



Resource Management and Feedback Control

Jin Heo: TA
Tarek Abdelzaher: Instructor



Recap: OS Goals

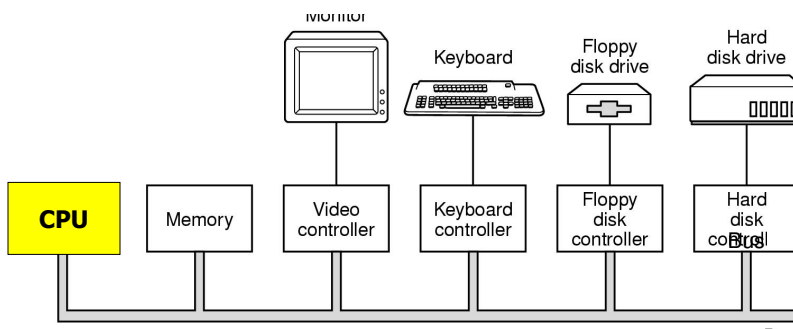
- Provide an interface for applications to do common functions.
- Manage resources

Recap: OS Goals

- Provide an interface for applications to do common functions.
- ***Manage resources***

3

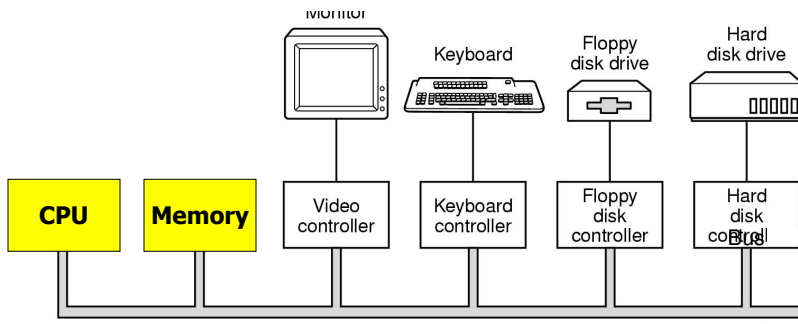
Managed Resources - CPU



Process/thread scheduling, context switching, etc.

4

Managed Resource - Memory



Paging, swapping, virtual memory management, etc.

5

Manage resources

- Mainly best effort services for processes
- How to allocate resources to meet a certain (application-layer) performance goal?
 - Provides the best VOD quality while using less than 50% of CPU
 - Allocate resources to an Apache Web server such that delay = 100ms

6



Which Resource Allocation Schemes?

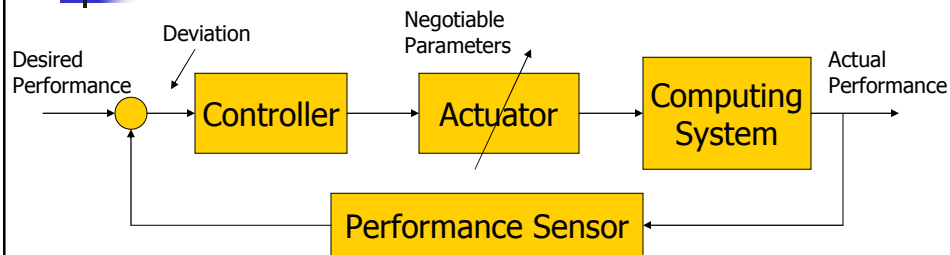
- Operating systems have large uncertainty in workload (e.g., number of processes, resource usage patterns) and resource availability
- Model-based resource allocation is not robust under uncertainty
 - Fine-grained assumptions about process computational requirements
 - Fine-grained assumptions about resource capacity
- Measurement-based heuristics are difficult to analyze
- How to manage performance of an application by properly allocating resources under load and resource uncertainty?



Feedback Control for Resource Management

- Challenges:
 - Make the least/weakest assumptions regarding workload parameters (e.g., arrival times, execution times, deadlines, and number of arrived requests)
 - Control perturbations which may violate these assumptions
- Direction:
 - Move from an "open-loop" (e.g., model-based) to a "closed-loop" resource allocation paradigm
 - Drop the *a priori* assumptions; use continuous feedback for resource allocation
 - Use feedback control theory to analyze closed loop performance

A Control Theory Approach Control Loop Operation:



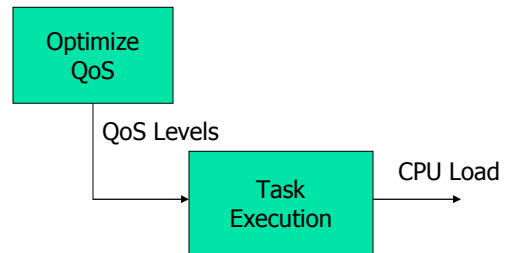
- Model the controlled process by a *differential equation*
- Feedback control is a process of continuous monitoring and correction of deviation from desired performance
- Control theory is involved with controller design

Why Control Theory For Resource Management?

