Feedback Control for Real-Time Systems

Chenyang Lu
Cyber-Physical Systems Laboratory
Department of Computer Science and Engineering

Washington University in St. Louis

CPS Week 2013
Outline

- CPU Utilization Control for Distributed Real-Time Systems
  - Model Predictive Control

- Thermal Control for Real-Time Systems
  - Nested Control Design
Outline

- CPU Utilization Control for Distributed Real-Time Systems
  - Model Predictive Control

- Thermal Control for Real-Time Systems
  - Nested Control Design
Control for Distributed Real-Time Systems

- Common characteristics of computing problems
  - MIMO: multi-input (knobs), multi-output (objectives)
  - Coupling between objectives.
  - Constraints on knobs.

- Model Predictive Control
  - Optimization + Prediction + Feedback
Why CPU Utilization Control?

- Overload protection
  - CPU over-utilization $\rightarrow$ system crash

- Meet response time requirement
  - CPU utilization < bound $\rightarrow$ meet deadlines
Challenge: Uncertainties

- Execution times?
  - Unknown sensor data or user input

- Request arrival rate?
  - Aperiodic events
  - Bursty service requests

- Disturbance?
  - Denial of Service attacks

Control-theoretic approach

→ Robust utilization control in face of workload uncertainty
End-to-End Tasks in Distributed Systems

- Task $T_i$: sequence of subtasks $\{T_{ij}\}$ on different processors
  - Periodic: All the subtasks of a task run at a same rate.
- Task rate can be adjusted
  - Within a range
  - Higher rate $\Rightarrow$ higher utility

![Diagram](image-url)
Problem Formulation

- \( B_i \): Utilization set point of processor \( P_i \) \((1 \leq i \leq n)\)
- \( u_i(k) \): Utilization of \( P_i \) in the \( k^{th} \) sampling period
- \( r_j(k) \): Rate of task \( T_j \) \((1 \leq j \leq m)\) in the \( k^{th} \) sampling period

\[
\min_{\{r_j(k)|1 \leq j \leq n\}} \sum_{i=1}^{n} (B_i - u_i(k))^2
\]

subject to rate constraint:

\[
R_{\min,j} \leq r_j(k) \leq R_{\max,j} \quad (1 \leq j \leq m)
\]
Single-Input-Single-Output (SISO) Control
Single Processor

Set point
\( U_s = 69\% \)

Task Rates
\( R_1: [1, 5] \text{ Hz} \)
\( R_2: [10, 20] \text{ Hz} \)

New in Distributed Systems

- Need to control utilization of multiple CPUs
- Utilization of CPUs are coupled due to end-to-end tasks
  → Replicating a SISO controller on all processors does not work!
- Constraints on task rates
EUCON: Multi-Input-Multi-Output Control

Control Design Methodology

1. Derive a dynamic model of the system
2. Design a controller
3. Analyze stability
Dynamic Model: One Processor

\[ u_i(k) = u_i(k - 1) + g_i \sum_{T_{jl} \in S_i} c_{jl} \Delta r_j(k - 1) \]

- \( S_i \): set of subtasks on \( P_i \)
- \( c_{jl} \): estimated execution time of \( T_{il} \)
- \( g_i \): utilization gain of \( P_i \)
  - ratio between actual and estimated change in utilization
  - models uncertainty in execution times
Dynamic Model: Multiple Processors

\[ u(k) = u(k-1) + GF\Delta r(k-1) \]

- **G**: diagonal matrix of utilization gains
- **F**: subtask allocation matrix
  - models the coupling among processors
  - \( f_{ij} = c_{ji} \) if task \( T_j \) has a subtask \( T_{ji} \) on processor \( P_i \)
  - \( f_{ij} = 0 \) if \( T_j \) has no subtask on \( P_i \)

\[
\begin{bmatrix}
T_1 & T_{11} \\
T_2 & T_{21} \\
& \end{bmatrix}
\begin{bmatrix}
& T_{22} \\
& T_3 \\
& T_{31} \\
& \end{bmatrix}
\]

\[
F = \begin{bmatrix}
c_{11} & c_{21} & 0 \\
0 & c_{22} & c_{31}
\end{bmatrix}
\]
Model Predictive Control

- Suitable for coupled MIMO control problems with constraints.

- Compute input to minimize cost over a future interval.
  - Cost function: tracking error and control cost.
  - Predict cost based on a system model and feedback.
  - Compute input subject to constraints.

- Optimization + Prediction + Feedback
Cost Function

- **Cost**

\[ V(k) = \sum_{i=1}^{P} \left\| u(k + i) - \text{ref}(k + i) \right\|^2 + \sum_{i=0}^{M-1} \left\| \Delta r(k + i) - \Delta r(k + i - 1) \right\|^2 \]

- **Tracking Error**

- **Control Cost**

- **Reference trajectory**: exponential convergence to \( B \)

\[ \text{ref}(k + i) = B - e^{-\frac{T_s}{T_{ref}} i} (B - u(k)) \]
Model Predictive Controller

At the end of each sampling period
- Compute inputs in future sampling periods
  \( \Delta r(k), \Delta r(k+1), \ldots, \Delta r(k+M-1) \)
  to minimize the cost function
- Cost is predicted using
  1. feedback \( u(k-1) \)
  2. approximate dynamic model
- Apply \( \Delta r(k) \) to the system

At the end of the next sampling period
- Shift time window and re-compute \( \Delta r(k+1), \Delta r(k+2), \ldots, \Delta r(k+M) \) based on feedback
EUCON Controller

Model Predictive Controller

System Model

Rate Constraints

Least Squares Solver

Cost Function

Reference Trajectory

\[
\begin{bmatrix}
B_1 \\
\vdots \\
B_n \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
u_1(k) \\
\vdots \\
u_n(k) \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
\Delta r_1(k+1) \\
\vdots \\
\Delta r_m(k+1) \\
\end{bmatrix}
\]

Constrained optimization solver

Desired trajectory for \(u(k)\) to converge to \(B\)

Difference from reference trajectory

CPS Week 2013
Stability Analysis

- Stability: utilization of all processors converge to set points
- Derive stability condition $\rightarrow$ range of $G$
  - Tolerable variation of execution times

$\rightarrow$ Provides analytical assurance despite uncertainty
Stable System

execution time factor = 0.5
(actual execution times = \( \frac{1}{2} \) of estimates)
Unstable System

execution time factor = 7
(actual execution times = 7 times estimates)
Stability

- Stability condition ➔ tolerable range of execution times
- Analytical assurance on utilizations despite uncertainty

![Graph showing CPU utilization and error bars for actual execution times compared to estimation.](image)

**Predicted bound for stability**
FC-ORB Middleware

Workload Uncertainty

disturbance from periodic tasks

time-varying execution times
Processor Failure

1. Norbert fails.
2. move its tasks to other processors.
3. reconfigure controller
4. control utilization by adjusting task rates
Summary: Model Predictive Control

- Application to CPU utilization control
  - Robust utilization control for distributed systems
  - Handle coupling among processors
  - Enforce constraints on task rates
  - Analyze tolerable range of execution times

- Applicable to many computing problems
  - MIMO: multi-input (knobs), multi-output (objectives)
  - Coupling between objectives
  - Constraints on knobs
Outline

- CPU Utilization Control for Distributed Real-Time Systems
  - Model Predictive Control

- Thermal Control for Real-Time Systems
  - Nested Control Design
Nested Control

- Multiple control objectives
  - Coupling between objectives
  - Dynamics at different time scales

- Approach: Nested feedback control loop
Thermal Control for Real-Time Systems

- Temperature control
  - Prevent processor overheating
  - Avoid hardware throttling \(\rightarrow\) unpredictable slowdown

- Utilization control
  - Maintain real-time performance
  - Enforce schedulable utilization bound

- Uncertainties
  - Power, ambient temperature, thermal faults, execution time
Goals

- Enforce both thermal and real-time constraints
  - Temperature bound $T_b <$ hardware throttling threshold
  - CPU utilization bound $U_b <$ schedulable utilization bound

- Robust against uncertainties

- Run-time efficiency
Control-theoretic Approach

- Deal with uncertainties through feedback control
  - Rate adaptation based on temperature and utilization feedback

- Nested control structure
  - Modular: separate controllers for temperature and utilization
  - Efficiency control algorithms: $O(1)$ complexity
  - Rigorous stability and sensitivity analysis

Dynamic System Model

Temperature (controlled variable)

\[ \frac{dT(t)}{dt} = -c_2(T(t) - T_0) + c_1 P(t) \]

Power

\[ 
\bar{P}(k) = G_p P_a U(k) + P_{idle}(1 - U(k)) 
\]

Utilization (control input)

(controlled variable)

Utilization Control

\[ U(k + 1) = U(k) + G_u \sum_i c_i \Delta r_i \]

Tasks rates (control input)

Thermal Control

CPS Week 2013
TCUB: Thermal Control under Utilization Bound

- **Outer loop**: thermal control
  - Handle slower thermal dynamics
- **Inner loop**: CPU utilization control
  - Handle faster load dynamics

![TCUB Diagram]

- $T_b$, $U_b$, $T(k)$, $U_s(k)$, $U(k')$, $U_{ClizaCon}$, $U_{Controller}$, $\Delta r_i(k')$
Varying Power

Active power = 2 x estimate
Varying Execution Times

Execution time = 2 x estimate

TCUB

Thermal Control
Summary: Nested Control

- Example: Thermal control for real-time systems
  - Control both temperature and utilization bounds
  - Robust against uncertainties

- Nested control approach
  - Control variables with dynamics at different time scales
  - Modular design
  - Efficient control algorithm
References


- **Model Predictive Control:** J.M. Maciejowski, Predictive Control with Constraints, Prentice Hall, 2002.

- **Adaptive QoS Control Project:** [http://www.cse.wustl.edu/~lu/control.html](http://www.cse.wustl.edu/~lu/control.html)