NWO
Netherlands Organisation for Scientific Research

Controller design for networks of switching servers with setup times

Erjen Lefeber

Alumni lunchlezing, Universiteit Twente

TU/e Technische Universiteit **Eindhoven** University of Technology

November 23, 2010

Where innovation starts

Wie ben ik

Abacus verleden

- Mathematisch Café
- Ouderdag
- Kaleidoscoopdag
- TWIK’93 (toen: IKTW)
- Bestuur ’93-’94
- Ideaal!
- Coco
- Beleidscommissies
- Kascommissies
- Oud-bestuursledendag

Wie ben ik

Abacus verleden

- Mathematisch Café
- Ouderdag
- Kaleidoscoopdag
- TWIK'93 (toen: IKTW)
- Bestuur '93-'94
- Ideaal!
- Coco
- Beleidscommissies
- Kascommissies
- Oud-bestuursledendag

Ander vrijwilligerswerk

- Vierkant voor Wiskunde
- Onderwijs visitatie commissie
- NOCW (Nederlandse Onderwijscommissie voor de Wiskunde)
- Redactie Pythagoras
- Wiskunde Olympiade
- Kerk

Wie ben ik

Privé

Wie ben ik

Privé

Wat doe ik

Universitair Docent

- Onderwijs
- Onderzoek

Wie ben ik

Privé

Wat doe ik

Universitair Docent

- Onderwijs
- Onderzoek

This work was supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 639.072.072)

Motivation

Problem

Problem

How to control these networks?

Decisions: When to switch, and to which job-type

Goals: Minimal number of jobs, minimal flow time

Problem

Problem

How to control these networks?

Decisions: When to switch, and to which job-type

Goals: Minimal number of jobs, minimal flow time

Current approach

Start from policy, analyze resulting dynamics

Problem

How to control these networks?

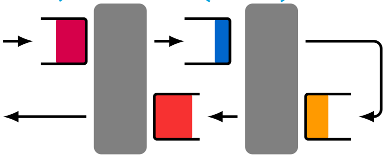
Decisions: When to switch, and to which job-type

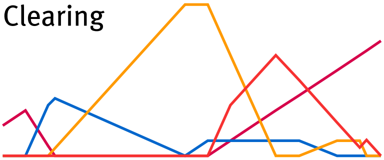
Goals: Minimal number of jobs, minimal flow time

Current approach

Start from policy, analyze resulting dynamics

Kumar, Seidman (1990)





Problem

Current status (after two decades)

Several policies exist that guarantee stability of the network

Problem

Current status (after two decades)

Several policies exist that guarantee stability of the network

Remark

Stability is only a prerequisite for a good policy

Problem

Current status (after two decades)

Several policies exist that guarantee stability of the network

Remark

Stability is only a prerequisite for a good policy

Open issues

- Do existing policies yield satisfactory network performance?
- How to obtain pre-specified network behavior?

Problem

Current status (after two decades)

Several policies exist that guarantee stability of the network

Remark

Stability is only a prerequisite for a good policy

Open issues

- Do existing policies yield satisfactory network performance?
- How to obtain pre-specified network behavior?

Main subject of study (modest)

Fixed, deterministic flow networks (not evolving, constant inflow)

Approach

Approach

Use ideas/concepts from control theory

Technische Universiteit
Eindhoven
University of Technology

Approach

Notions from control theory

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Approach

Notions from control theory

1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Parallels with this problem

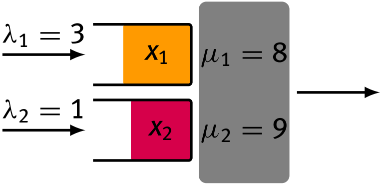
1. Determine desired system behavior
2. Derive non-distributed/centralized controller
3. Can state be reconstructed?
4. Derive distributed/decentralized controller

Technische Universiteit Eindhoven University of Technology

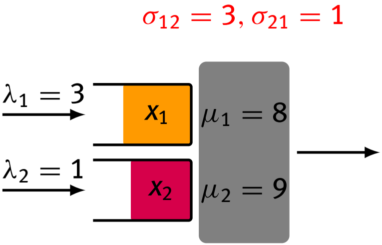
Example 1: Single machine

Single machine

$\sigma_{12} = 3, \sigma_{21} = 1$



Single machine



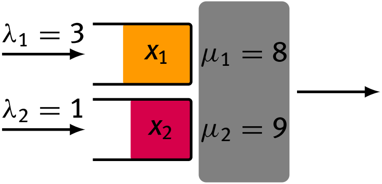
Objective

Minimize:

$$\limsup_{t \to \infty} \frac{1}{t} \int_0^t x_1(\tau) + x_2(\tau) \, \mathrm{d}\,\tau \qquad \text{or} \qquad \frac{1}{T} \int_0^T x_1(\tau) + x_2(\tau) \, \mathrm{d}\,\tau$$

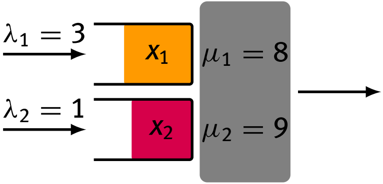
Desired behavior (Problem I)

Single machine

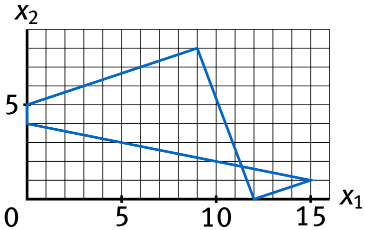


Desired behavior (Problem I)

Single machine

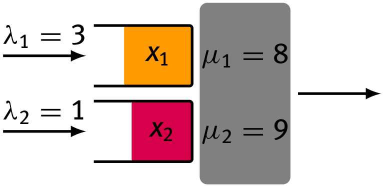


Desired behavior

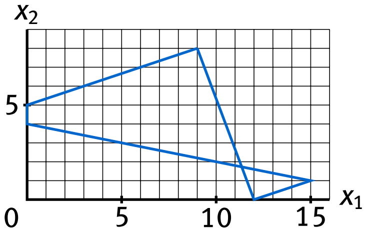


Desired behavior (Problem I)

Single machine



Desired behavior



Remarks

- Many existing policies assume non-idling a-priori
- Slow-mode optimal if $\lambda_1(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2}{\mu_2})-(\lambda_1-\lambda_2)(1-\frac{\lambda_2}{\mu_2})<0.$
- Trade-off in wasting capacity: idle $\Leftrightarrow$ switch more often

TU/e Technische Universiteit Eindhoven University of Technology

Controller design (Problem II)

Main idea

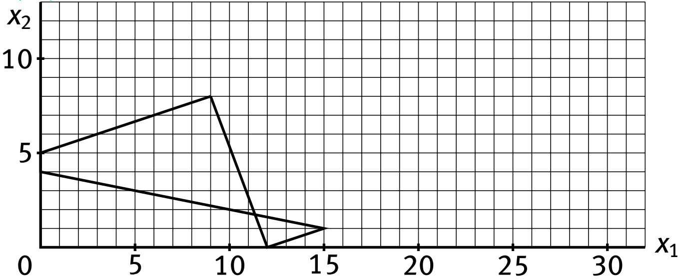
Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate

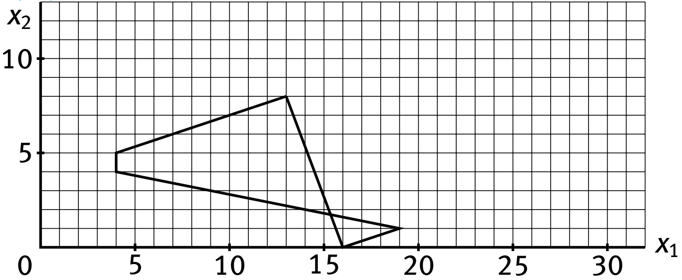


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate

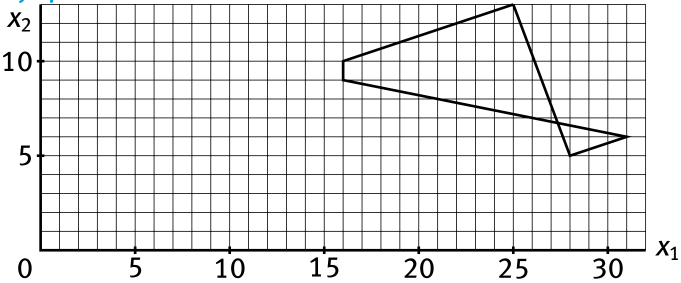


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate

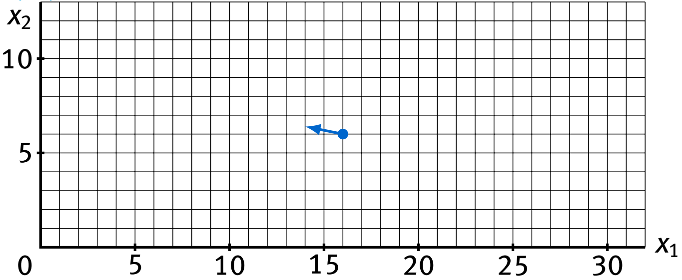


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

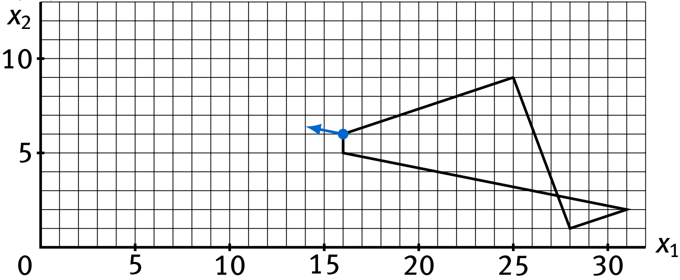
Lyapunov function candidate



Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate

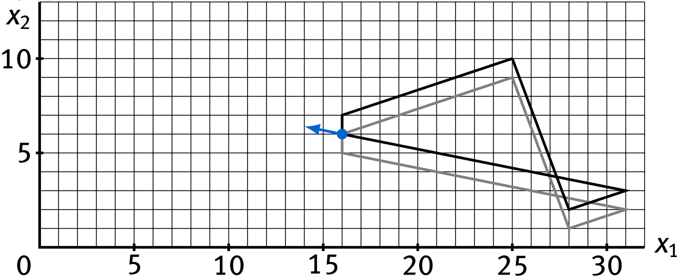


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate

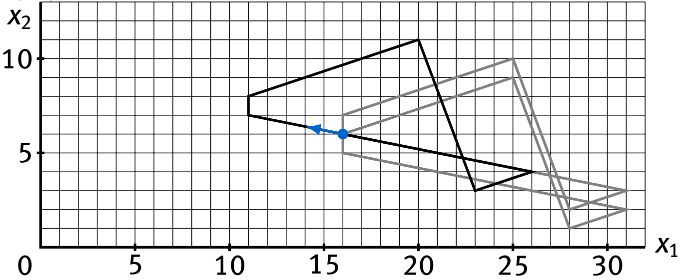


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

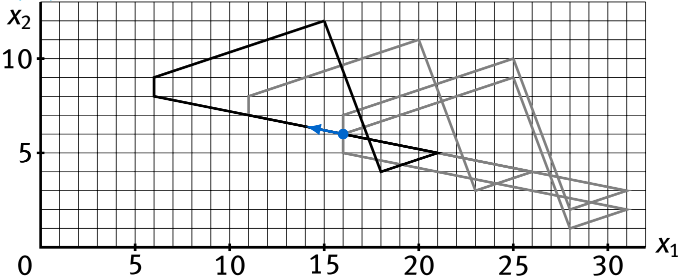
Lyapunov function candidate



Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate



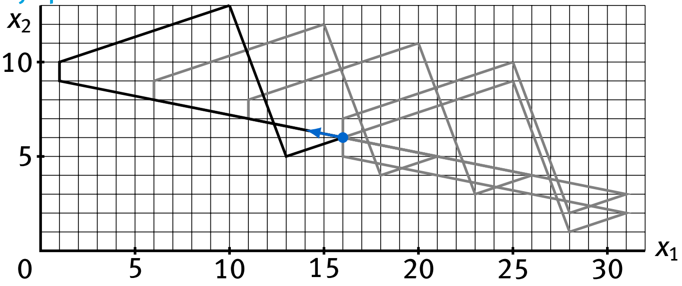
TU/e Technische Universiteit Eindhoven University of Technology

Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time $\Rightarrow$ system settles down at constant energy level

Lyapunov function candidate

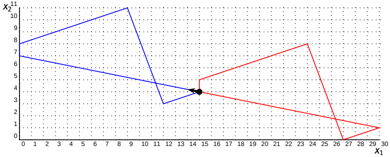


Controller design

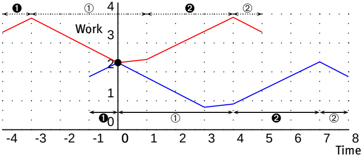
Lyapunov function candidate

The smallest additional mean amount of work from all feasible curves for state (work: $x_1/\mu_1 + x_2/\mu_2$).

Phase plane



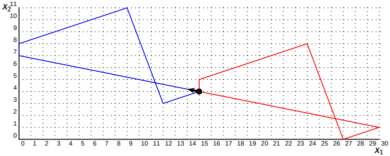
Time evolution work



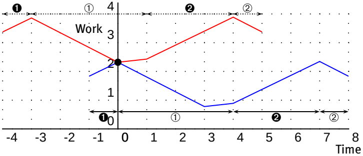
Lyapunov function candidate

The smallest additional mean amount of work from all feasible curves for state (work: $x_1/\mu_1 + x_2/\mu_2$).

Phase plane



Time evolution work



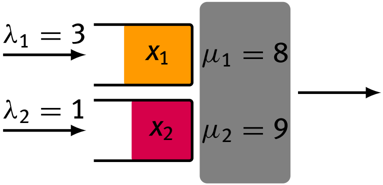
Controller design

Let Lyapunov function candidate decrease as quickly as possible

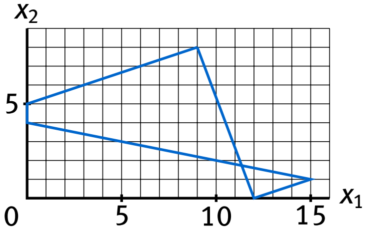
Technische Universiteit Eindhoven University of Technology

Controller design (Result)

Single machine

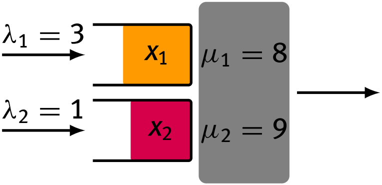


Desired behavior

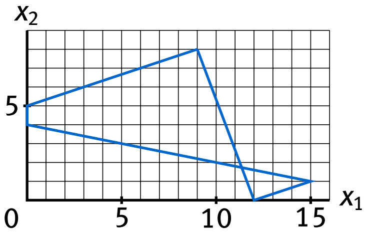


Controller design (Result)

Single machine



Desired behavior

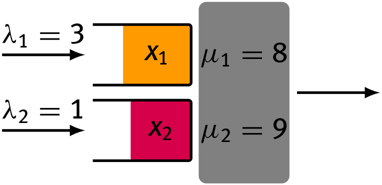


Resulting Controller, cf. [Lefeber, Rooda (2006)]

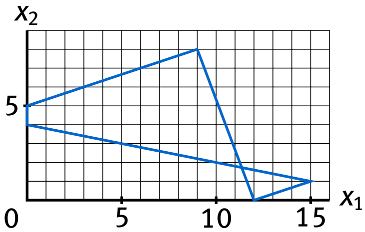
- When serving type 1:
  1. empty buffer
  2. serve until $x_2 \geq 5$
  3. switch to type 2

Controller design (Result)

Single machine



Desired behavior



Resulting Controller, cf. [Lefeber, Rooda (2006)]

- When serving type 1:
  1. empty buffer
  2. serve until $x_2 \geq 5$
  3. switch to type 2
- When serving type 2:
  1. empty buffer
  2. serve until $x_1 \geq 12$
  3. switch to type 1

Recap

Notions from control theory

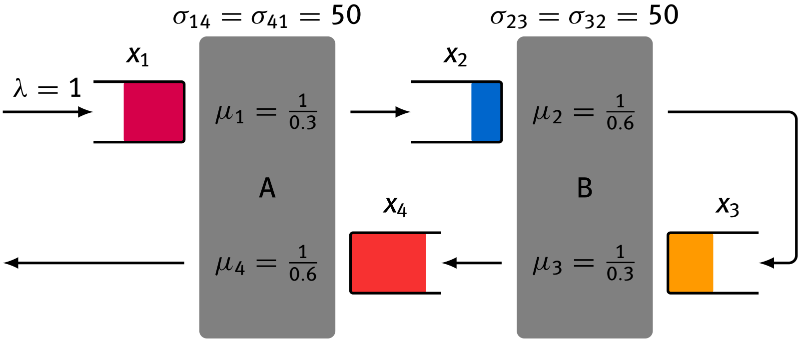
1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Parallels with this problem

1. Determine desired system behavior
2. Derive non-distributed/centralized controller
3. Can state be reconstructed?
4. Derive distributed/decentralized controller

Example 2: Kumar-Seidman case

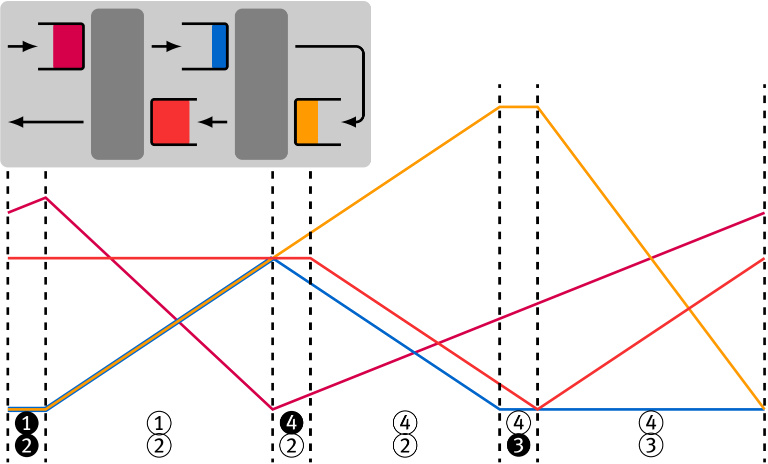
Transactions on Automatic Control, Vol 35, No 3, March 1990



Observation

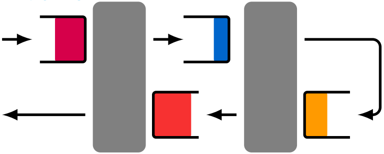
Sufficient capacity (consider period of at least 1000).

Desired behavior



Resulting controller

Network



Desired behavior

Resulting controller

Mode (1,2): to (4,2) when both $x_1 = 0$ and $x_2 + x_3 \geq 1000$

Mode (4,2): to (4,3) when both $x_2 = 0$ and $x_4 \leq 83\frac{1}{3}$

Mode (4,3): to (1,2) when $x_3 = 0$

Remark:

- Non-distributed/centralized controller

Technische Universiteit Eindhoven University of Technology

Proof

Monodromy operator

x_i^k: buffer contents at k^{th} start of mode (1,2). For $k > 2$:

$$\begin{aligned} x_1^{k+1} &= 50 + \frac{3}{7}(x_1^k + 50) + \max\left(\frac{3}{7}(x_1^k + 50), \frac{3}{5}x_4^k\right) \\ x_2^{k+1} &= 0 \qquad x_3^{k+1} = 0 \qquad x_4^{k+1} = \frac{5}{7}(x_1^k + 50) \end{aligned} \tag{1}$$

Observation

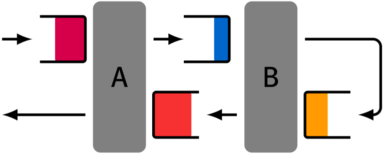
With $y_1^k = (x_1^k - 650)/7$, $y_4^k = (x_4^k - 500)/5$ we get from (1):

$$0 \leq \max(y_1^{k+2}, y_4^{k+2}) \leq \frac{6}{7}\max(y_1^k, y_4^k)$$

So system converges to fixed point $(650, 0, 0, 500)$.

Observability

Network

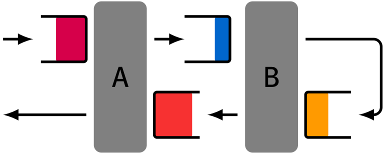


Assumptions

- Clearing policy used for machine B
- At $t = t_1$: ③ starts
- At $t = t_2 > t_1$: ③ stops

Observability

Network



Assumptions

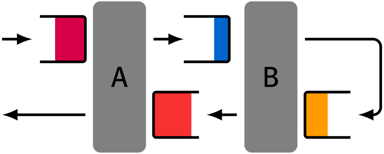
- Clearing policy used for machine B
- At $t = t_1$: ③ starts
- At $t = t_2 > t_1$: ③ stops

System state can be reconstructed at machine A

- $x_3(t_2) = 0$, and $(t_2 - t_1)/0.3 = x_3(t_1) = x_3(t_1 - 50)$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\, \mathrm{d}\, \tau$

Observability

Network



Assumptions

- Clearing policy used for machine B
- At $t = t_1$: ③ starts
- At $t = t_2 > t_1$: ③ stops

System state can be reconstructed at machine A

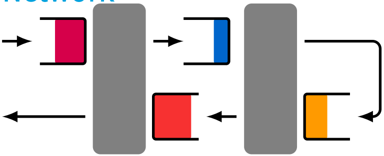
- $x_3(t_2) = 0$, and $(t_2 - t_1)/0.3 = x_3(t_1) = x_3(t_1 - 50)$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\,\mathrm{d}\,\tau$

Observation

Observability determined by network topology

Distributed controller, cf. [Lefeber, Rooda (08)]

Network



Desired behavior

Distributed controller

Serving 1: Serve at least 1000 jobs until $x_1 = 0$, then switch. Let $\bar{x}_1$ be nr of jobs served.

Serving 2: Serve at least 1000 jobs until $x_2 = 0$, then switch.

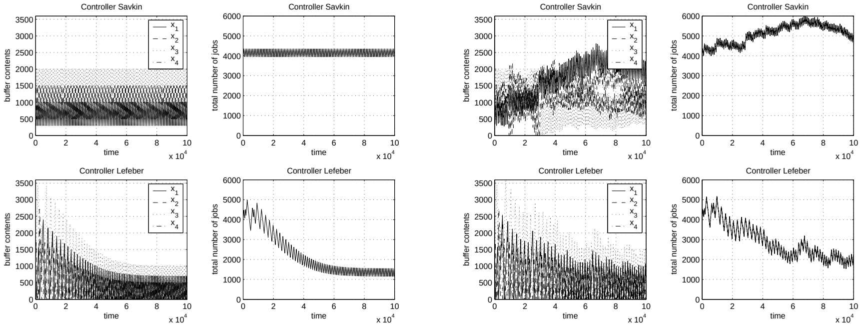
Serving 4: Let $\bar{x}_4$ be nr of jobs in Buffer 4 after setup. Serve $\bar{x}_4 + \frac{1}{2}\bar{x}_1$ jobs, then switch.

Serving 3: Empty buffer, then switch.

Simulation results

Initial condition $(1000, 1000, 1000, 1000)$.
Deterministic/Exponential service times, setup times.

Distributed controller



Technische Universiteit Eindhoven University of Technology

Conclusions

New approach

1. Determine desired system behavior (trajectory generation)
2. Derive non-distributed/centralized controller (state feedback)
3. Derive distributed/decentralized controller (output feedback)

Conclusions

New approach

1. Determine desired system behavior (trajectory generation)
2. Derive non-distributed/centralized controller (state feedback)
3. Derive distributed/decentralized controller (output feedback)

Advantage

All three problems can be considered separately

Technische Universiteit Eindhoven University of Technology

Conclusions

New approach

1. Determine desired system behavior (trajectory generation)
2. Derive non-distributed/centralized controller (state feedback)
3. Derive distributed/decentralized controller (output feedback)

Advantage

All three problems can be considered separately

Centralized control

Approach can deal with

- Arbitrary networks
- Finite buffers
- Transportation delays

Technische Universiteit Eindhoven University of Technology

Conclusions

New approach

1. Determine desired system behavior (trajectory generation)
2. Derive non-distributed/centralized controller (state feedback)
3. Derive distributed/decentralized controller (output feedback)

Advantage

All three problems can be considered separately

Centralized control

Approach can deal with

- Arbitrary networks
- Finite buffers
- Transportation delays

Decentralized control

- Observer based approach results in new, tailor-made controllers that perform better

Technische Universiteit Eindhoven University of Technology

For further reading

E. Lefeber and J.E. Rooda.
Controller design of switched linear systems with setups.
Physica A, 363(1):48–61, April 2006.

E. Lefeber and J.E. Rooda.
Controller Design for Flow Networks of Switched Servers with Setup Times: the Kumar-Seidman Case as an Illustrative Example.
Asian Journal of Control, 10(1), 55-66, 2008.