Controller design for networks of switching servers

Erjen Lefeber

Eindhoven University of Technology
Department of Mechanical Engineering

Queueing Colloquium
November 21, 2008, CWI

Acknowledgements

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

Inspired by discussions with:

- Varvara Feoktistova, Alexey Matveev (St. Petersburg)
- Jan van der Wal, Josine Bruin
- Stefan Lämmer (TU Dresden)
- Gideon Weiss, Yoni Nazarathy (Haifa)

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

Current approach

Start from policy, analyze resulting dynamics

Kumar, Seidman (1990)

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

Kumar, Seidman (1990)

Problem

Problem

How to control these networks?

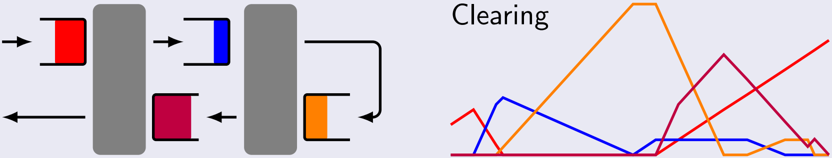
Decisions: When to switch, and to which job-type

Goals: Minimal number of jobs, minimal flow time

Current approach

Start from policy, analyze resulting dynamics

Kumar, Seidman (1990)



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)

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)

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)

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

Background: Control theory

System dynamics (linear)

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x \in R^n, u \in R^k$$
$$y(t) = Cx(t) \qquad y \in R^m$$

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

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller $u(\cdot)$, such that

$$\lim_{t\to\infty} \|x(t) - x_r(t)\| = 0.$$

Background: Control theory

System dynamics (linear)

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x \in R^n, u \in R^k$$
$$y(t) = Cx(t) \qquad y \in R^m$$

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

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller $u(\cdot)$, such that

$$\lim_{t\to\infty} \|x(t) - x_r(t)\| = 0.$$

Background: Control theory

System dynamics (linear)

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x \in R^n, u \in R^k$$
$$y(t) = Cx(t) \qquad y \in R^m$$

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

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller $u(\cdot)$, such that

$$\lim_{t\to\infty} \|x(t) - x_r(t)\| = 0.$$

Background: Control theory, Example tracking control

Controller

$$u = u_r + K(x - x_r)$$

Error dynamics

Define $e = x - x_r$, then:

$$\dot{e} = Ax + B(u_r + Ke) - (Ax_r + Bu_r) = (A + BK)e$$

Make sure that K is such that eigenvalues of $A + BK$ are in left half of complex plane.

Remark

The controller design holds for arbitrary reference.

Background: Control theory, Example tracking control

Controller

$$u = u_r + K(x - x_r)$$

Error dynamics

Define $e = x - x_r$, then:

$$\dot{e} = Ax + B(u_r + Ke) - (Ax_r + Bu_r) = (A + BK)e$$

Make sure that K is such that eigenvalues of $A + BK$ are in left half of complex plane.

Remark

The controller design holds for arbitrary reference.

Background: Control theory, Example tracking control

Controller

$$u = u_r + K(x - x_r)$$

Error dynamics

Define $e = x - x_r$, then:

$$\dot{e} = Ax + B(u_r + Ke) - (Ax_r + Bu_r) = (A + BK)e$$

Make sure that K is such that eigenvalues of $A + BK$ are in left half of complex plane.

Remark

The controller design holds for arbitrary reference.

Background: Control theory

System dynamics (linear)

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x \in R^n, u \in R^k$$
$$y(t) = Cx(t) \qquad y \in R^m$$

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller

Problem III: Observer design

Reconstruct x using only measurement of y

Background: Control theory

System dynamics (linear)

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x \in R^n, u \in R^k$$
$$y(t) = Cx(t) \qquad y \in R^m$$

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller

Problem III: Observer design

Reconstruct x using only measurement of y

Background: Control theory, Example observer design

Observer

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

Observer error dynamics

Define $e = x - \hat{x}$, then

$$\dot{e} = A\hat{x} + Bu + LCe - (Ax + Bu) = (A + LC)e$$

Make sure that L is such that eigenvalues of $A + LC$ are in left half of complex plane.

Background: Control theory

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller assuming x is available for measurement

Problem III: Observer design

Reconstruct x using only measurement of y

Problem IV: Output feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller assuming only y is available for measurement

Background: Control theory

Problem I: Trajectory generation

Determine feasible functions $x_r(t)$, $u_r(t)$.

