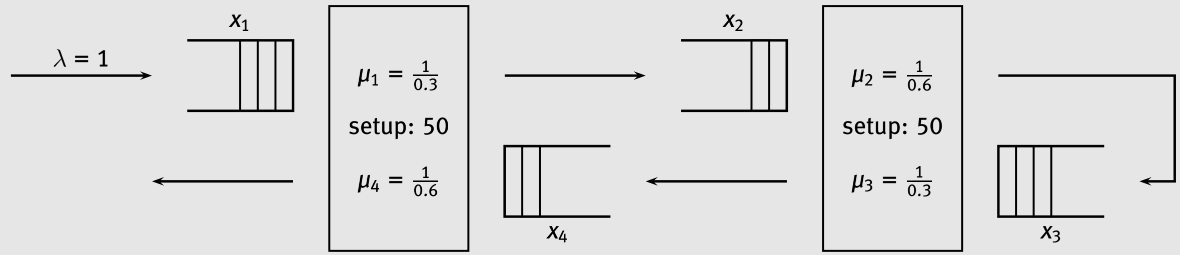
Controller design for switched server networks

Erjen Lefeber

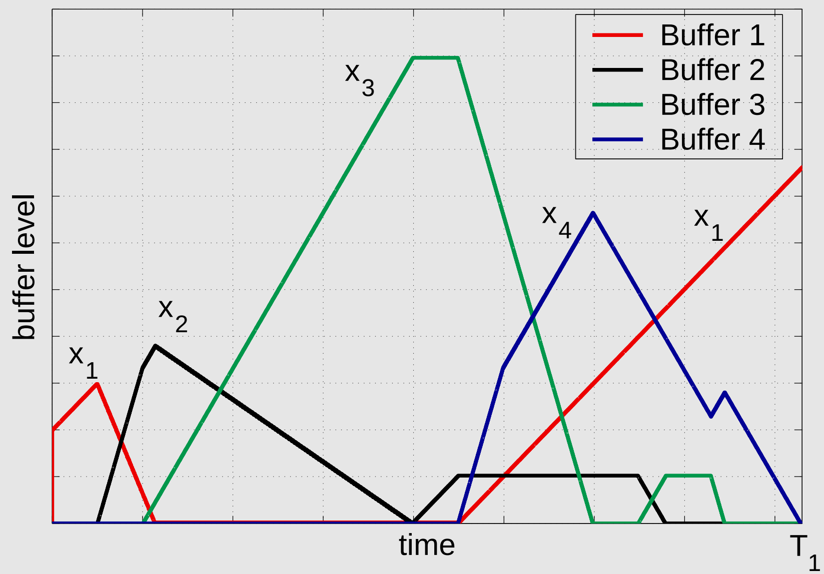
Goldrain, September 18, 2006

Kumar-Seidman, 1990

System (deterministic)



Clearing results in



Observation

- Even though $\rho < 1$: instability

Observation

- Even though $\rho < 1$: instability

Claim

- Due to policy, not to system
- $\rho < 1$ at each server is necessary and sufficient

Observation

- Even though $\rho < 1$: instability

Claim

- Due to policy, not to system
- $\rho < 1$ at each server is necessary and sufficient

Objective

Do not start from policy, then study closed-loop behavior

But start from desired behavior and derive policy

Standard Linear Control Theory

Control theorists study

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R^k \tag{1a}$$

$$y = Cx \qquad y \in R^m \tag{1b}$$

where $u(\cdot)$ is a *function* to be designed.

Standard Linear Control Theory

Control theorists study

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R^k \tag{2a}$$

$$y = Cx \qquad y \in R^m \tag{2b}$$

where $u(\cdot)$ is a *function* to be designed.

Lemma: The system (2) is *controllable* iff

$$\text{rank} \left[B \mid AB \mid A^2B \mid \dots \mid A^{n-1}B \right] = n$$

Lemma: The system (2) is *observable* iff

$$\text{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n$$

Controller design: Static state feedback

Consider the system

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R^k$$

Using the (static state) feedback

$$u = -Kx$$

results in the closed-loop dynamics

$$\dot{x} = (A - BK)x$$

Lemma: If the system is controllable, then the poles of the matrix $A - BK$ can be placed arbitrarily.

Observer design

Consider the system

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R^k$$

$$y = Cx \qquad y \in R^m$$

Consider the observer

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$

$$\hat{y} = C\hat{x}$$

Define the observer-error $e = x - \hat{x}$. Then we get

$$\dot{e} = Ae - LCe = (A - LC)e$$

Lemma: If the system is observable, then the poles of the matrix $A - LC$ can be placed arbitrarily.

