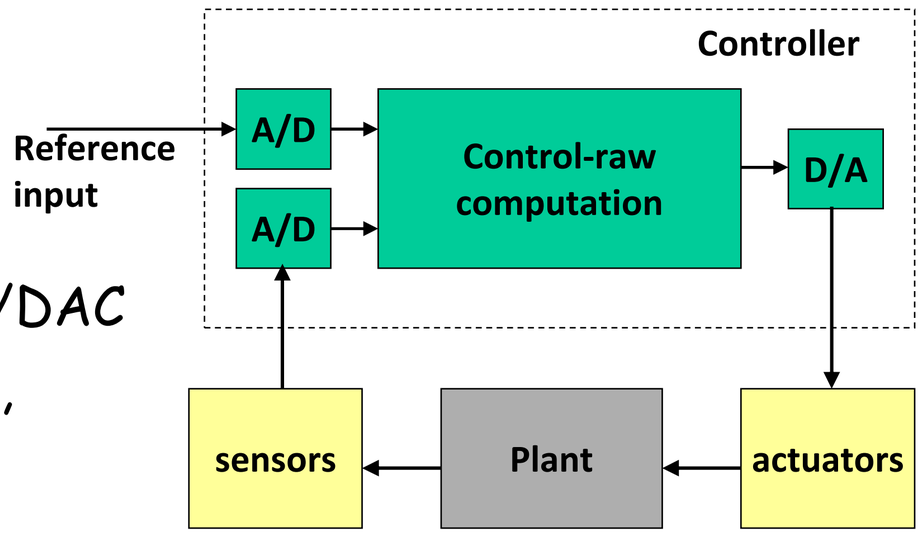


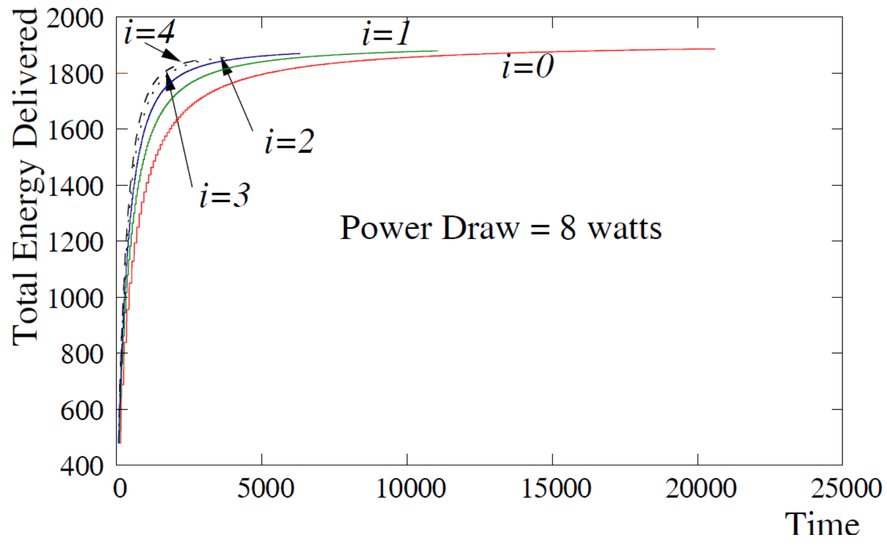
Battery/Supercapacitor Management

- Batteries have
  - High energy/weight ratio
  - Limits power delivery
- Supercapacitors have:
  - Lower energy/weight ratio
  - Excellent capability to deliver high power surges for short periods of time
  - Use in tandem to combine advantages of both:
    - Battery - backup energy reservoir to keep the supercap charged
    - Supercap - provide high-power pulses

CPS Control System

- Cyber: Controller, ADC/DAC
- Physical: Plant, Sensors, actuators
- Coupling: time