Problem II: State feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller assuming x is available for measurement

Problem III: Observer design

Reconstruct x using only measurement of y

Problem IV: Output feedback tracking control

Given arbitrary feasible $x_r(t)$, $u_r(t)$, find a controller assuming only y is available for measurement

Background: Control theory, Example tracking control

System dynamics (linear)

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x \in R^n, u \in R^k$$
$$y(t) = Cx(t) \qquad y \in R^m$$

Dynamic output feedback tracking controller

$$u = u_r + K(\hat{x} - x_r)$$
$$\dot{\hat{x}} = A\hat{x} + Bu(t) + L(y - \hat{y})$$
$$\hat{y} = C\hat{x}$$

where K and L from previous designs can be used.

TU/e
NWO

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

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

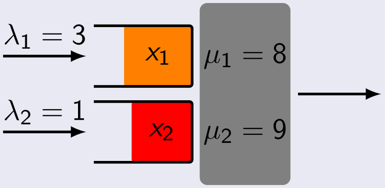
Parallels with this problem

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

Example 1: Single machine

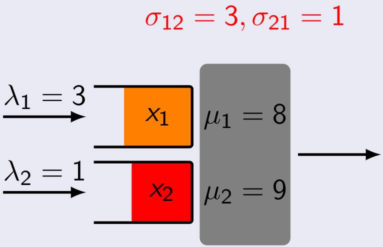
Single machine

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



Example 1: Single machine

Single machine



State

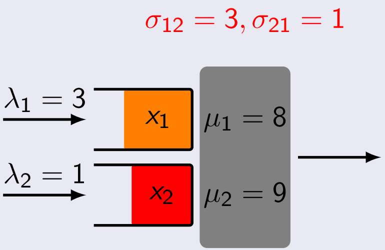
x_0	remaining setup time
x_i	buffer contents ($i = 1, 2$)
m	mode $\in \{1, 2\}$

Input

u_0	activity $\in \{①, ②, ❶, ❷\}$
u_i	service rate step $i = 1, 2$

Example 1: Single machine

Single machine



Continuous dynamics

$$\dot{x}_0(t) = \begin{cases} -1 & \text{if } u_0 \in \{❶, ❷\} \\ 0 & \text{if } u_0 \in \{①, ②\} \end{cases}$$

$$\dot{x}_1(t) = \lambda_1 - u_1(t)$$

$$\dot{x}_2(t) = \lambda_2 - u_2(t)$$

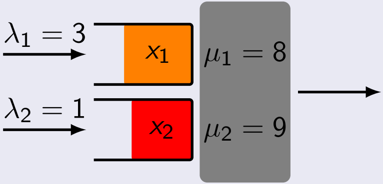
Discrete event dynamics

$x_0 := \sigma_{21}$	$m := 1$	if $u_0 =$ ❶ and $m = 2$
$x_0 := \sigma_{12}$	$m := 2$	if $u_0 =$ ❷ and $m = 1$

Example 1: Single machine

Single machine

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



Continuous dynamics

$$\dot{x}_0(t) = \begin{cases} -1 & \text{if } u_0 \in \{\text{❶}, \text{❷}\} \\ 0 & \text{if } u_0 \in \{\text{①}, \text{②}\} \end{cases}$$

$$\dot{x}_1(t) = \lambda_1 - u_1(t)$$

$$\dot{x}_2(t) = \lambda_2 - u_2(t)$$

Discrete event dynamics

$$x_0 := \sigma_{21} \qquad m := 1 \qquad \text{if } u_0 = \text{❶} \text{ and } m = 2$$

$$x_0 := \sigma_{12} \qquad m := 2 \qquad \text{if } u_0 = \text{❷} \text{ and } m = 1$$

Example 1: Single machine

Input contraints

$$
\begin{array}{llll}
u_0 \in \{❶, ❷\} & u_1 = 0 & u_2 = 0 & \text{if } x_0 > 0 \\
u_0 \in \{①, ❷\} & u_1 \leq \mu_1 & u_2 = 0 & \text{if } x_0 = 0,\ x_1 > 0,\ m = 1 \\
u_0 \in \{①, ❷\} & u_1 \leq \lambda_1 & u_2 = 0 & \text{if } x_0 = 0,\ x_1 = 0,\ m = 1 \\
u_0 \in \{❶, ②\} & u_1 = 0 & u_2 \leq \mu_2 & \text{if } x_0 = 0,\ x_2 > 0,\ m = 2 \\
u_0 \in \{❶, ②\} & u_1 = 0 & u_2 \leq \lambda_2 & \text{if } x_0 = 0,\ x_2 = 0,\ m = 2
\end{array}
$$

Objective

Minimize:

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

Example 1: Single machine

Input contraints

$u_0 \in \{❶, ❷\}$	$u_1 = 0$	$u_2 = 0$	if $x_0 > 0$
$u_0 \in \{①, ❷\}$	$u_1 \leq \mu_1$	$u_2 = 0$	if $x_0 = 0,\ x_1 > 0,\ m = 1$
$u_0 \in \{①, ❷\}$	$u_1 \leq \lambda_1$	$u_2 = 0$	if $x_0 = 0,\ x_1 = 0,\ m = 1$
$u_0 \in \{❶, ②\}$	$u_1 = 0$	$u_2 \leq \mu_2$	if $x_0 = 0,\ x_2 > 0,\ m = 2$
$u_0 \in \{❶, ②\}$	$u_1 = 0$	$u_2 \leq \lambda_2$	if $x_0 = 0,\ x_2 = 0,\ m = 2$

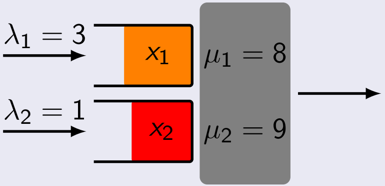
Objective

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



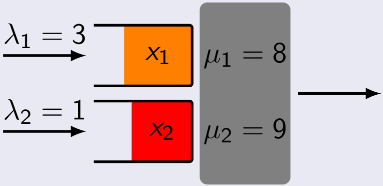
Desired behavior

Remarks

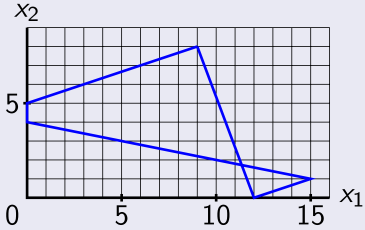
- Many existing policies assume non-idling a-priori
- Slow-mode optimal if $\lambda_1(\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2}) - (\lambda_1 - \lambda_2)(1 - \frac{\lambda_2}{\mu_2}) < 0$.
- Trade-off in wasting capacity: idle $\Leftrightarrow$ switch more often

Desired behavior (Problem I)

Single machine



Desired behavior



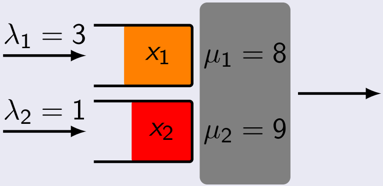
Remarks

- 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

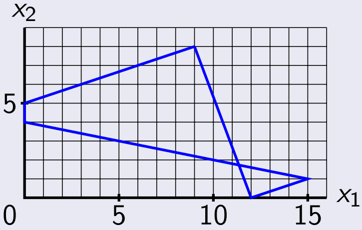
TU/e NWO

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 NWO

Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time ⇒ system settles down at constant energy level

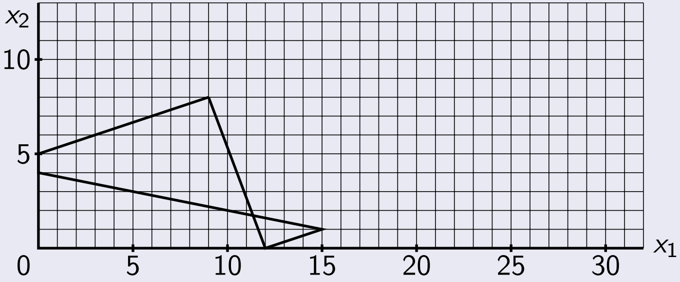
Lyapunov function candidate

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

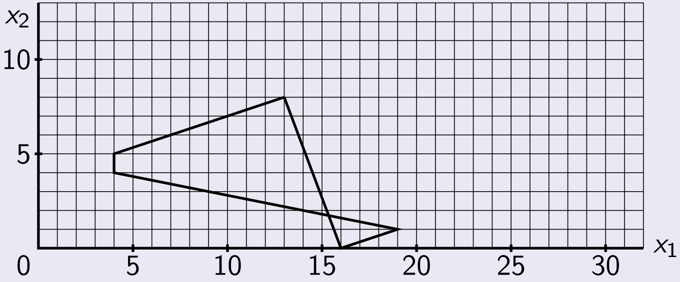


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time ⇒ system settles down at constant energy level

Lyapunov function candidate

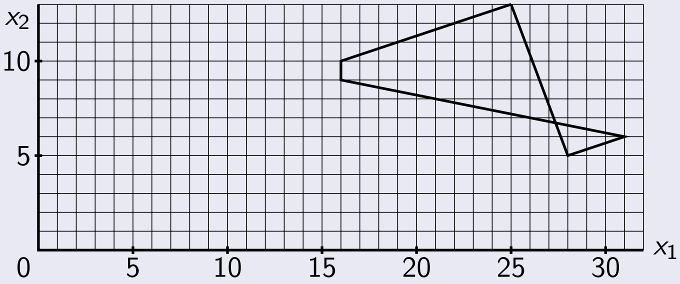


Controller design (Problem II)