- Performance metrics such as response time and queuing delay depend on (resource) queue state dynamics, capacity and queue length, e.g., of:
 - The CPU ready queue
 - Semaphore queues
 - Socket queues
 - I/O device queues
- A queue acts as a capacitor (a water tank) which integrates the difference between input and output flows, and hence can be modeled by a differential equation → **Perfect for control theory!**

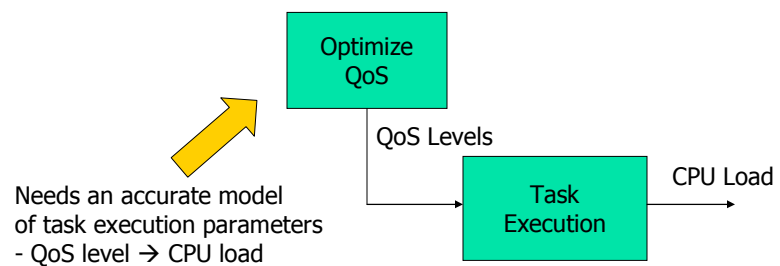
Motivating Example: CPU Allocation for QoS Optimization

- Consider a QoS optimization example



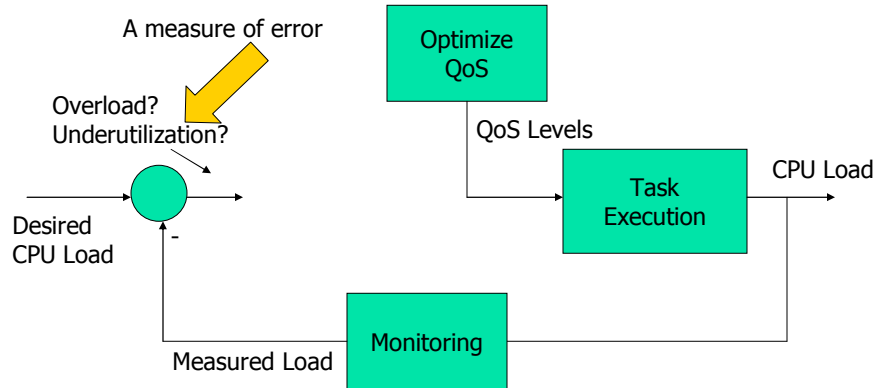
Example

- Optimizer needs a good execution model



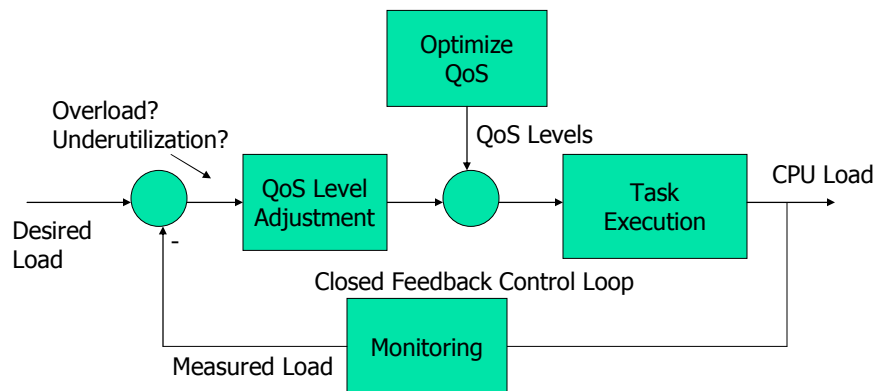
Example

- An error is typically found



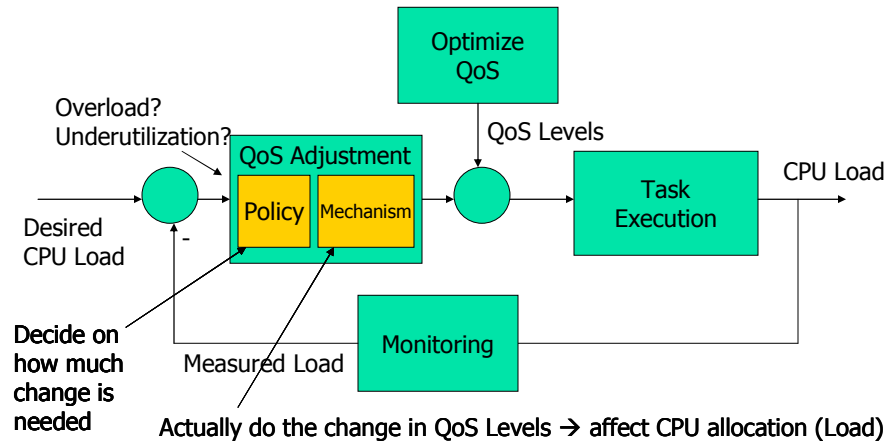
Example

- A controller attempts to reduce error to zero

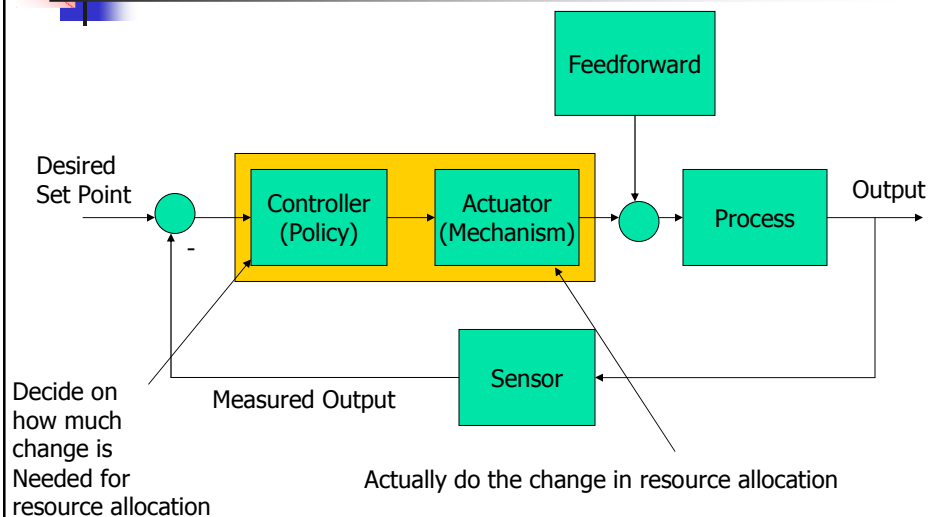


Example

■ A QoS feedforward/feedback loop

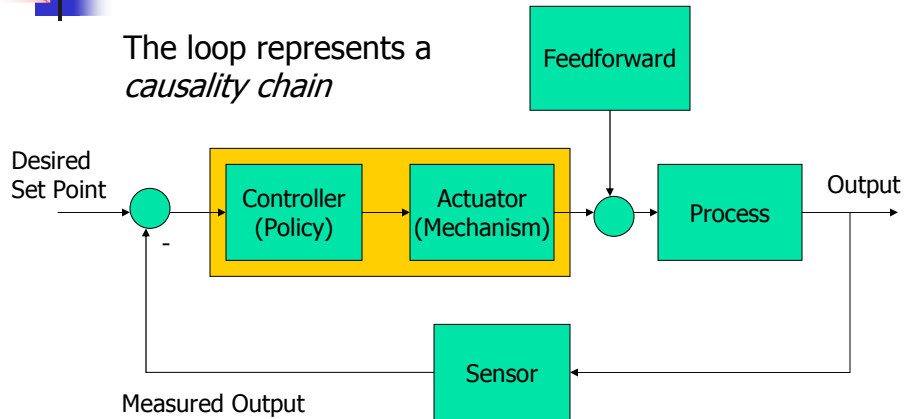


Feedback Control Loops

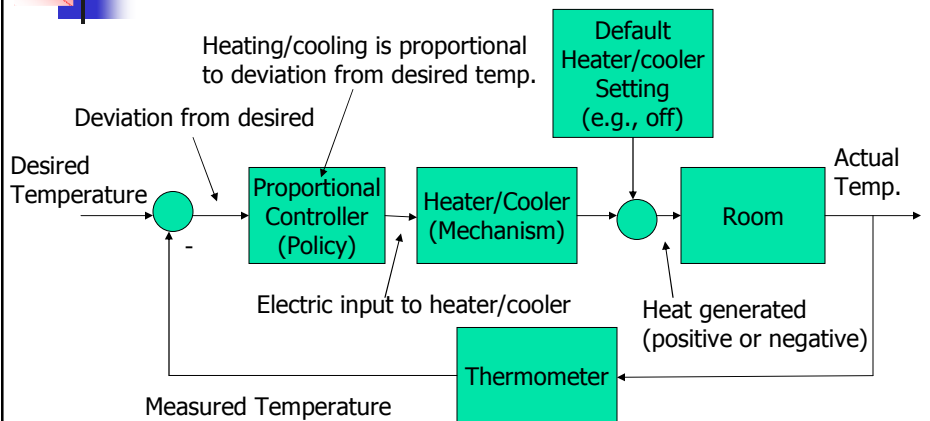


Feedback Control

The loop represents a *causality chain*



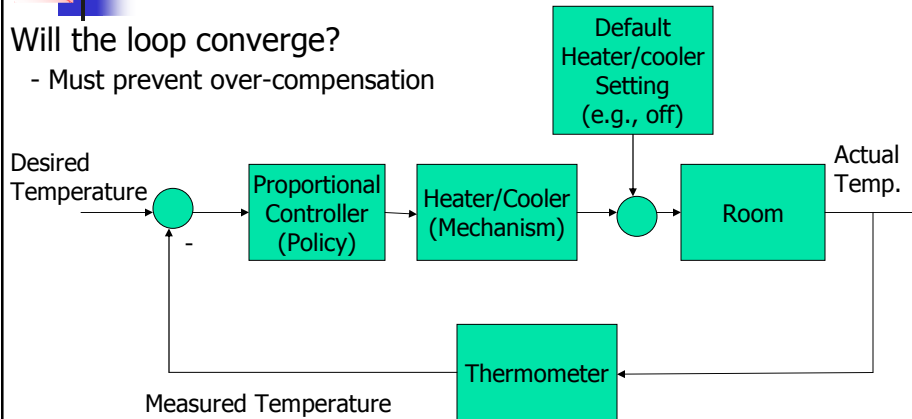
Non-CS Feedback Control Example



Feedback Design Concern #1: Stability

Will the loop converge?

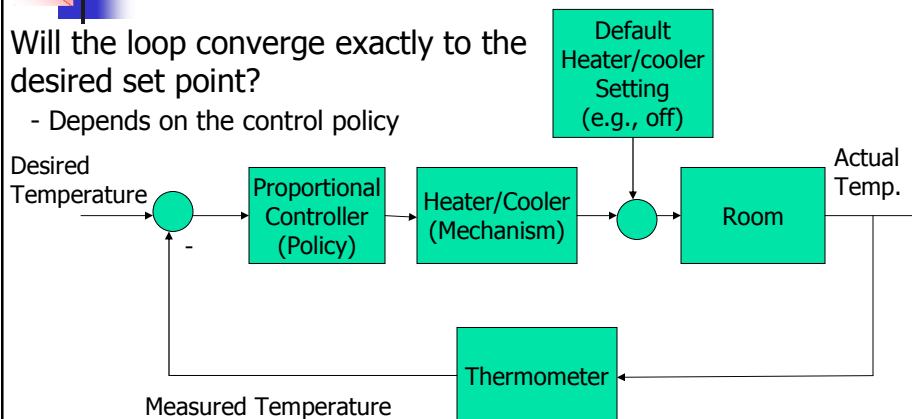
- Must prevent over-compensation



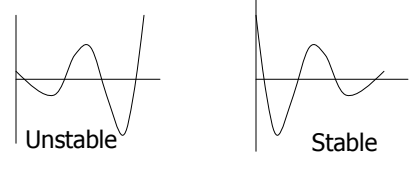
Feedback Design Concern #2: Steady State Error

Will the loop converge exactly to the desired set point?

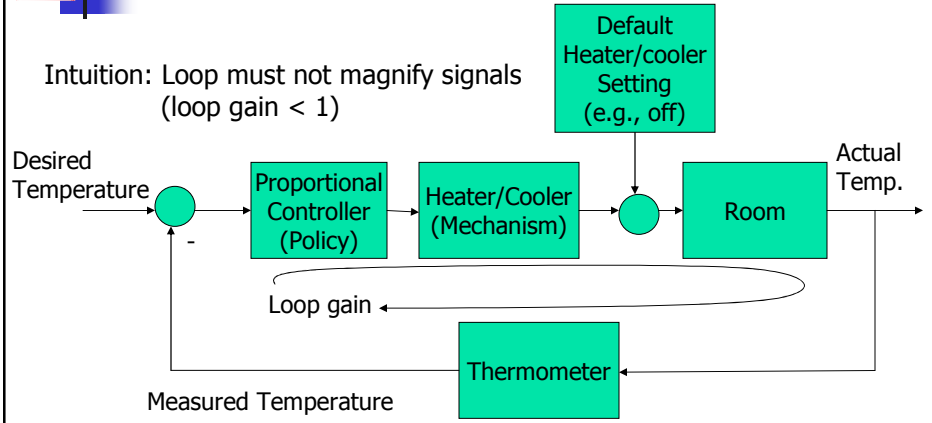
- Depends on the control policy



Stability

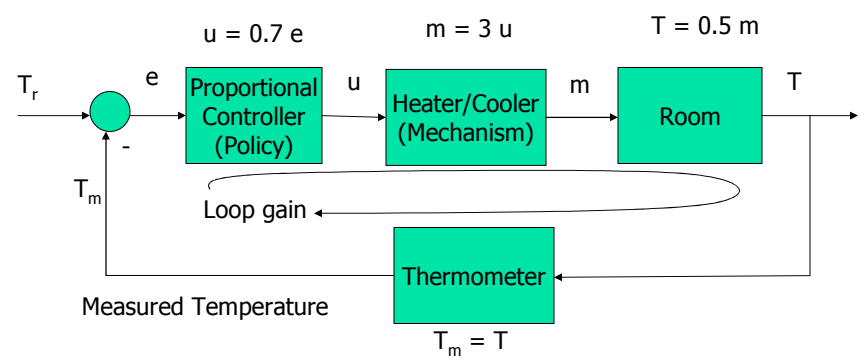


Intuition: Loop must not magnify signals
(loop gain < 1)



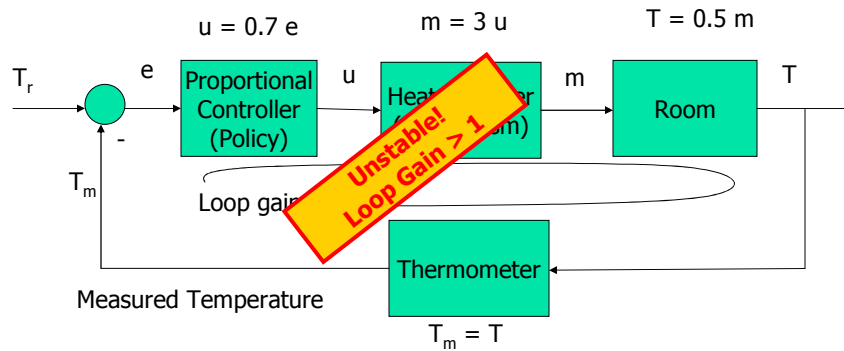
Stability Example

Is the loop below stable?



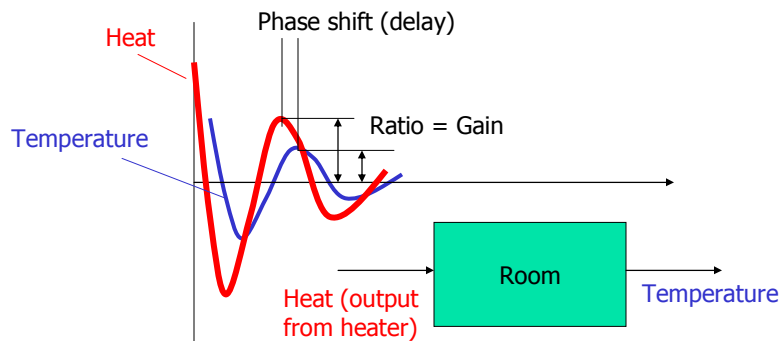
Stability Example

Is the loop below stable?



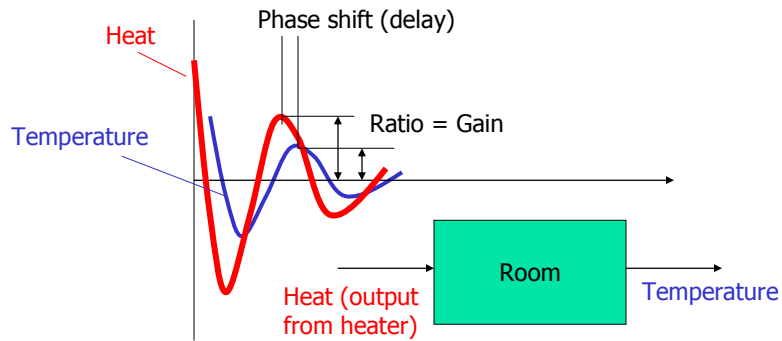
Stability and Phase Shift

- Fact 1: Most reactions are not instantaneous



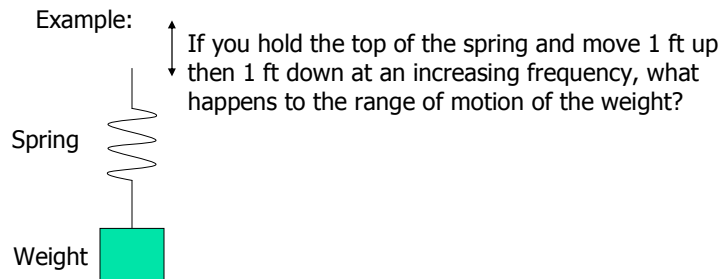
Stability and Phase Shift

- Fact 2: Gain and phase shift depend on frequency



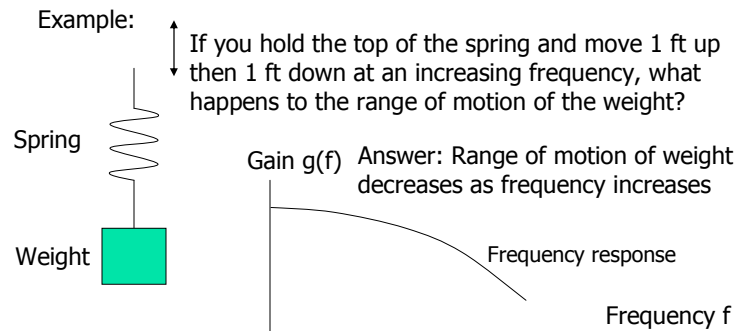
Stability and Phase Shift

- Fact 2: Gain and phase shift depend on frequency



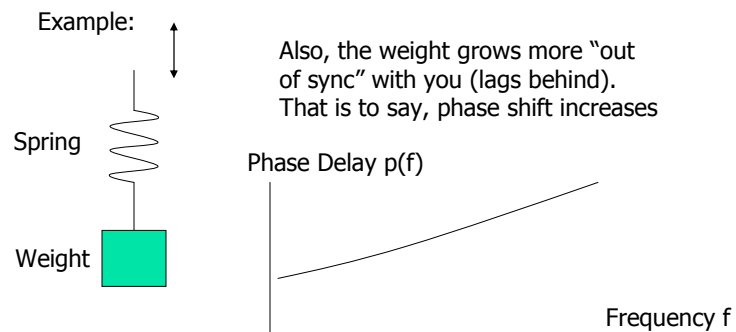
Stability and Phase Shift

- Fact 2: Gain and phase shift depend on frequency

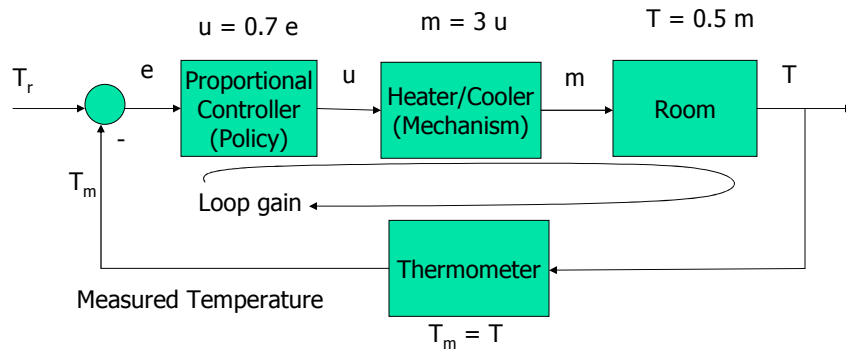


Stability and Phase Shift

- Fact 2: Gain and phase shift depend on frequency

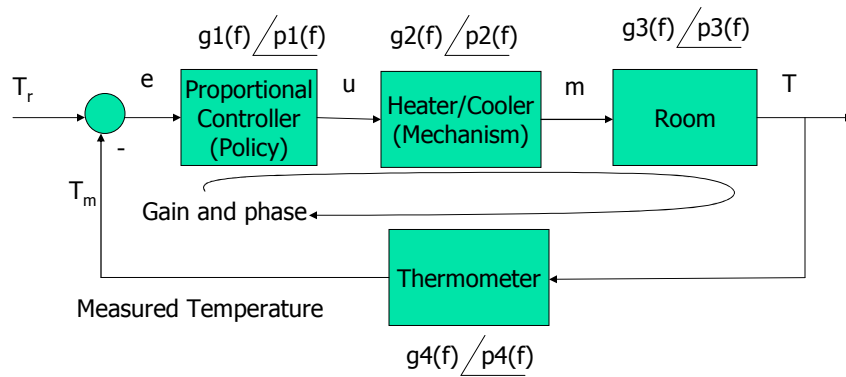


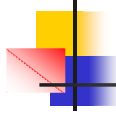
Stability Example Revisited



Stability Example Revisited

Which frequency f to consider?

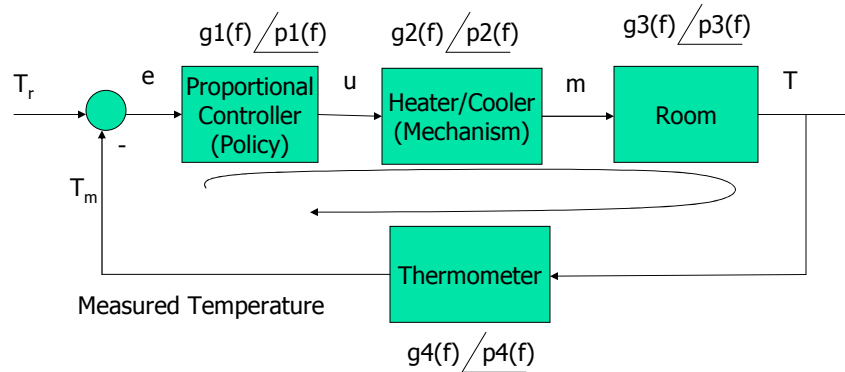




Stability Example Revisited

Which frequency f to consider?

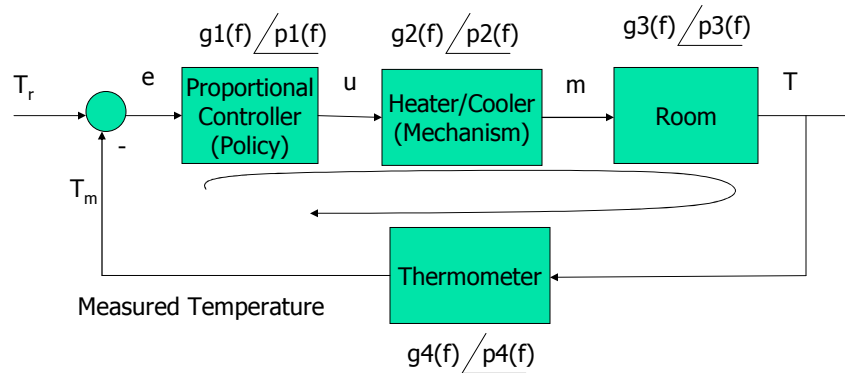
Answer: The one that makes the sum of $p_i(f) = 180^\circ$ (why?)



Stability Example Revisited

Phase equation: $\sum_i p_i(f) = \pi$ \longrightarrow f is obtained

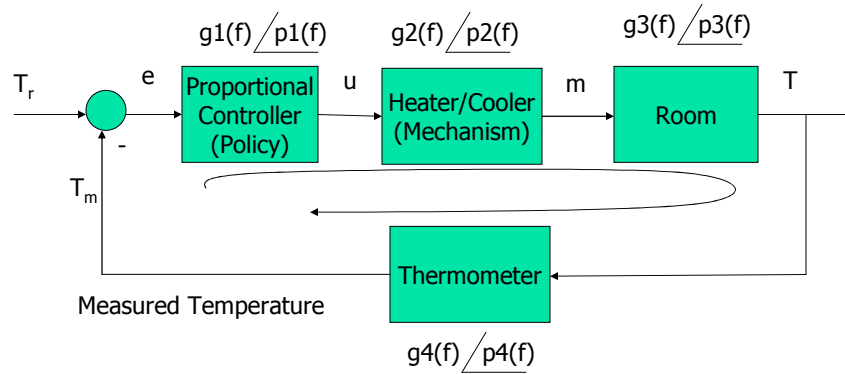
Gain equation: $\prod_i g_i(f)$ must be less than 1 for stability



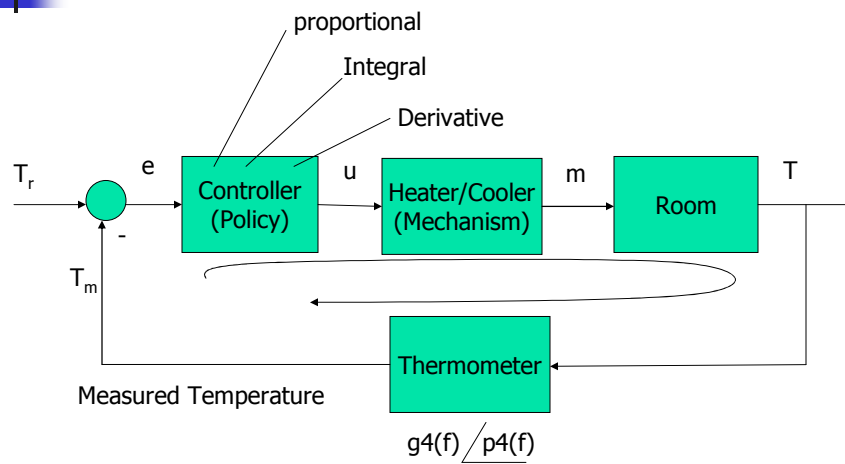
Tuning the Control Loop

Phase equation: $\sum_i p_i(f) = \pi$ \longrightarrow f is obtained

Gain equation: $\prod_i g_i(f) = 0.5$



The Controller





Computing the Transfer Function

- Example 1: Find gain and phase of an integrator?



Computing the Transfer Function

- Example 1: Find gain and phase of an integrator?
- Observation: The integral of $\sin(\omega t)$ is $-\cos(\omega t) / \omega$
 - Gain = $1/\omega$
 - Phase = -90°



Computing the Transfer Function

- Example 2: Find gain and phase of a differentiator?



Computing the Transfer Function

- Example 2: Find gain and phase of a differentiator?
- Observation: The derivative of $\sin(\omega t)$ is $\omega \cos(\omega t)$
 - Gain = ω
 - Phase = 90°



Computing the Transfer Function

- Example 3: Find gain and phase of a pure delay element?



Computing the Transfer Function

- Example 3: Find gain and phase of a pure time-delay element?
- Observation: Delay does not magnify signal. Phase shift is equal to proportional to frequency and delay
 - Gain = 1
 - Phase = ωD

Sampled Queue Model

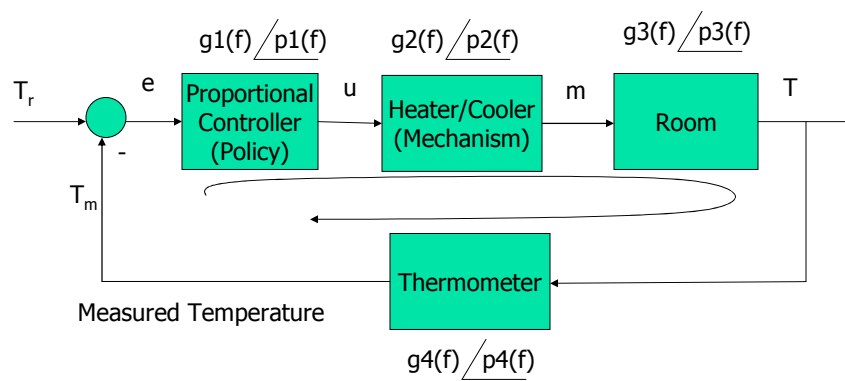
- A queue is an integrator of flow
- Sampling queue state introduces delay equal to half the sampling time
- Assume a proportional controller is used

Phase equation: $\sum_i p_i(f) = \pi$

Gain equation: $\prod_i g_i(f) = 0.5$

Steady State Error

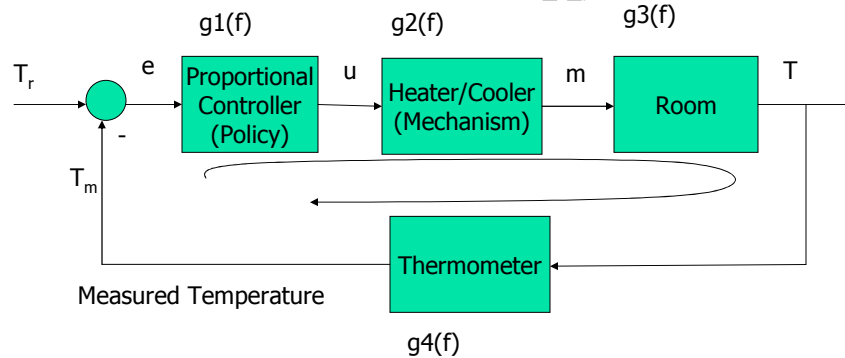
Loop must contain an integrator for the steady state error to be zero.
(Why?)



Steady State Error

At steady state the system "catches up" – phase shift is zero.

$$T_r - e \cdot g_1(f) \cdot g_2(f) \cdot g_3(f) \cdot g_4(f) = e \quad e = \frac{T_r}{1 + \prod_i g_i(f)}$$



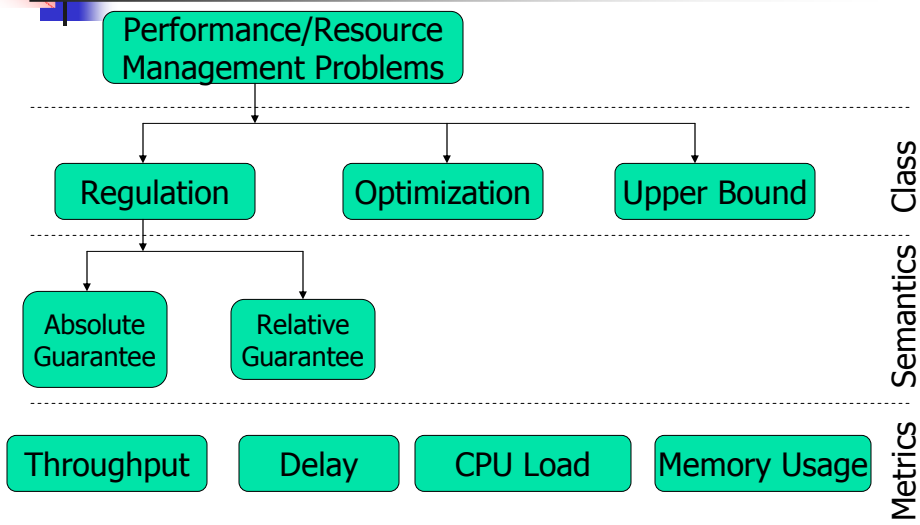
Recap

- We have addressed stability conditions (will the loop converge?)
- We have quantified steady state error (will it converge to the right value?)
- We can now design stable feedback control loops for performance control in computing systems

Performance Management and Resource Management

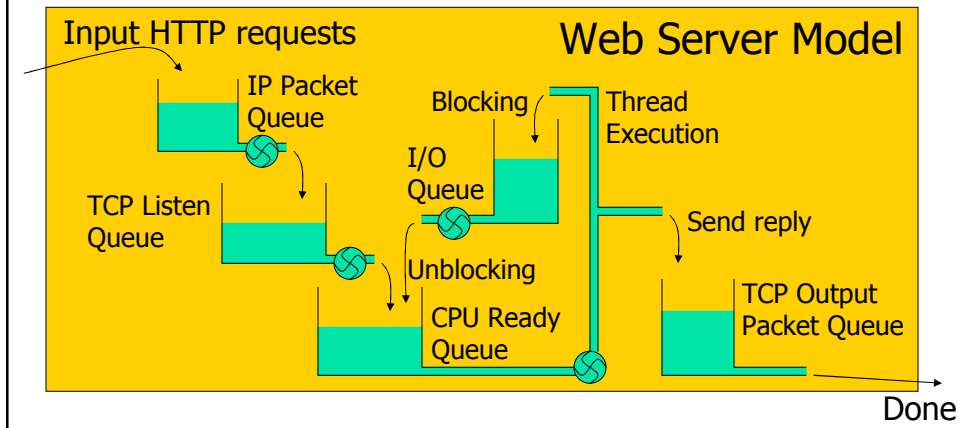
- Performance management and resource management are not two separate problems
 - Changing resource allocation affects performance
 - To change the performance, resource allocation needs to be properly adjusted
- Control theory integrates the two problems into one framework without fine-grained assumptions (or models) on resource allocation and the resulting performance

Performance/Resource Management as a Feedback Problem: A Taxonomy



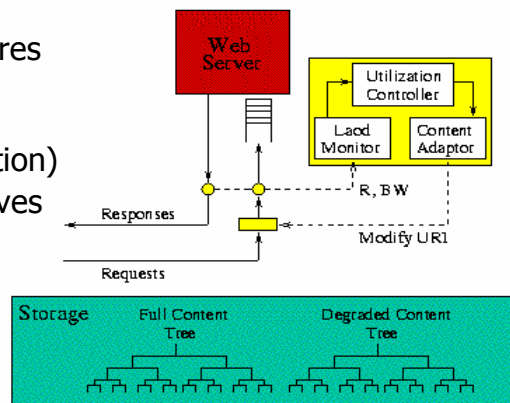
Example: Web Server Modeling

Why web servers can be modeled by difference equations!



Example: Web Server Utilization Control Loop

- Load Monitor (measures utilization)
- Utilization Controller (determines degradation)
- Content Adaptor (serves appropriate content version)

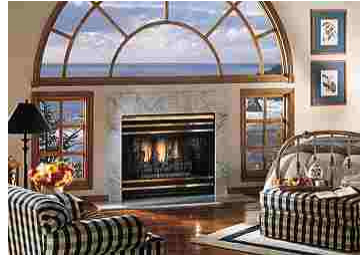


Actuation Example: Image Compression



74KB GIF

High QoS



8.4 KB JPEG

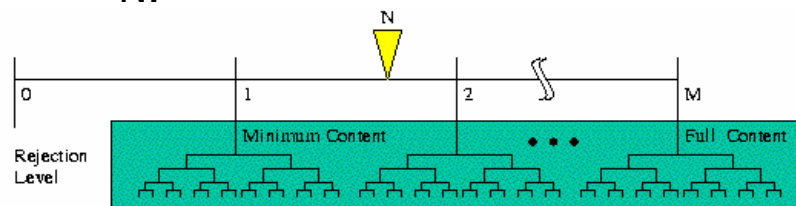
Low QoS

2010-04-19



Content Adaptor

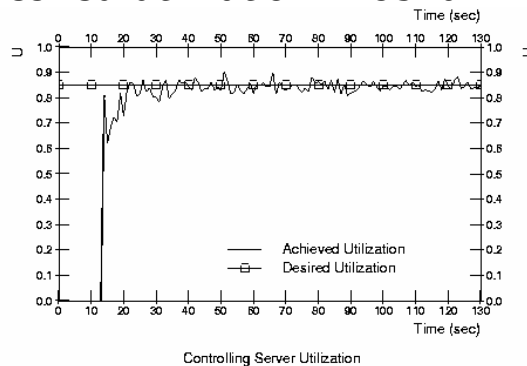
- Degrades or rejects a fraction of requests depending on controller output, N .



Experiment:

Efficacy of Utilization Control

- Offered Load = 300% (at t=13)
- Desired utilization = 85%



Summary of Utilization Control

- Utilization control loops are of great use for controlling throughput-like metrics in web servers
- The problem has a control-theoretic formulation
- The feedback control loop is successful in providing desired performance