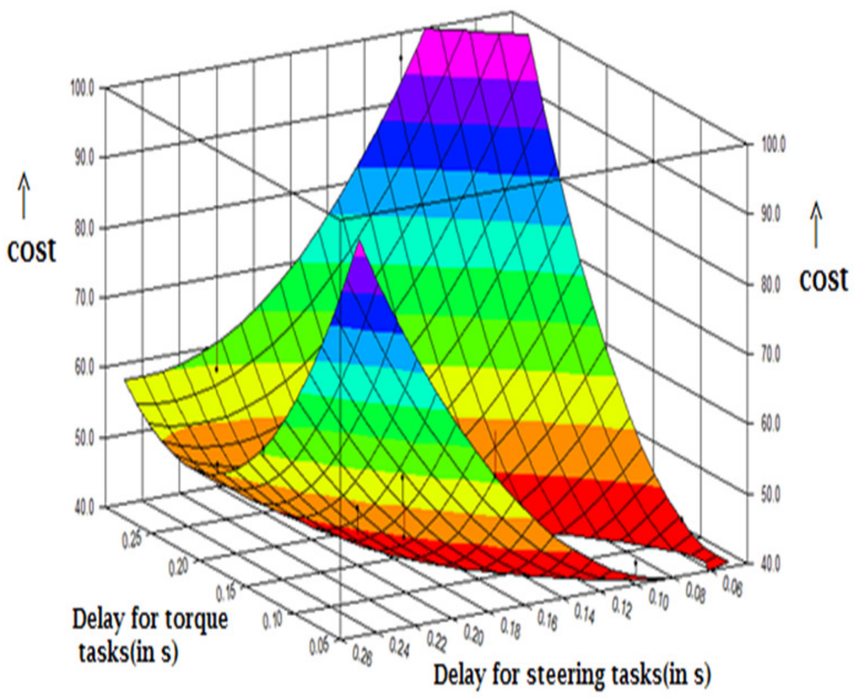


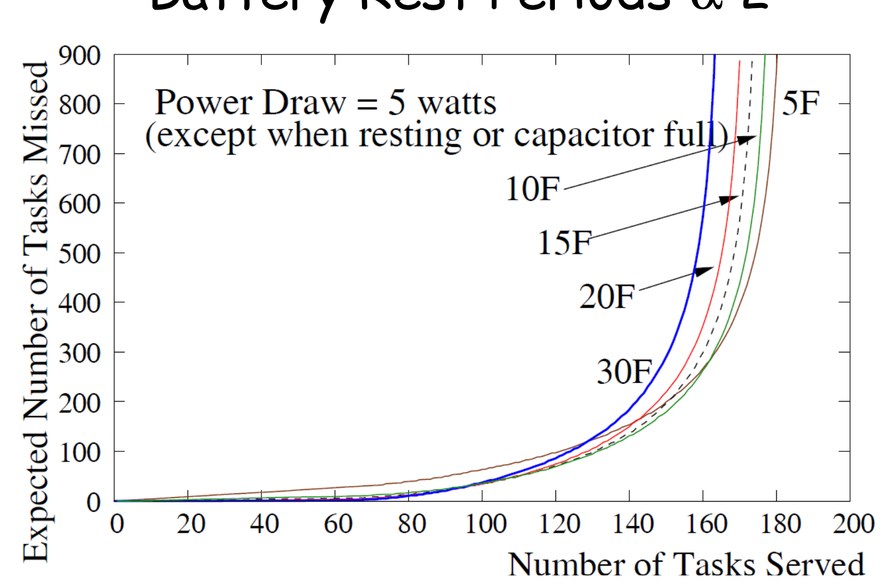


Battery Rest Periods α 2^{-i}

Example 1: Automobile Control

- Torque and steering calculated and applied
- Objective: track desired trajectory
- Cost function – the area between actual and desired trajectories

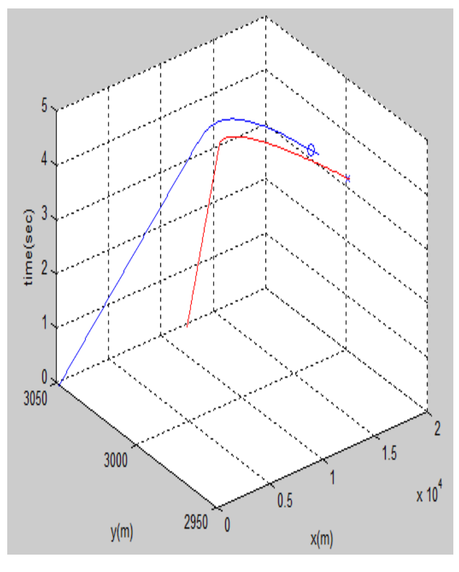




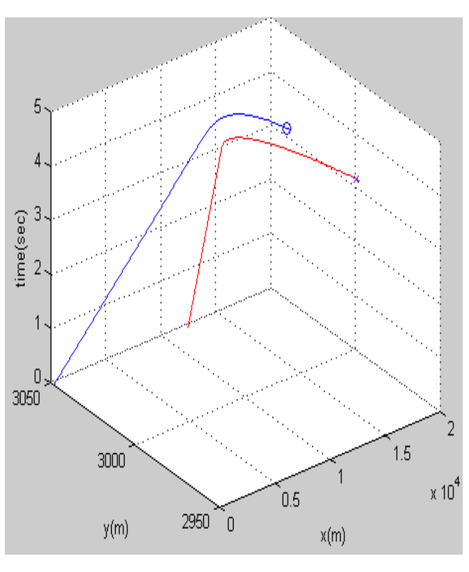
Capacitor range: 0.74-2.75V
Task arrival rate: 0.5/sec
Energy per task: 10 joules