Controller design: Dynamic output feedback

Consider the system

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R^k$$

$$y = Cx \qquad y \in R^m$$

where only the output y can be measured.

Controller design: Dynamic output feedback

Consider the system

$$\dot{x} = Ax + Bu \qquad x \in R^n, u \in R^k$$

$$y = Cx \qquad y \in R^m$$

where only the output y can be measured.

Strategy: use dynamic output feedback

$$u = -K\hat{x}$$

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y})$$

$$\hat{y} = C\hat{x}$$

The poles of the closed-loop dynamics are given by the poles of the matrices $A - BK$ and $A - LC$.

Summarizing

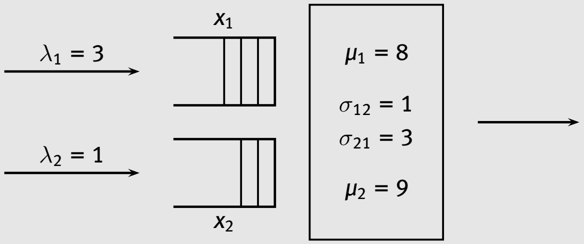
Separate problems:

- Design controller using full state
- Design observer, reconstructing state from measurements
- Design dynamic controller using measurements

Other separate problem:

- Specify desired behavior (trajectory generation)

Single machine: switched linear system



- Constant arrival rates λ_1 and λ_2
- Maximal service rates μ_1 and μ_2
- Setup times σ_{12} and σ_{21}
- Buffer contents x_1 and x_2
- Activities: ➊: setup for Type 1 jobs
  ①: serve Type 1 jobs
  ➋: setup for Type 2 jobs
  ②: serve Type 2 jobs

Goal

Design a policy which controls this system towards optimal behavior.

Two (separate) sub-problems

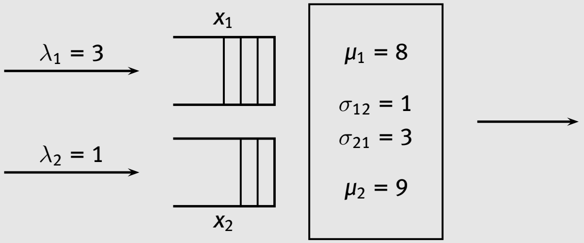
- Determine optimal behavior
- Derive policy

Problem 1: Optimal behavior

Determine "optimal periodic orbit"

$$\frac{1}{T}\int_0^T c_1 x_1(t) + c_2 x_2(t)\,\mathrm{d}t$$

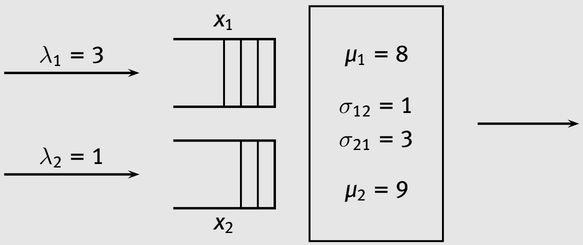
where $c_1 = c_2 = 1$



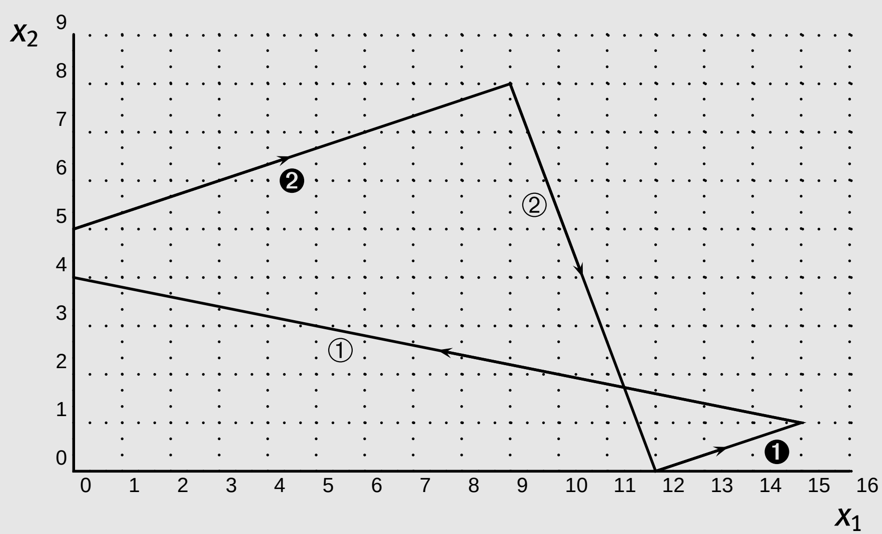
Problem 1: Optimal behavior

Determine “optimal periodic orbit”

$$\frac{1}{T}\int_0^T c_1 x_1(t) + c_2 x_2(t)\,\mathrm{d}t$$



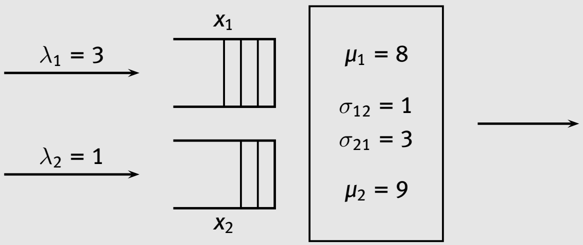
Solution



Problem 1: Optimal behavior

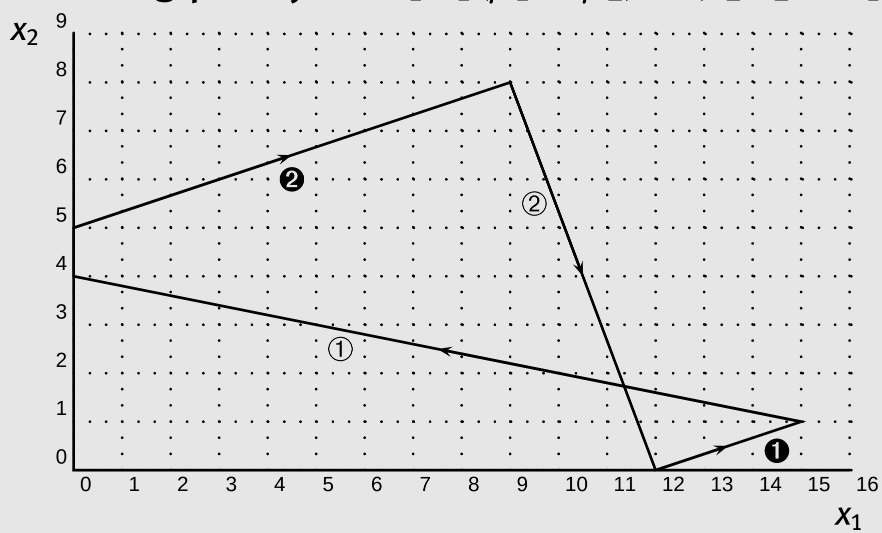
Determine “optimal periodic orbit”

$$\frac{1}{T}\int_0^T c_1 x_1(t) + c_2 x_2(t)\,\mathrm{d}t$$



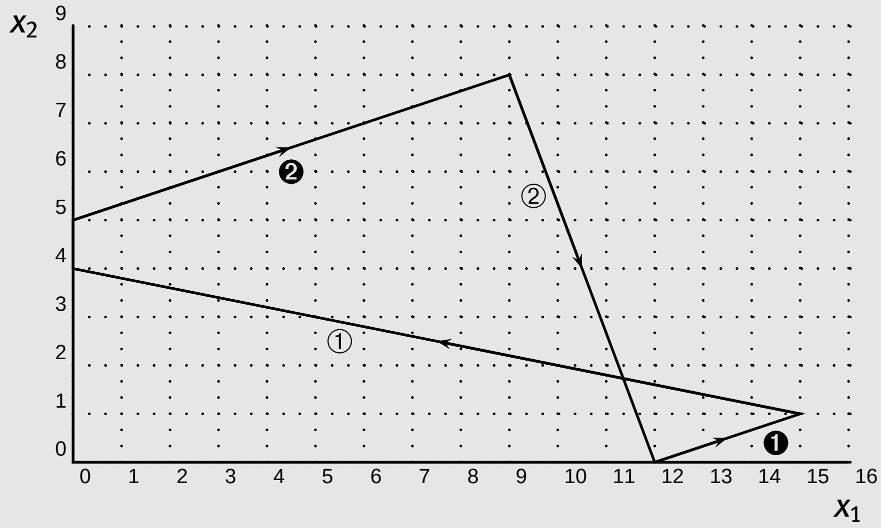
Solution

- Idling policy $\quad c_1\lambda_1(\rho_1 + \rho_2) + (c_2\lambda_2 - c_1\lambda_1)(1 - \rho_2) < 0$



Problem 2: Design policy

Design feedback which make system converge towards “optimal periodic orbit”



Controller design: main idea

Lyapunov: If energy is decreasing all the time $\Rightarrow$ system should settle down at constant energy level

Controller design: main idea

Lyapunov: If energy is decreasing all the time $\Rightarrow$ system should settle down at constant energy level

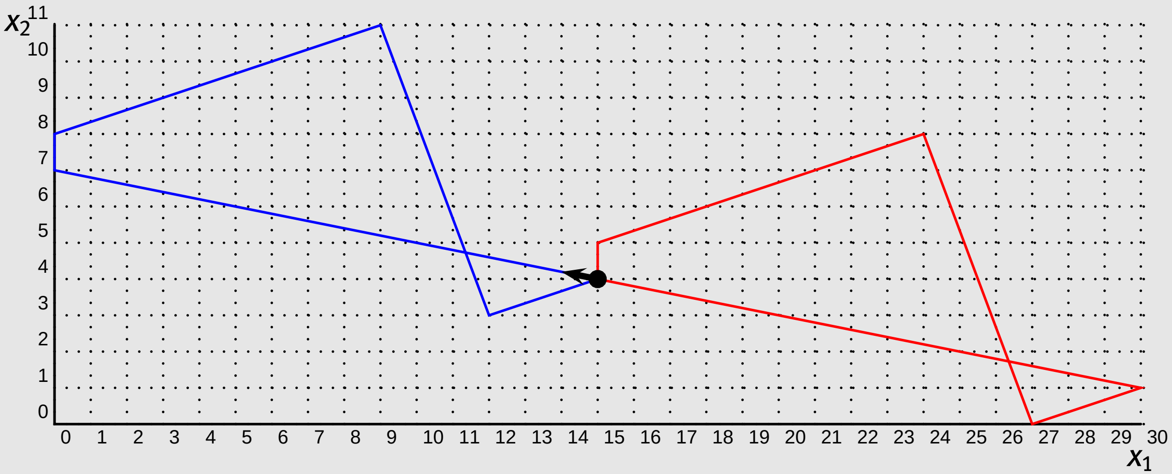
Amount of work in system: $\frac{x_1}{\mu_1} + \frac{x_2}{\mu_2}$

Associate with periodic orbit: mean amount of work.

Lyapunov function candidate

For given point X

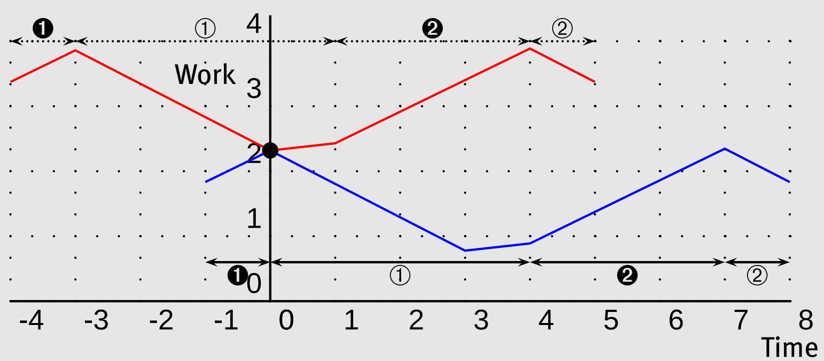
- Of all curves going through X, take the one with minimal mean amount of work



Lyapunov function candidate

For given point X

- Of all curves going through X, take the one with minimal mean amount of work



Lyapunov function candidate

For given point X

- Of all curves going through X, take the one with minimal mean amount of work
- Subtract from this amount of work, that of the desired periodic orbit
- This number is the value $V(X)$ of the Lyapunov function candidate in X

Controller design

Of all possible control-actions, take the one which makes $V(X)$ decrease the most.

Resulting controller

- If in Mode 1: Stay in this mode until both $x_1 = 0$ and $x_2 \geq 5$. Then switch to Mode 2.
- If in Mode 2: Stay in this mode until both $x_2 = 0$ and $x_1 \geq 12$. Then switch to Mode 1.

Resulting controller

- If in Mode 1: Stay in this mode until both $x_1 = 0$ and $x_2 \geq 5$. Then switch to Mode 2.
- If in Mode 2: Stay in this mode until both $x_2 = 0$ and $x_1 \geq 12$. Then switch to Mode 1.

Finite buffers

If $x_1 \leq x_1^{\max}$ and $x_2 \leq x_2^{\max}$ with $x_1^{\max} \geq 15$ and $x_2^{\max} \geq 8$:

Resulting controller

- If in Mode 1: Stay in this mode until both $x_1 = 0$ and $x_2 \geq 5$. Then switch to Mode 2.
- If in Mode 2: Stay in this mode until both $x_2 = 0$ and $x_1 \geq 12$. Then switch to Mode 1.

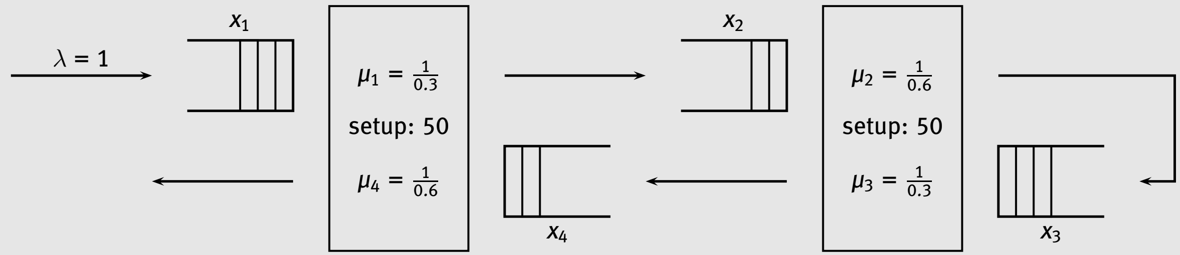
Finite buffers

If $x_1 \leq x_1^{\max}$ and $x_2 \leq x_2^{\max}$ with $x_1^{\max} \geq 15$ and $x_2^{\max} \geq 8$:

- If in Mode 1: Stay in this mode until either both $x_1 = 0$ and $x_2 \geq 5$ or $x_2 = x_2^{\max} - 3$. Then switch to Mode 2.
- If in Mode 2: Stay in this mode until either both $x_2 = 0$ and $x_1 \geq 12$ or $x_1 = x_1^{\max} - 3$. Then switch to Mode 1.

Kumar-Seidman

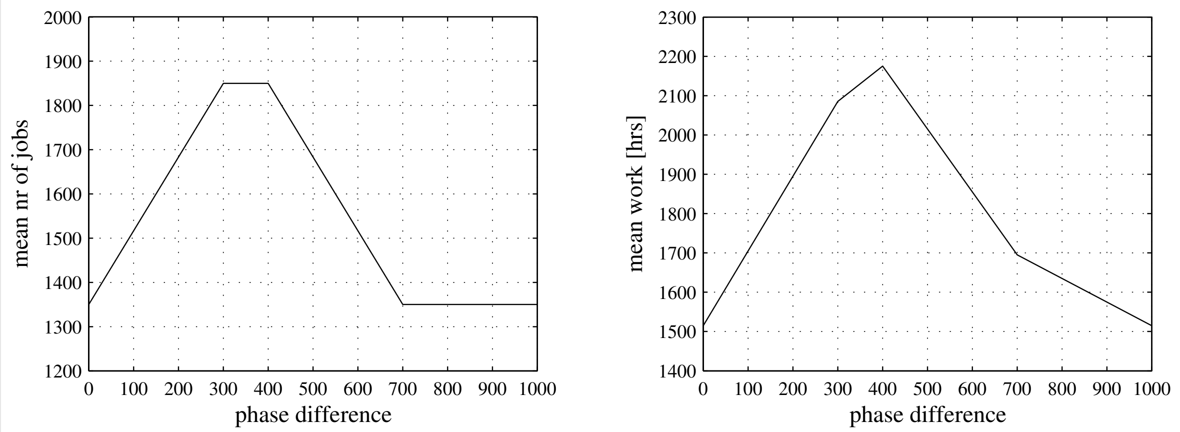
System



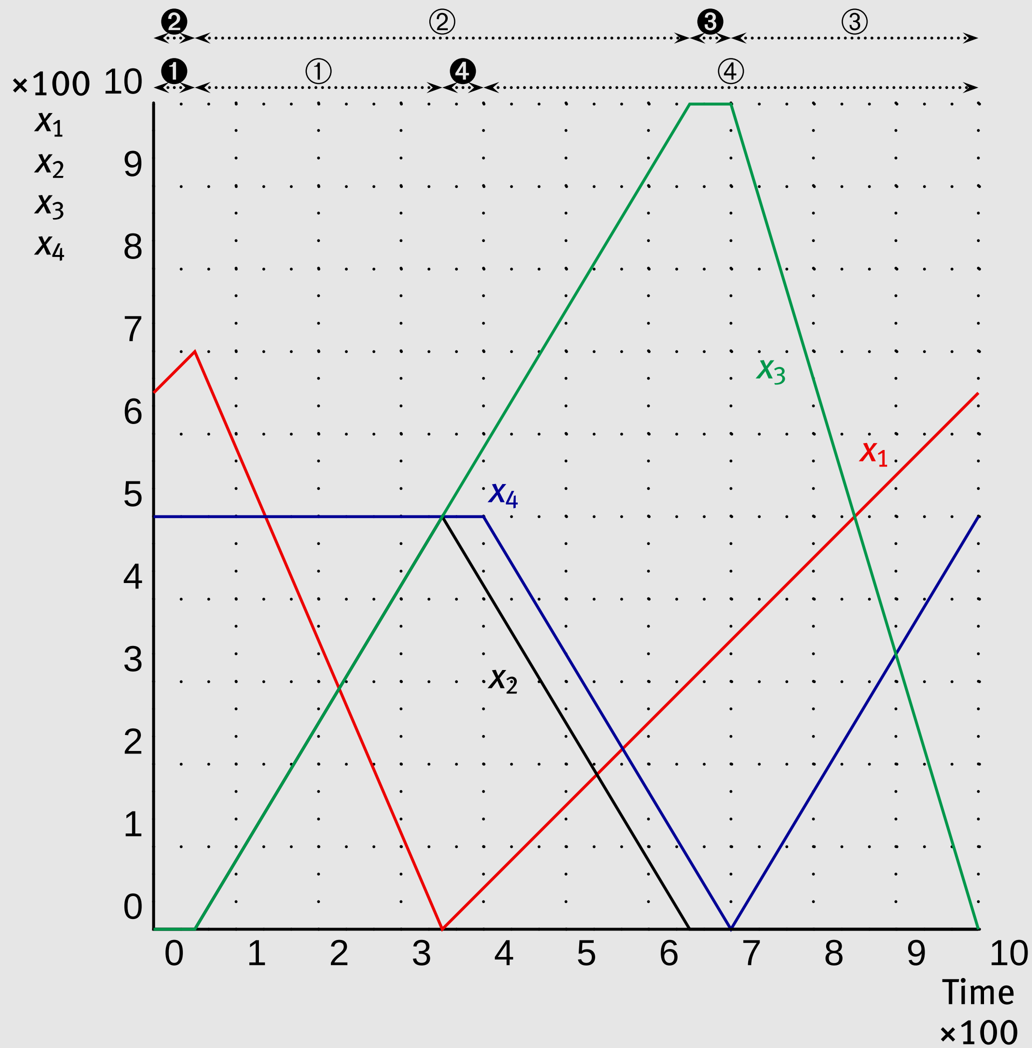
Two (separate) problems

- Desired behavior
- Derive policy

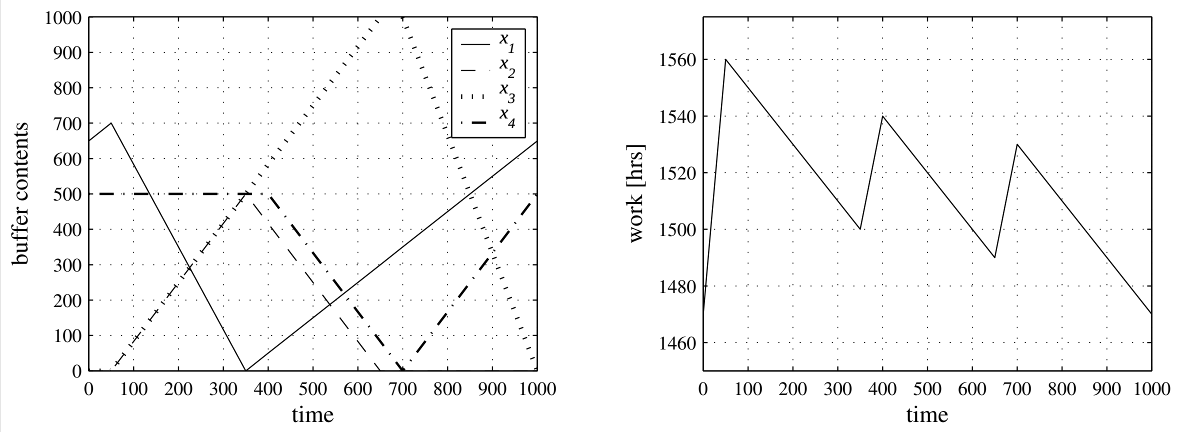
Desired periodic orbit



Desired periodic orbit



Desired periodic orbit



Observations

Largest amount of work at start of Mode (1, 2)

Mode (1, 2) Only x_1 and x_2 can decrease.

At end of mode: $x_1 = 0$, $x_2 = 500$.

x_2 and x_3 can increase. At end: $x_2 = 500$, $x_3 = 500$.

Mode (4, 2) Only x_2 and x_4 can decrease.

At the end of mode: $x_2 = 0$ and $x_4 = 83\frac{1}{3}$.

x_1 and x_3 can increase. At end: $x_1 = 300$, $x_3 = 1000$.

Mode (4, 3) Only x_3 and x_4 can decrease.

At end of the mode: $x_3 = 0$ and $x_4 = 500$.

x_1 and x_4 can increase. At end: $x_1 = 650$, $x_4 = 500$.

Resulting controller

- If initially in Mode $(1,3)$, go to Mode $(1,2)$.
- If in Mode $(1,2)$, serve both types at maximal rate until

  $x_1 = 0$, $x_2 = 500$, $x_3 \geq 500$.

  Then go to Mode $(4,2)$.
- If in Mode $(4,2)$, serve both types at maximal rate until

  either $x_2 = 0$ or $x_4 \leq 83\frac{1}{3}$, $x_1 \geq 300$, $x_3 \geq 1000$.

  Then go to Mode $(4,3)$.
- If in Mode $(4,3)$, serve both types at maximal rate until

  $x_3 = 0$, $x_1 \geq 650$, $x_4 \geq 500$.

  Then go to Mode $(1,2)$.

Resulting controller

- If initially in Mode $(1,3)$, go to Mode $(1,2)$.
- If in Mode $(1,2)$, serve both types at maximal rate until

  $x_1 = 0$, $x_2 = 500$, $x_3 \geq 500$.

  Then go to Mode $(4,2)$.
- If in Mode $(4,2)$, serve both types at maximal rate until

  either $x_2 = 0$ or $x_4 \leq 83\frac{1}{3}$, $x_1 \geq 300$, $x_3 \geq 1000$.

  Then go to Mode $(4,3)$.
- If in Mode $(4,3)$, serve both types at maximal rate until

  $x_3 = 0$, $x_1 \geq 650$, $x_4 \geq 500$.

  Then go to Mode $(1,2)$.

Even though $\dot{V} \leq 0$ we do not have $\lim_{t\to\infty} V(t) = 0$.

Resulting controller

- If initially in Mode $(1,3)$, go to Mode $(1,2)$.
- If in Mode $(1,2)$, serve both types at maximal rate until

  $x_1 = 0$, $x_2 = 500$, $x_3 \geq 500$.

  Then go to Mode $(4,2)$.
- If in Mode $(4,2)$, serve both types at maximal rate until

  both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$, $x_1 \geq 300$, $x_3 \geq 1000$.

  Then go to Mode $(4,3)$.
- If in Mode $(4,3)$, serve both types at maximal rate until

  $x_3 = 0$, $x_1 \geq 650$, $x_4 \geq 500$.

  Then go to Mode $(1,2)$.

Now have $\lim_{t\to\infty} V(t) = 0$ (but not always $\dot{V} \leq 0$).

Proof

Buffer contents when M1 starts processing Type 1 for k^{th} time: $(x_1^k, x_2^k, x_3^k, x_4^k)$

Then for $k > 2$:

$$x_1^{k+1} = 100 + \frac{3}{7}x_1^k + \max(\frac{3}{7}x_1^k, \frac{3}{5}x_4^k)$$

$$x_2^{k+1} = 0$$

$$x_3^{k+1} = 0$$

$$x_4^{k+1} = \max(500, \frac{5}{7}x_1^k)$$

Since

$$x_1^{k+1} \leq 100 + \frac{3}{7}x_1^k + \frac{3}{7}\max(x_1^k, x_1^{k-1})$$

Contraction with fixed point (700, 0, 0, 500).

Global policy

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$. If $x_4 \leq 83\frac{1}{3}$ then possible idling to guarantee $x_2 = 0$, next continu at full rate again.

Machine 2

- If serving 2: continue until both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$. If $x_2 < 500$ and $x_1 = 0$ then possible idling to guarantee $x_2 = 500$ and $x_3 > 0$, next continue at full rate again.
- If serving 3: continue until $x_3 = 0$.

Global policy

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$. If $x_4 \leq 83\frac{1}{3}$ then possible idling to guarantee $x_2 = 0$, next continu at full rate again.

Machine 2

- If serving 2: continue until both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$. If $x_2 < 500$ and $x_1 = 0$ then possible idling to guarantee $x_2 = 500$ and $x_3 > 0$, next continue at full rate again.
- If serving 3: continue until $x_3 = 0$.

Modified policy (1)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$.

Machine 2

- If serving 2: continue until both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$.
- If serving 3: continue until $x_3 = 0$.

Modified policy (1)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$.

Machine 2

- If serving 2: continue until both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$.
- If serving 3: continue until $x_3 = 0$.

Modified policy (1)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$.

Machine 2

- If serving 2: continue until both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$.
- If serving 3: continue until $x_3 = 0$.

In case $x_4 > 83\frac{1}{3}$ and $x_2 = 0$: Idling of Machine 2. Can be shifted.

Improved policy (2)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$. Switch (possible) idling duration of Machine 2 later.

Machine 2

- If serving 2: continue until $x_2 = 0$.
- If serving 3: continue until $x_3 = 0$.

Improved policy (2)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$. Switch (possible) idling duration of Machine 2 later.

Machine 2

- If serving 2: continue until $x_2 = 0$.
- If serving 3: continue until $x_3 = 0$.

Machine 2: Local policy.

Machine 1: Global information needed.

Improved policy (2)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_3 = 0$. Switch (possible) idling duration of Machine 2 later.

Machine 2

- If serving 2: continue until $x_2 = 0$.
- If serving 3: continue until $x_3 = 0$.

Machine 2: Local policy.

Machine 1: Global information needed. Or not?

Improved policy (2)

Machine 1

- If serving 1: continue until both $x_1 = 0$ and 1000 jobs served.
- If serving 4: continue until $x_4 = \max(500, \frac{5}{7} x_1^{\text{prev}})$

where x_1^{prev} = duration of previous serving period + 100

Machine 2

- If serving 2: continue until $x_2 = 0$.
- If serving 3: continue until $x_3 = 0$.

Machine 2: Local policy.

Machine 1: Local policy.

Decentralized Policy

Machine 1:

- Record buffercontents $\bar{x}_1 = x_1$, $\bar{x}_4 = x_4$
- Serve step 1 until both $x_1 = 0$ and at least 1000 jobs processed
- Serve step 4 for $\bar{x}_4 + \frac{1}{2}\bar{x}_1$ jobs

Machine 2:

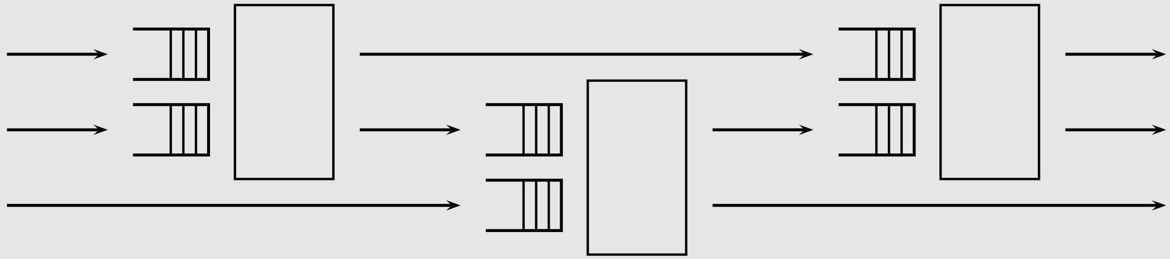
- Clearing

Conclusions

- Given a desired periodic orbit, a policy has been *derived*.
- Buffer constraints can be taken into account easily
- Transportation delays might also be incorporated
- Policy is not necessarily clearing
- Policy is neither gated nor k-limited
- Policy is global (non-autonomous)
- For Kumar-Seidman case: distributed policy (local)

Networks

An acyclic network



A non-acyclic network