Main idea

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

Lyapunov function candidate

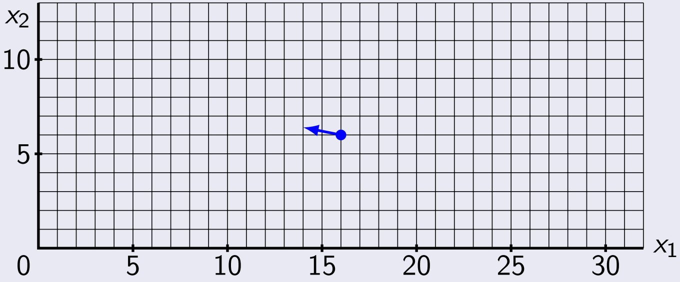


Controller design (Problem II)

Main idea

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

Lyapunov function candidate

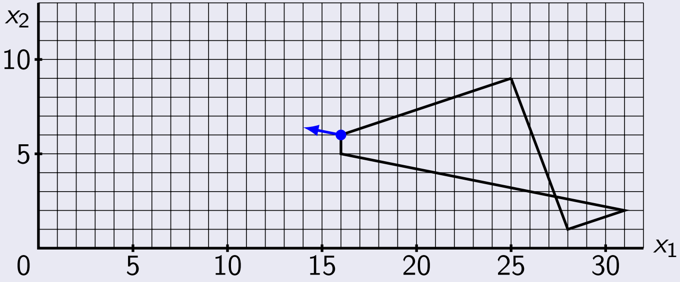


Controller design (Problem II)

Main idea

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

Lyapunov function candidate

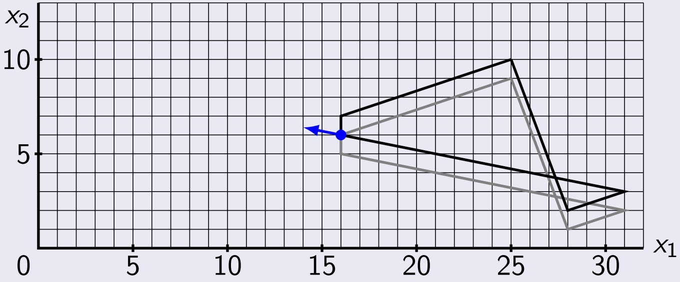


Controller design (Problem II)

Main idea

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

Lyapunov function candidate

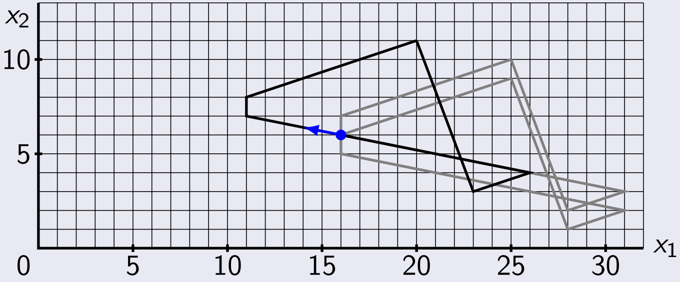


Controller design (Problem II)

Main idea

Lyapunov: if energy is decreasing all the time ⇒ system settles down at constant energy level

Lyapunov function candidate

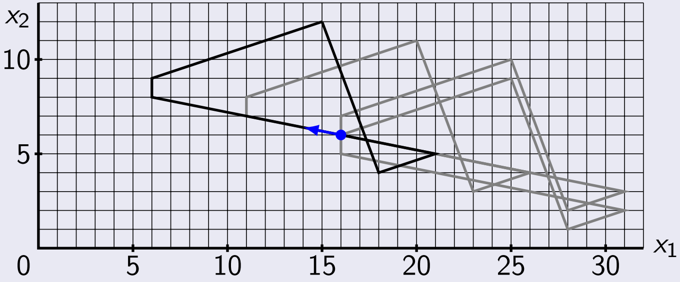


Controller design (Problem II)

Main idea

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

Lyapunov function candidate

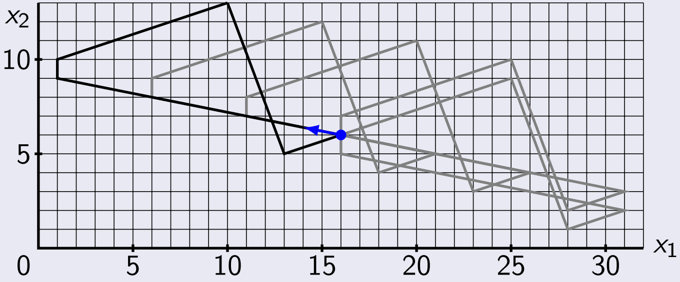


Controller design (Problem II)

Main idea

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

Lyapunov function candidate

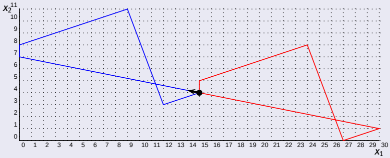


Controller design

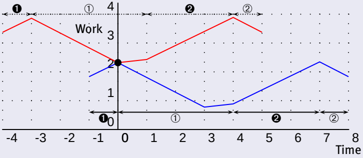
Lyapunov function candidate

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

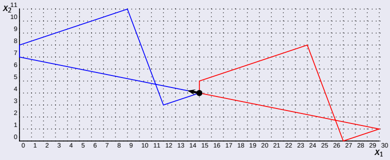
Let Lyapunov function candidate decrease as quickly as possible

Controller design

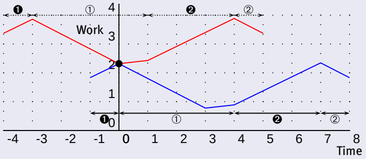
Lyapunov function candidate

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

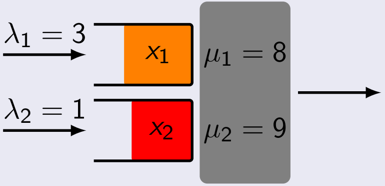


Controller design

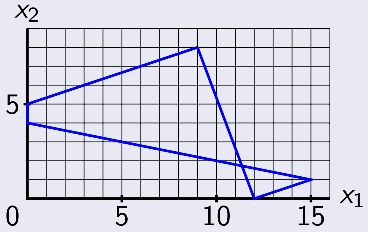
Let Lyapunov function candidate decrease as quickly as possible

Controller design (Result)

Single machine



Desired behavior

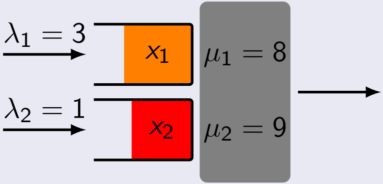


Resulting Controller, cf. [Lefeber, Rooda (2006)]

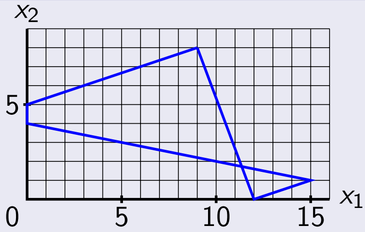
- When serving type 1:
  1. empty buffer
  2. serve until $x_2 \geq 5$
  3. switch to type 2
- When serving type 2:
  1. empty buffer
  2. serve until $x_1 \geq 12$
  3. switch to type 1

Controller design (Result)

Single machine



Desired behavior

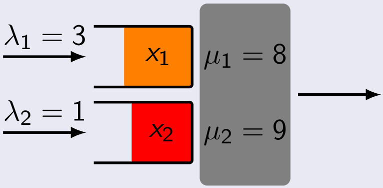


Resulting Controller, cf. [Lefeber, Rooda (2006)]

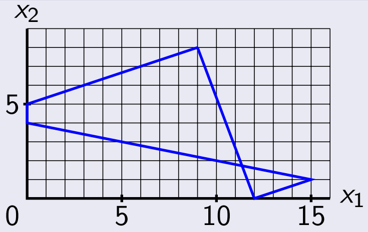
- When serving type 1:
  1. empty buffer
  2. serve until $x_2 \geq 5$
  3. switch to type 2
- When serving type 2:
  1. empty buffer
  2. serve until $x_1 \geq 12$
  3. switch to type 1

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

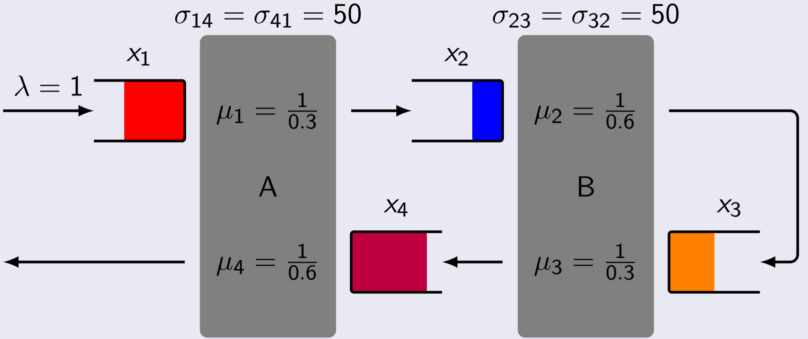
1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

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

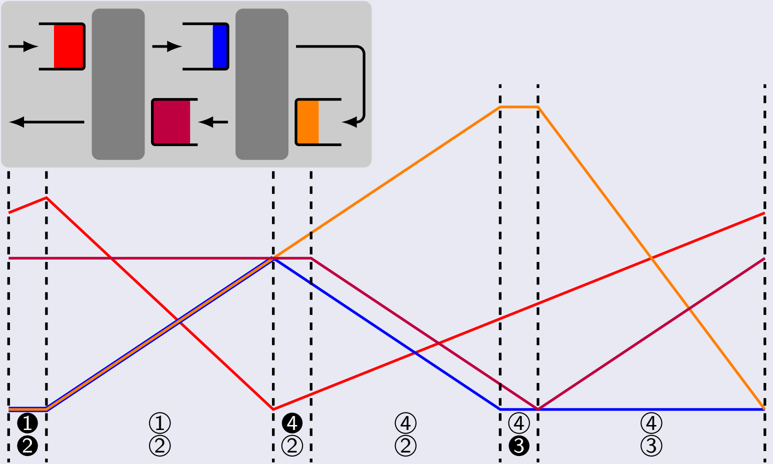


Observation

Sufficient capacity (consider period of at least 1000).

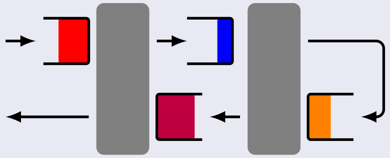
Desired behavior

Desired behavior

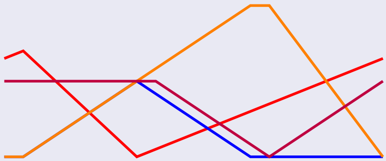


Resulting controller

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

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

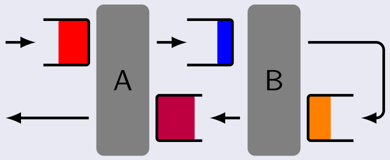
With $y_1^k = (x_1^k - 650)/7$, $y_4^k = (x_4^k - 500)/5$ we get from (1):

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

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

System state can be reconstructed at machine A

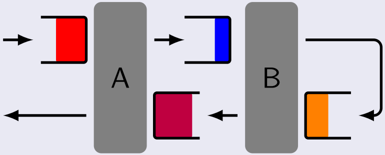
- $x_3(t_2) = 0$, and $0.3(t_2 - t_1) = x_3(t_1) = x_3(t_1 - 50)$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\, d\tau$

Observation

Observability determined by network topology

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

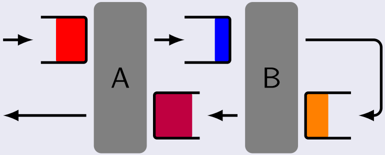
- $x_3(t_2) = 0$, and $0.3(t_2 - t_1) = x_3(t_1) = x_3(t_1 - 50)$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\,\mathrm{d}\,\tau$

Observation

Observability determined by network topology

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

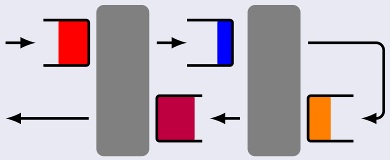
- $x_3(t_2) = 0$, and $0.3(t_2 - t_1) = x_3(t_1) = x_3(t_1 - 50)$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\, d\tau$

Observation

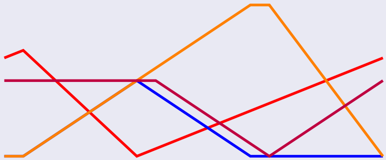
Observability determined by network topology

Distributed controller, cf. [Lefeber, Rooda (2008)]

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.

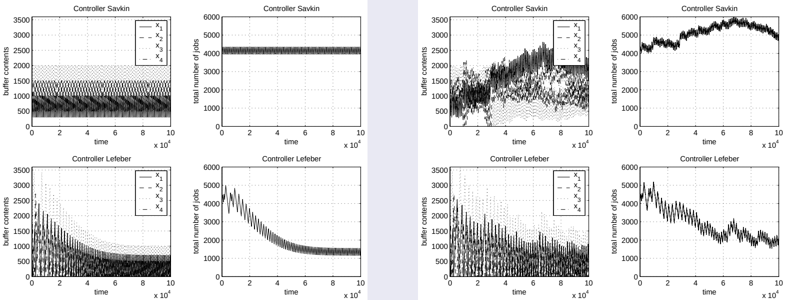
Serving 4: Let $\bar{x}_4$ be nr of jobs in Buffer 4 after setup. Serve $\bar{x}_4 + \frac{1}{2}\bar{x}_1$ jobs, then switch.