Example 2: Chasing an evasive target

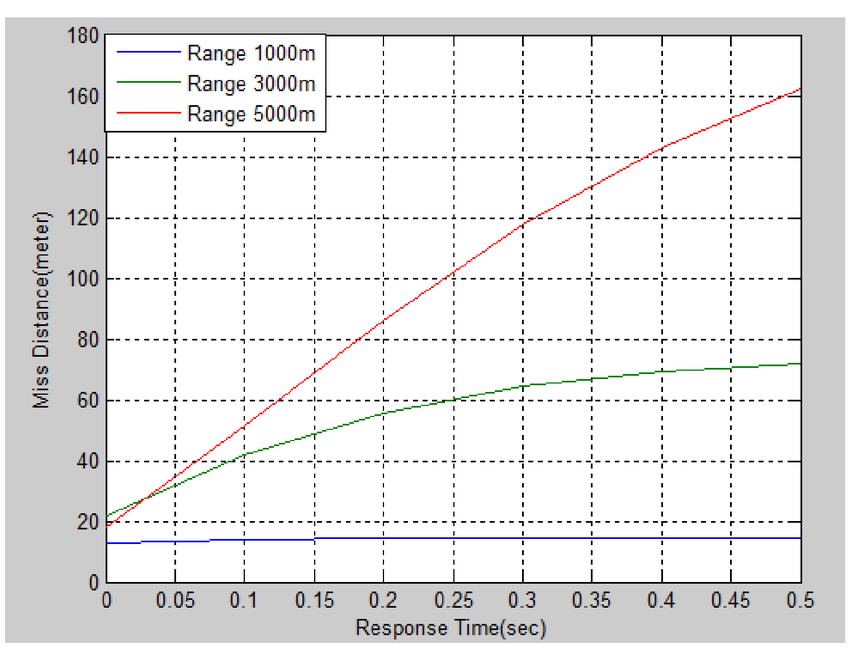
- A point target maneuvering (at 10g max) when at a 3km distance from chaser
- Approximated 1st order model



