Netherlands Organisation for Scientific Research

Control of Multi-class Queueing Networks with Infinite Virtual Queues

Statistics, Modelling and Operations Research Seminar

Erjen Lefeber (TU/e)

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Brisbane, February 5, 2013

Starting remarks

Acknowledgment

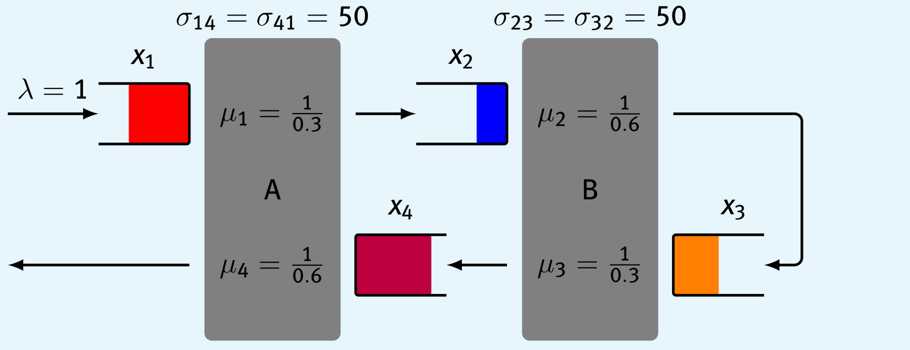
This work is supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 639.072.072).

Co-author

Joint work with Gideon Weiss

Multiclass queueing network: Kumar-Seidman

Kumar-Seidman: Trans. Autom. Ctrl. 35(3) 1990

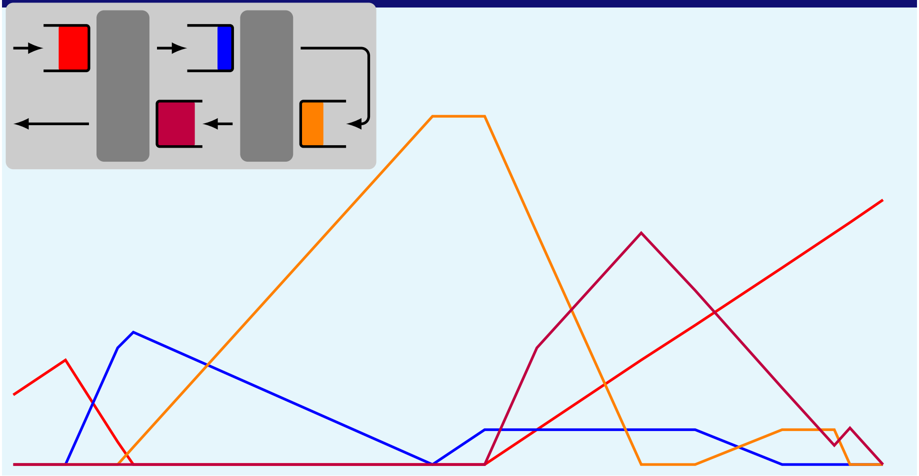


Observation

Sufficient capacity (consider period of at least 1000).

Clearing policy

Resulting behavior



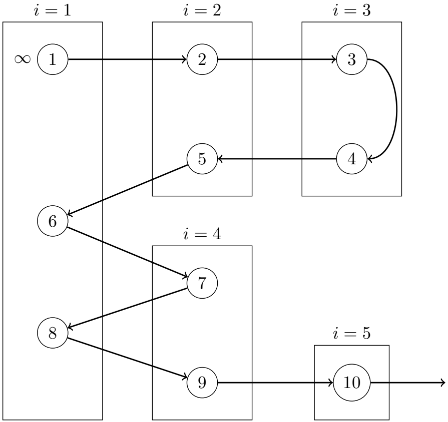
Technische Universiteit Eindhoven University of Technology

“standard” multi class queueing network

Status

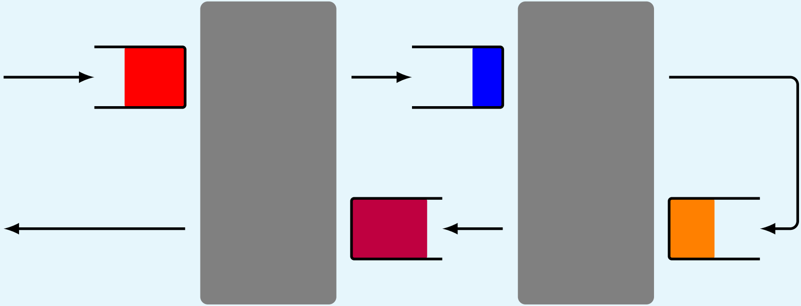
- For given priority policy: network not always stable
- If for all servers $\rho_i < 1$: stabilizing policies exist
  e.g., maximum pressure
- Under technical conditions:
  stability of fluid limit model $\rightarrow$ stability of stochastic model

Multi-class queueing network with IVQs



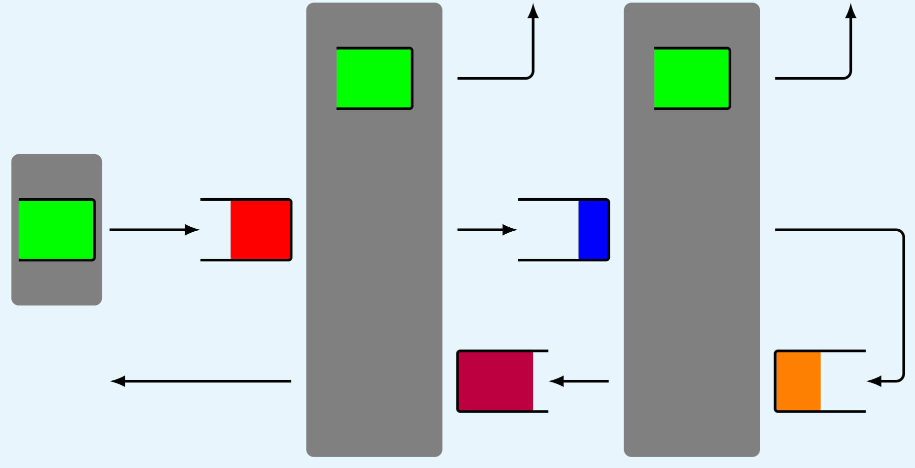
Kumar Seidman as MCQN with IVQ

System



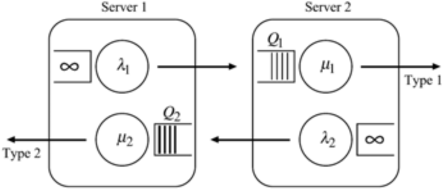
Kumar Seidman as MCQN with IVQ

System



Example: Push pull queueing system

Kopzon, Weiss (2002); Kopzon, Nazarathy, Weiss (2009); Nazarathy, Weiss (2010)



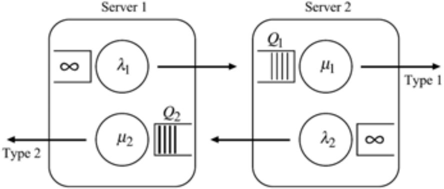
Static production planning problem

$$\max_{u,\alpha} w'\alpha$$

α_1, α_2 nominal input rates

u_i fraction of time spent on class i

Kopzon, Weiss (2002); Kopzon, Nazarathy, Weiss (2009); Nazarathy, Weiss (2010)



Static production planning problem

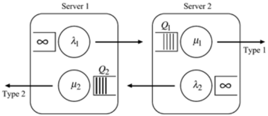
$$\max_{u,\alpha} w'\alpha$$

α_1, α_2 nominal input rates

u_i fraction of time spent on class i

Technische Universiteit Eindhoven University of Technology

Example: Push pull queueing system

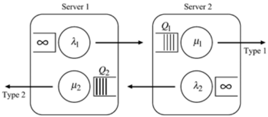


$$\max_{u,\alpha} \; w_1\alpha_1 + w_2\alpha_2$$

$$\text{s.t.} \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ \lambda_1 & -\mu_1 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_2 & -\mu_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \\ \alpha_2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$u, \alpha \geq 0$$



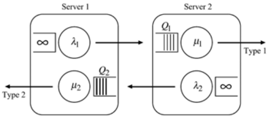
Three possible solutions (excluding singular case)

1. $\alpha_1 = \min\{\lambda_1, \mu_1\}$, $\alpha_2 = 0$,
2. $\alpha_1 = 0$, $\alpha_2 = \min\{\lambda_2, \mu_2\}$,
3. $\alpha_1 = \frac{\lambda_1\mu_1(\lambda_2-\mu_2)}{\lambda_1\lambda_2-\mu_1\mu_2}$, $\alpha_2 = \frac{\lambda_2\mu_2(\lambda_1-\mu_1)}{\lambda_1\lambda_2-\mu_1\mu_2}$.

Interesting solution: solution 3

- $\rho_1 = \rho_2 = 1$ (full utilization of servers)
- $\tilde{\rho}_1 = \frac{\lambda_2(\lambda_1-\mu_1)}{\lambda_1\lambda_2-\mu_1\mu_2} < 1$, $\tilde{\rho}_2 = \frac{\lambda_1(\lambda_2-\mu_2)}{\lambda_1\lambda_2-\mu_1\mu_2} < 1$.

Example: Push pull queueing system



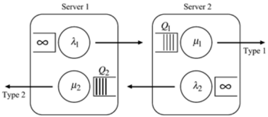
Three possible solutions (excluding singular case)

1. $\alpha_1 = \min\{\lambda_1, \mu_1\}$, $\alpha_2 = 0$,
2. $\alpha_1 = 0$, $\alpha_2 = \min\{\lambda_2, \mu_2\}$,
3. $\alpha_1 = \frac{\lambda_1\mu_1(\lambda_2-\mu_2)}{\lambda_1\lambda_2-\mu_1\mu_2}$, $\alpha_2 = \frac{\lambda_2\mu_2(\lambda_1-\mu_1)}{\lambda_1\lambda_2-\mu_1\mu_2}$.

Interesting solution: solution 3

- $\rho_1 = \rho_2 = 1$ (full utilization of servers)
- $\tilde{\rho}_1 = \frac{\lambda_2(\lambda_1-\mu_1)}{\lambda_1\lambda_2-\mu_1\mu_2} < 1$, $\tilde{\rho}_2 = \frac{\lambda_1(\lambda_2-\mu_2)}{\lambda_1\lambda_2-\mu_1\mu_2} < 1$.

Example: Push pull queueing system



Question

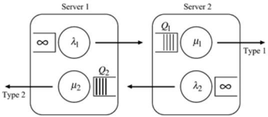
Can we stabilize system with $\rho_i = 1$ and $\tilde{\rho}_i < 1$?

Two cases

inherently stable case: $\lambda_1 < \mu_1$ and $\lambda_2 < \mu_2$

inherently unstable case: $\lambda_1 > \mu_1$ and $\lambda_2 > \mu_2$

Example: Push pull queueing system



Question

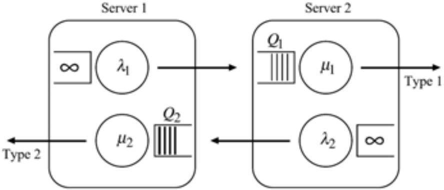
Can we stabilize system with $\rho_i = 1$ and $\tilde{\rho}_i < 1$?

Two cases

inherently stable case: $\lambda_1 < \mu_1$ and $\lambda_2 < \mu_2$

inherently unstable case: $\lambda_1 > \mu_1$ and $\lambda_2 > \mu_2$

Inherently stable case: $\lambda_1 < \mu_1$, $\lambda_2 < \mu_2$



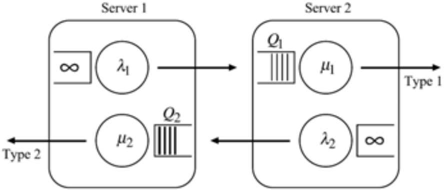
Positive result

Pull priority stabilizes network

Observation

For inherently unstable case: pull priority is not stabilizing.

Inherently stable case: $\lambda_1 < \mu_1$, $\lambda_2 < \mu_2$



Positive result

Pull priority stabilizes network

Observation

For inherently unstable case: pull priority is not stabilizing.

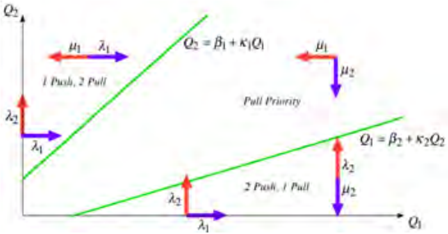
Technische Universiteit Eindhoven University of Technology

Inherently unstable case: $\lambda_1 > \mu_1, \lambda_2 > \mu_2$

Kopzon, Nazarathy, Weiss (2009); Nazarathy, Weiss (2010):

Positive result

Threshold policy stabilizes network



Observation

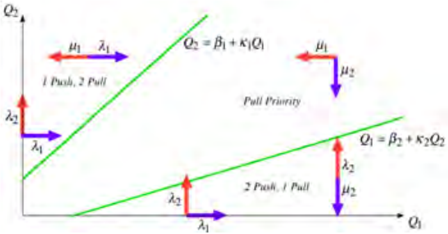
Global network state needs to be taken into account.

Inherently unstable case: $\lambda_1 > \mu_1, \lambda_2 > \mu_2$

Kopzon, Nazarathy, Weiss (2009); Nazarathy, Weiss (2010):

Positive result

Threshold policy stabilizes network



Observation

Global network state needs to be taken into account.

Problem

Guo, Lefeber, Nazarathy, Weiss, Zhang (2011,2013):

Key research question

Can we stabilize a MCQN-IVQ with $\tilde{\rho}_i < 1$ for all servers?

Some positive results

- IVQ re-entrant line (LBFS stable; FBFS not necessarily)
- Two re-entrant lines on two servers (pull priority)
- Ring of machines (pull priority)

Fluid model framework for verifying stability

Problem

Guo, Lefeber, Nazarathy, Weiss, Zhang (2011,2013):

Key research question

Can we stabilize a MCQN-IVQ with $\tilde{\rho}_i < 1$ for all servers?

Some positive results

- IVQ re-entrant line (LBFS stable; FBFS not necessarily)
- Two re-entrant lines on two servers (pull priority)
- Ring of machines (pull priority)

Fluid model framework for verifying stability

Problem setting

s Servers

- 1 IVQ, $n_i \geq 0$ std queues
- $\rho = 1$, $\tilde{\rho} < 1$

Assumptions

- P has spectral radius < 1, i.e. $(I - P')$ invertible

Data

- constituency matrix C
- $n \times n$ Routing matrix P
- $s \times n$ matrix P_{IVQ}
- IVQ: $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_s) > 0$
- Std: $M = \text{diag}(\mu_1, \dots, \mu_n) > 0$

Dynamics fluid model ($u_j(t)$ fraction of time spent on std. queue j)

$$\dot{Q}(t) = P'_{\text{IVQ}}\Lambda[1 - Cu(t)] - (I - P')Mu(t) \qquad Q(0) = Q_0$$

subject to

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Technische Universiteit Eindhoven University of Technology

Problem setting

s Servers

- 1 IVQ, $n_i \geq 0$ std queues
- $\rho = 1$, $\tilde{\rho} < 1$

Assumptions

- P has spectral radius < 1, i.e. $(I - P')$ invertible

Data

- constituency matrix C
- $n \times n$ Routing matrix P
- $s \times n$ matrix P_{IVQ}
- IVQ: $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_s) > 0$
- Std: $M = \mathrm{diag}(\mu_1, \ldots, \mu_n) > 0$

Dynamics fluid model ($u_j(t)$ fraction of time spent on std. queue j)

$$\dot{Q}(t) = P'_{\mathrm{IVQ}}\Lambda[1 - Cu(t)] - (I - P')Mu(t) \qquad Q(0) = Q_0$$

subject to

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Technische Universiteit Eindhoven University of Technology

Problem setting

Dynamics fluid model

$$\dot{Q}(t) = P'_{\text{IVQ}}\Lambda[1 - Cu(t)] - (I - P')Mu(t) \qquad Q(0) = Q_0$$

$$= \underbrace{P'_{\text{IVQ}}\Lambda 1}_{\alpha} - \underbrace{[P'_{\text{IVQ}}\Lambda C + (I - P')M]}_{R}\, u(t)$$

subject to

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Additional assumptions

- Controllable system, i.e. R is invertable
- All standard queues are served: $u^* = R^{-1}\alpha > 0$
- $\tilde{\rho} < 1$, i.e. $CR^{-1}\alpha < 1$