Serving 3: Empty buffer, then switch.

Simulation results

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

Distributed controller



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

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

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

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

Future work

Research

- Centralized control
  - Finalize techniques for convergence proofs
  - Derive class of controllers (instead of only one)
  - Finite buffers: reachability of desired orbit
  - Deal with parametric uncertainty; robustness if parameters are either different or time-varying.
- Decentralized control (!!PhD vacancy!!)
  - Observability (including tests)
  - Observer design
  - Stability analysis of distributed policies
- Stochastic extensions
  - Analyze performance of derived (de)centralized controllers for stochastic queueing networks

Future work

Research

- Centralized control
  - Finalize techniques for convergence proofs
  - Derive class of controllers (instead of only one)
  - Finite buffers: reachability of desired orbit
  - Deal with parametric uncertainty; robustness if parameters are either different or time-varying.
- Decentralized control (!!PhD vacancy!!)
  - Observability (including tests)
  - Observer design
  - Stability analysis of distributed policies
- Stochastic extensions
  - Analyze performance of derived (de)centralized controllers for stochastic queueing networks

TU/e NWO

Future work

Research

- Centralized control
  - Finalize techniques for convergence proofs
  - Derive class of controllers (instead of only one)
  - Finite buffers: reachability of desired orbit
  - Deal with parametric uncertainty; robustness if parameters are either different or time-varying.
- Decentralized control (!!PhD vacancy!!)
  - Observability (including tests)
  - Observer design
  - Stability analysis of distributed policies
- Stochastic extensions
  - Analyze performance of derived (de)centralized controllers for stochastic queueing networks

Adaptive control

System dynamics

$$\dot{x} = ax + u \qquad a \text{ unknown parameter}$$

Controller

$$u = -\hat{a}x - kx \qquad k > 0$$
$$\dot{\hat{a}} = \gamma x^2 \qquad \gamma > 0$$

Result

$$\lim_{t \to \infty} x(t) = 0$$

Furthermore, $\hat{a}(t)$ converges to a constant (not to a!)

Adaptive control

System dynamics

$$\dot{x} = ax + u \qquad a \text{ unknown parameter}$$

Controller

$$u = -\hat{a}x - kx \qquad k > 0$$
$$\dot{\hat{a}} = \gamma x^2 \qquad \gamma > 0$$

Result

$$\lim_{t \to \infty} x(t) = 0$$

Furthermore, $\hat{a}(t)$ converges to a constant (not to a!)

Adaptive control

System dynamics

$$\dot{x} = ax + u \qquad a \text{ unknown parameter}$$

Controller

$$u = -\hat{a}x - kx \qquad k > 0$$

$$\dot{\hat{a}} = \gamma x^2 \qquad \gamma > 0$$

Result

$$\lim_{t\to\infty} x(t) = 0$$

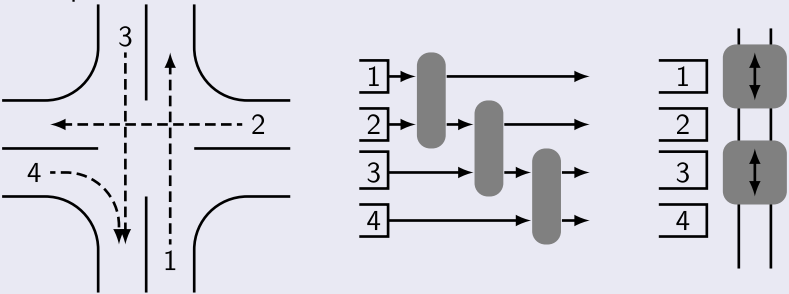
Furthermore, $\hat{a}(t)$ converges to a constant (not to a!)

System

System

Server can serve several queues simultaneously, each queue at rate μ_i, independent of the number of queues being served.

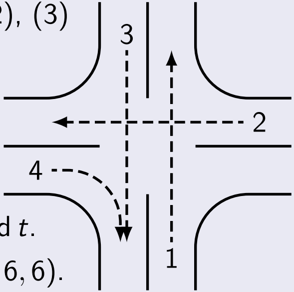
Examples



Problem

Example

- Modes: (1,4), (2,4), (1,3), (4), (1), (2), (3)
- Deterministic fluid queues
- No arrivals, i.e. $\lambda_i = 0$
- $\mu_i = 1$
- Objective: minimize
  $\int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}\,t.$
- Initial condition $(x_1, x_2, x_3, x_4) = (6, 6, 6, 6).$



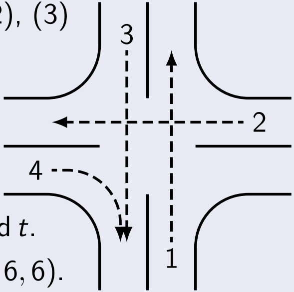
Possible policies

- Mode (1,4) for 6, Mode (2) for 6, Mode (3) for 6: costs 504.
- Mode (2,4) for 6, Mode (1,3) for 6: costs 468.
- Optimal costs: 456.