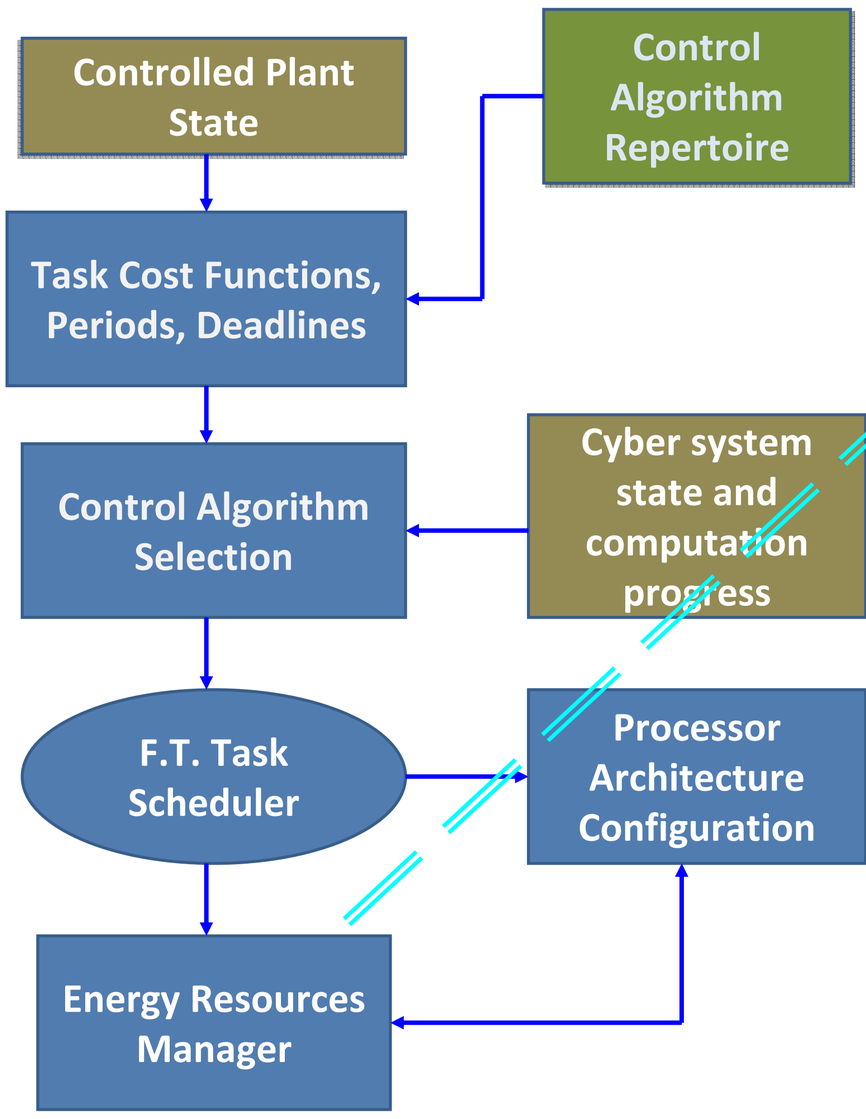
Computer delay 0.01s



Computer delay 0.1s



Abstraction of Cyber-Physical Interplays and Its Application to CPS Design



Thermal-aware Scheduling

- Thermal hazards exist in modern multi-core chips with nano-scale CMOS technolog
- Large temperature variance and existence of hotspots degrade chip reliability
- Approach: Minimize temperature variance on a multi-core chip while meeting the real-time application's timing constraints

Proposed Solution

- Assume existence of on-chip temperature sensors deployed on a multi-core chip
  - Periodic measurement of core temperature
- Based on measured temperature values, a runtime *Thermal-aware Scheduler* (TAS) adjusts the voltage/frequency of cores periodically
  - To meet the thermal constraints while guaranteeing the applications' timing constraints
- Use of per-core DVS
  - On-chip regulator to guarantee the independent power/clock domain of each core
  - The voltage switching overhead is tens of nano-seconds
- Use of job migration
  - The speed of a hot core is reduced further by redistributing the assigned workload to other cool cores
- Obtain the thermal model of a multi-core chip
  - Estimate the correlation of the voltages/frequencies of cores and their temperature behavior before runtime
- Require the warm-up period to estimate the thermal model before the system starts to operate

The period of TAS invocation

Unknown parameters to be estimated before runtime

$$\vec{T}(t+p) = A \cdot \vec{T}(t) + B \cdot U(t)$$

Predicted core temperature

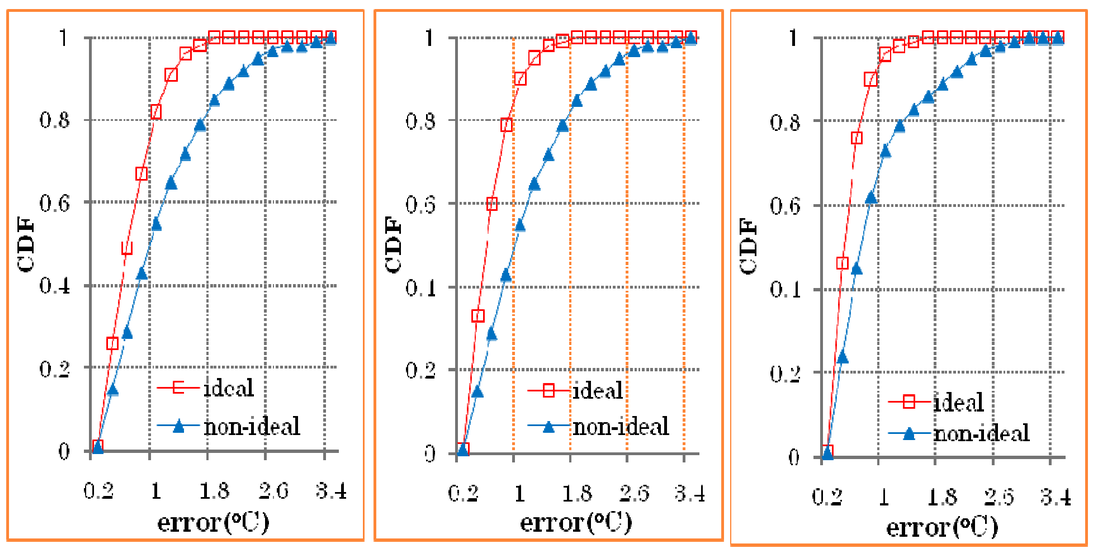
Measured core temperature

$$U(t) = \begin{bmatrix} V_1 \\ \dots \\ V_n \end{bmatrix}$$

A discrete set of voltages to be determined periodically by TAS

Results

1. HotSpot simulator used with randomly-generated task sets.
2. MCPat and Wattch power simulator used to estimate power characteristics of tasks.
3. Period of TAS: 40msec.
4. "Ideal" – when all tasks have the same power dissipation.



4-core 8-core 16-core

- Evaluation of the TAS
- Statistics of max. core temperature and temperature variations on 16-core chip: