Controller design for flow networks of switched servers with setup times

The Kumar-Seidman case as an illustrative example

Erjen Lefeber

Eindhoven University of Technology

Mathematical modeling of transport and production logistics
January 11, 2008, Bremen

Motivation

Motivation

Motivation

Motivation

Problem

Problem

How to control these networks?

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

Goals: Maximal throughput, 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: Maximal throughput, minimal flow time

Current approach

Start from policy, analyze resulting dynamics

Kumar, Seidman (1990)

Problem

Problem

How to control these networks?

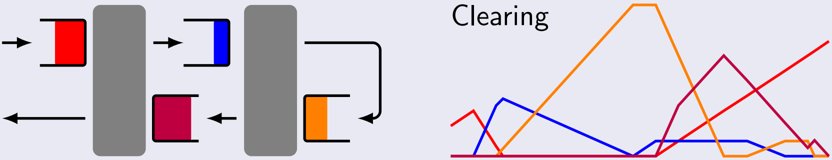
Decisions: When to switch, and to which job-type

Goals: Maximal throughput, 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)

Main idea

Important observation

"The main interest is in the resulting behavior. So why not use that as a starting point?"

Approach

Start from desired behavior and *design* policy, instead of start from policy and analyze resulting dynamics

Consequence

Separation of concern: desired behavior and controller can be designed separately.

Main idea

Important observation

"The main interest is in the resulting behavior. So why not use that as a starting point?"

Approach

Start from desired behavior and *design* policy, instead of start from policy and analyze resulting dynamics

Consequence

Separation of concern: desired behavior and controller can be designed separately.

Main idea

Important observation

"The main interest is in the resulting behavior. So why not use that as a starting point?"

Approach

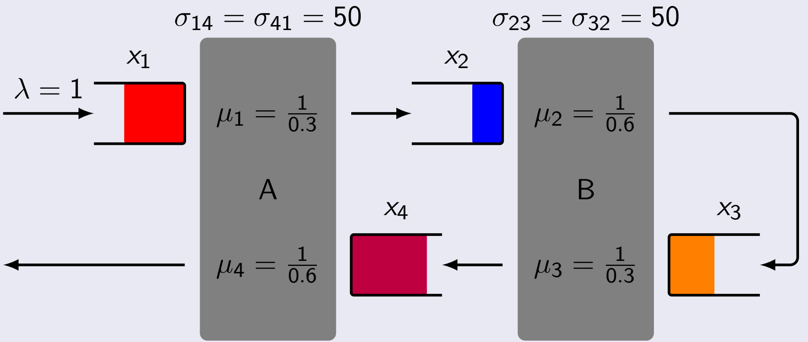
Start from desired behavior and *design* policy, instead of start from policy and analyze resulting dynamics

Consequence

Separation of concern: desired behavior and controller can be designed separately.

Kumar-Seidman case

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



Observation

Sufficient capacity (consider period of at least 1000).

Model (hybrid)

State

x_0^A, x_0^B	remaining setup time machine A,B
x_i	buffer contents ($i \in \{1,2,3,4\}$)
$m = (m^A, m^B)$	mode $\in \{(1,2),(1,3),(4,2),(4,3)\}$

Input

u_0^A, u_0^B	activity $\in$ {①, ②, ③, ④, ❶, ❷, ❸, ❹}
u_i	service rate step $i \in \{1,2,3,4\}$

Model (hybrid)

State

x_0^A, x_0^B	remaining setup time machine A,B
x_i	buffer contents ($i \in \{1, 2, 3, 4\}$)
$m = (m^A, m^B)$	mode $\in \{(1, 2), (1, 3), (4, 2), (4, 3)\}$

Input

u_0^A, u_0^B	activity $\in \{①, ②, ③, ④, ❶, ❷, ❸, ❹\}$
u_i	service rate step $i \in \{1, 2, 3, 4\}$)

Model (hybrid)

Continuous dynamics

$$\dot{x}_0^A(t) = \begin{cases} -1 & \text{if } u_0^A \in \{\text{❶}, \text{❹}\} \\ 0 & \text{if } u_0^A \in \{\text{①}, \text{④}\} \end{cases} \qquad \dot{x}_0^B(t) = \begin{cases} -1 & \text{if } u_0^B \in \{\text{❷}, \text{❸}\} \\ 0 & \text{if } u_0^B \in \{\text{②}, \text{③}\} \end{cases}$$

$$\dot{x}_1(t) = \lambda - u_1(t) \qquad \dot{x}_2(t) = u_1(t) - u_2(t)$$

$$\dot{x}_4(t) = u_3(t) - u_4(t) \qquad \dot{x}_3(t) = u_2(t) - u_3(t).$$

Discrete event dynamics

$$x_0^A := \sigma_{14}; \quad m^A := 4 \quad \text{if } u_0^A = \text{❹} \text{ and } m^A = 1$$

$$x_0^A := \sigma_{41}; \quad m^A := 1 \quad \text{if } u_0^A = \text{❶} \text{ and } m^A = 4$$

$$x_0^B := \sigma_{23}; \quad m^B := 3 \quad \text{if } u_0^B = \text{❸} \text{ and } m^B = 2$$

$$x_0^B := \sigma_{32}; \quad m^B := 2 \quad \text{if } u_0^B = \text{❷} \text{ and } m^B = 3$$

Model (hybrid)

Continuous dynamics

$$\dot{x}_0^A(t) = \begin{cases} -1 & \text{if } u_0^A \in \{❶, ❹\} \\ 0 & \text{if } u_0^A \in \{①, ④\} \end{cases} \qquad \dot{x}_0^B(t) = \begin{cases} -1 & \text{if } u_0^B \in \{❷, ❸\} \\ 0 & \text{if } u_0^B \in \{②, ③\} \end{cases}$$

$$\dot{x}_1(t) = \lambda - u_1(t) \qquad \dot{x}_2(t) = u_1(t) - u_2(t)$$

$$\dot{x}_4(t) = u_3(t) - u_4(t) \qquad \dot{x}_3(t) = u_2(t) - u_3(t).$$

Discrete event dynamics

$$x_0^A := \sigma_{14}; \quad m^A := 4 \quad \text{if } u_0^A = ❹ \text{ and } m^A = 1$$

$$x_0^A := \sigma_{41}; \quad m^A := 1 \quad \text{if } u_0^A = ❶ \text{ and } m^A = 4$$

$$x_0^B := \sigma_{23}; \quad m^B := 3 \quad \text{if } u_0^B = ❸ \text{ and } m^B = 2$$

$$x_0^B := \sigma_{32}; \quad m^B := 2 \quad \text{if } u_0^B = ❷ \text{ and } m^B = 3$$

Model (hybrid)

Input constraints

$u_0^A \in \{\text{❶}, \text{❹}\}, u_1 = 0, u_4 = 0$	if $x_0^A > 0$
$u_0^A \in \{\text{①}, \text{❹}\}, u_1 \leq \mu_1, u_4 = 0$	if $x_0^A = 0, x_1 > 0, m^A = 1$
$u_0^A \in \{\text{①}, \text{❹}\}, u_1 \leq \lambda, u_4 = 0$	if $x_0^A = 0, x_1 = 0, m^A = 1$
$u_0^A \in \{\text{❶}, \text{④}\}, u_1 = 0, u_4 \leq \mu_4$	if $x_0^A = 0, x_4 > 0, m^A = 4$
$u_0^A \in \{\text{❶}, \text{④}\}, u_1 = 0, u_4 \leq \min(u_3, \mu_4)$	if $x_0^A = 0, x_4 = 0, m^A = 4$
$u_0^B \in \{\text{❷}, \text{❸}\}, u_2 = 0, u_3 = 0$	if $x_0^B > 0$
$u_0^B \in \{\text{②}, \text{❸}\}, u_2 \leq \mu_2, u_3 = 0$	if $x_0^B = 0, x_2 > 0, m^B = 2$
$u_0^B \in \{\text{②}, \text{❸}\}, u_2 \leq \min(u_1, \mu_2), u_3 = 0$	if $x_0^B = 0, x_2 = 0, m^B = 2$
$u_0^B \in \{\text{❷}, \text{③}\}, u_2 = 0, u_3 \leq \mu_3$	if $x_0^B = 0, x_3 > 0, m^B = 3$
$u_0^B \in \{\text{❷}, \text{③}\}, u_2 = 0, u_3 \leq u_2$	if $x_0^B = 0, x_3 = 0, m^B = 3$

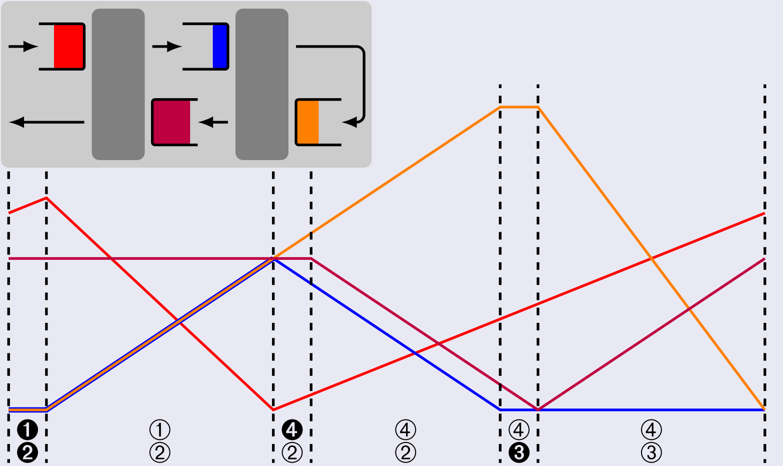
Model (hybrid)

Input constraints

$u_0^A \in \{\text{❶}, \text{❹}\}, u_1 = 0, u_4 = 0$	if $x_0^A > 0$
$u_0^A \in \{\text{①}, \text{❹}\}, u_1 \leq \mu_1, u_4 = 0$	if $x_0^A = 0, x_1 > 0, m^A = 1$
$u_0^A \in \{\text{①}, \text{❹}\}, u_1 \leq \lambda, u_4 = 0$	if $x_0^A = 0, x_1 = 0, m^A = 1$
$u_0^A \in \{\text{❶}, \text{④}\}, u_1 = 0, u_4 \leq \mu_4$	if $x_0^A = 0, x_4 > 0, m^A = 4$
$u_0^A \in \{\text{❶}, \text{④}\}, u_1 = 0, u_4 \leq \min(u_3, \mu_4)$	if $x_0^A = 0, x_4 = 0, m^A = 4$
$u_0^B \in \{\text{❷}, \text{❸}\}, u_2 = 0, u_3 = 0$	if $x_0^B > 0$
$u_0^B \in \{\text{②}, \text{❸}\}, u_2 \leq \mu_2, u_3 = 0$	if $x_0^B = 0, x_2 > 0, m^B = 2$
$u_0^B \in \{\text{②}, \text{❸}\}, u_2 \leq \min(u_1, \mu_2), u_3 = 0$	if $x_0^B = 0, x_2 = 0, m^B = 2$
$u_0^B \in \{\text{❷}, \text{③}\}, u_2 = 0, u_3 \leq \mu_3$	if $x_0^B = 0, x_3 > 0, m^B = 3$
$u_0^B \in \{\text{❷}, \text{③}\}, u_2 = 0, u_3 \leq u_2$	if $x_0^B = 0, x_3 = 0, m^B = 3$

Desired behavior

Desired behavior



Controller design

Main idea

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

Challenge

Determine energy-function (based on desired periodic orbit)

Observation I

Desired periodic orbit provides a fixed sequence of modes, with a given duration.

Controller design

Main idea

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

Challenge

Determine energy-function (based on desired periodic orbit)

Observation I

Desired periodic orbit provides a fixed sequence of modes, with a given duration.

Controller design

Main idea

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

Challenge

Determine energy-function (based on desired periodic orbit)

Observation I

Desired periodic orbit provides a fixed sequence of modes, with a given duration.

Controller design

Observation II

Blindly applying fixed sequence of modes for corresponding duration makes system converge to translated desired periodic orbit, i.e. with additional lots in buffers (A.V. Savkin, 1998)

Observation III

Remaining duration of current mode can still be chosen

Final ingredient

Amount of work: $1.8x_1 + 1.5x_2 + 0.9x_3 + 0.6x_4$

Controller design

Observation II

Blindly applying fixed sequence of modes for corresponding duration makes system converge to translated desired periodic orbit, i.e. with additional lots in buffers (A.V. Savkin, 1998)

Observation III

Remaining duration of current mode can still be chosen

Final ingredient

Amount of work: $1.8x_1 + 1.5x_2 + 0.9x_3 + 0.6x_4$

Controller design

Observation II

Blindly applying fixed sequence of modes for corresponding duration makes system converge to translated desired periodic orbit, i.e. with additional lots in buffers (A.V. Savkin, 1998)

Observation III

Remaining duration of current mode can still be chosen

Final ingredient

Amount of work: $1.8x_1 + 1.5x_2 + 0.9x_3 + 0.6x_4$

Controller design

Notice

- Current state
- Remaining duration of current mode

⇒ Additional amount of work

Lyapunov function candidate

For given state: the lowest possible additional amount of work

Controller design

Over all possible inputs: pick one which makes Lyapunov function candidate decrease the most.

Controller design

Notice

- Current state
- Remaining duration of current mode

$\Rightarrow$ Additional amount of work

Lyapunov function candidate

For given state: the lowest possible additional amount of work

Controller design

Over all possible inputs: pick one which makes Lyapunov function candidate decrease the most.

Controller design

Notice

- Current state
- Remaining duration of current mode

$\Rightarrow$ Additional amount of work

Lyapunov function candidate

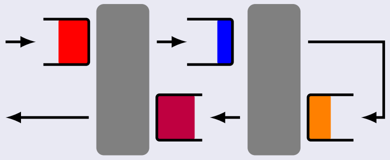
For given state: the lowest possible additional amount of work

Controller design

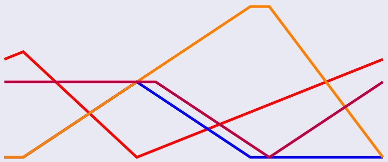
Over all possible inputs: pick one which makes Lyapunov function candidate decrease the most.

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 $x_3 = 0$

Resulting controller

Proof

Buffer amounts as A starts serving 1 for k^{th} time: $(x_1^k, x_2^k, x_3^k, x_4^k)$
Then for $k > 2$:

$$x_1^{k+1} = 100 + \frac{3}{7}x_1^k + \max(\frac{3}{7}x_1^k, \frac{3}{5}x_4^k) \qquad x_2^{k+1} = 0$$

$$x_4^{k+1} = \max(500, \frac{5}{7}x_1^k) \qquad x_3^{k+1} = 0$$

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 $(700, 0, 0, 500)$.

Remark

Centralized controller, i.e. non-distributed

Resulting controller

Proof

Buffer amounts as A starts serving 1 for k^{th} time: $(x_1^k, x_2^k, x_3^k, x_4^k)$
Then for $k > 2$:

$$x_1^{k+1} = 100 + \frac{3}{7}x_1^k + \max(\frac{3}{7}x_1^k, \frac{3}{5}x_4^k) \qquad x_2^{k+1} = 0$$
$$x_4^{k+1} = \max(500, \frac{5}{7}x_1^k) \qquad x_3^{k+1} = 0$$

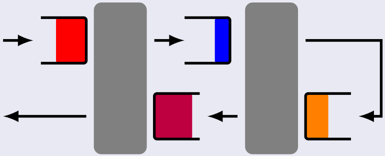
Since $x_1^{k+1} \leq 100 + \frac{3}{7}x_1^k + \frac{3}{7}\max(x_1^k, x_1^{k-1})$:
Contraction with fixed point $(700, 0, 0, 500)$.

Remark

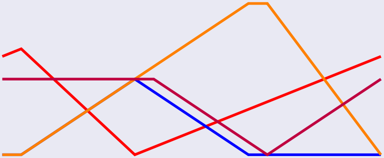
Centralized controller, i.e. non-distributed

Distributed controller

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.

Conclusions

Non-distributed/centralized control

- Given a feasible periodic orbit, a controller can be derived.
- Approach can deal with
  - General networks
  - Finite buffers
  - Transportation delays

Distributed control

- For case was shown that distributed implementation exists
- Relates to observability

Conclusions

Non-distributed/centralized control

- Given a feasible periodic orbit, a controller can be derived.
- Approach can deal with
  - General networks
  - Finite buffers
  - Transportation delays

Distributed control

- For case was shown that distributed implementation exists
- Relates to observability

Concluding remarks

Ideas from control theory can be useful

1. Determine optimal behavior (trajectory generation)
2. Derive centralized controller (state feedback control)
3. Derive decentralized controllers (dyn. output feedback)

Many questions remaining

- How to find good (or even optimal) network behavior?
- How to design decentralized controllers (observability)?
- Robustness against parameter changes?
- What if network is modified?
- What if arrival rate not constant?
- What if routing is not fixed?

Concluding remarks

Ideas from control theory can be useful

1. Determine optimal behavior (trajectory generation)
2. Derive centralized controller (state feedback control)
3. Derive decentralized controllers (dyn. output feedback)

Many questions remaining

- How to find good (or even optimal) network behavior?
- How to design decentralized controllers (observability)?
- Robustness against parameter changes?
- What if network is modified?
- What if arrival rate not constant?
- What if routing is not fixed?