Problem

Example

- Modes: (1,4), (2,4), (1,3), (4), (1), (2), (3)
- Deterministic fluid queues
- No arrivals, i.e. $\lambda_i = 0$
- $\mu_i = 1$
- Objective: minimize $\int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}\,t$.
- Initial condition $(x_1, x_2, x_3, x_4) = (6, 6, 6, 6)$.



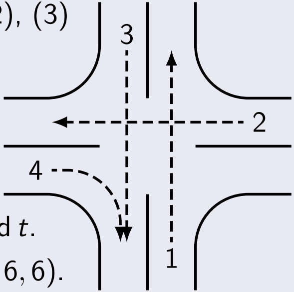
Possible policies

- Mode (1,4) for 6, Mode (2) for 6, Mode (3) for 6: costs 504.
- Mode (2,4) for 6, Mode (1,3) for 6: costs 468.
- Optimal costs: 456.

Problem

Example

- Modes: (1,4), (2,4), (1,3), (4), (1), (2), (3)
- Deterministic fluid queues
- No arrivals, i.e. $\lambda_i = 0$
- $\mu_i = 1$
- Objective: minimize
  $\int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}\,t$.
- Initial condition $(x_1, x_2, x_3, x_4) = (6, 6, 6, 6)$.



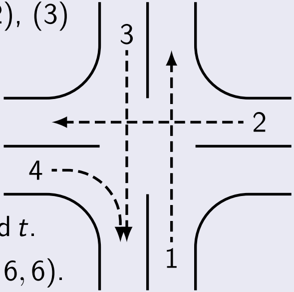
Possible policies

- Mode (1,4) for 6, Mode (2) for 6, Mode (3) for 6: costs 504.
- Mode (2,4) for 6, Mode (1,3) for 6: costs 468.
- Optimal costs: 456.

Problem

Example

- Modes: (1,4), (2,4), (1,3), (4), (1), (2), (3)
- Deterministic fluid queues
- No arrivals, i.e. $\lambda_i = 0$
- $\mu_i = 1$
- Objective: minimize $\int_0^\infty 4x_1(t) + 3x_2(t) + 2x_3(t) + 5x_4(t)\,\mathrm{d}\,t$.
- Initial condition $(x_1, x_2, x_3, x_4) = (6, 6, 6, 6)$.



Possible policies

- Mode (1,4) for 6, Mode (2) for 6, Mode (3) for 6: costs 504.
- Mode (2,4) for 6, Mode (1,3) for 6: costs 468.
- Optimal costs: 456.

Optimal controller

Initialization: Start in Mode (1,4).

Mode (1,4): Stay in mode until either $x_1 = 0$, or $x_4 = 0$, or $x_4 \leq x_2 \wedge x_1 \leq x_3 \wedge (\mu_1 c_1 - \mu_2 c_2 + \mu_3 c_3)\left(\frac{x_1}{\mu_1} + \frac{x_4}{\mu_4}\right) \leq \mu_3 c_3 \left(\frac{x_2}{\mu_2} + \frac{x_3}{\mu_3}\right)$ then switch to Mode (2,4).

Mode (2,4): Stay in mode until either $x_2 = 0$ or $x_4 = 0$, then switch to Mode (1,3).

Mode (1,3): Stay in mode until either $x_1 = 0$ or $x_3 = 0$, then switch to Mode (4).

Mode (4): Stay in mode until $x_4 = 0$, then switch to Mode (1).

Mode (1): Stay in mode until $x_1 = 0$, then switch to Mode (2).

Mode (2): Stay in mode until $x_2 = 0$, then switch to Mode (3).

Mode (3): Stay in mode until $x_3 = 0$.

With arrival rates

Some observations

- Can be formulated as SCLP (NO setup times)
- Combined modes
- μc-like rule (only for NO setup times)
- Stability conditions (odd cycle graph)
- Setup times: no μc-like rule:

$$c = (0.34, 0.33, 0.32, 0.35)^T \qquad x_0 = (30, 20, 20, 40)$$

No arrivals
μc rule: (1,4), (2,4), (1,3), 3: costs 1039.68
optimal: (2,4), (1,4), (1,3), 3: costs 1039.60