TU/e Technische Universiteit Eindhoven University of Technology

Problem setting

Dynamics fluid model

$$\begin{aligned}\dot{Q}(t) &= P'_{\mathrm{IVQ}}\Lambda[1 - Cu(t)] - (I - P')Mu(t) \qquad\qquad Q(0) = Q_0 \\ &= \underbrace{P'_{\mathrm{IVQ}}\Lambda 1}_{\alpha} - \underbrace{\left[P'_{\mathrm{IVQ}}\Lambda C + (I - P')M\right]}_{R} u(t)\end{aligned}$$

subject to

$$0 \leq Q(t) \qquad\qquad 0 \leq u(t) \qquad\qquad Cu(t) \leq 1$$

Additional assumptions

- Controllable system, i.e. R is invertable
- All standard queues are served: $u^* = R^{-1}\alpha > 0$
- $\tilde{\rho} < 1$, i.e. $CR^{-1}\alpha < 1$

Problem setting (summary)

Dynamics

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$\text{subject to} \quad 0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

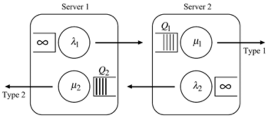
Assumptions

- $I - P'$ and R are invertible (also $(I - P')^{-1} \geq 0$)
- $0 < R^{-1}\alpha = u^*$
- $CR^{-1}\alpha < 1$

Problem

Determine stabilizing u (preferably not $u(t)$ but $u[Q(t)]$)

Example



Dynamics:

$$\begin{bmatrix}\dot{Q}_1(t)\\ \dot{Q}_2(t)\end{bmatrix} = \begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix} - \begin{bmatrix}\mu_1 & \lambda_1\\ \lambda_2 & \mu_2\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix} \qquad C = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$$

Constraints

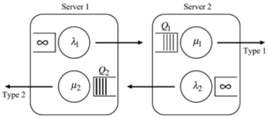
$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad u(t) \leq 1$$

Assumptions:

R invertible: $\mu_1\mu_2 \neq \lambda_1\lambda_2$ or $\frac{\lambda_1}{\mu_1}\frac{\lambda_2}{\mu_2} = \varrho_1\varrho_2 \neq 1$

$0 < R^{-1}\alpha$, $CR^{-1}\alpha < 1$: $\frac{1-\varrho_1}{1-\varrho_1\varrho_2} > 0$, $\frac{1-\varrho_2}{1-\varrho_1\varrho_2} > 0$

Example



Dynamics:

$$\begin{bmatrix} \dot{Q}_1(t) \\ \dot{Q}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} \mu_1 & \lambda_1 \\ \lambda_2 & \mu_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Constraints

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad u(t) \leq 1$$

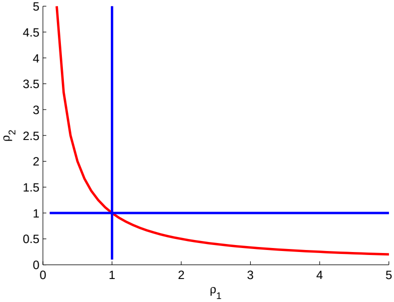
Assumptions:

R invertible: $\mu_1 \mu_2 \neq \lambda_1 \lambda_2$ or $\frac{\lambda_1}{\mu_1} \frac{\lambda_2}{\mu_2} = \varrho_1 \varrho_2 \neq 1$

$0 < R^{-1}\alpha$, $CR^{-1}\alpha < 1$: $\frac{1-\varrho_1}{1-\varrho_1\varrho_2} > 0$, $\frac{1-\varrho_2}{1-\varrho_1\varrho_2} > 0$

Example

Conditions: $\varrho_1\varrho_2 \neq 1$, $\frac{1-\varrho_1}{1-\varrho_1\varrho_2} > 0$, $\frac{1-\varrho_2}{1-\varrho_1\varrho_2} > 0$



Example: uncontrollable case

Some words about case $\lambda_1 = \mu_1, \lambda_2 = \mu_2$, i.e., R not invertible

Uncontrollable dynamics

$$\begin{bmatrix} \dot{Q}_1(t) \\ \dot{Q}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} \lambda_1 & \lambda_1 \\ \lambda_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Define change of coordinates:

$$z_1(t) = Q_1(t) + Q_2(t) \qquad z_2(t) = \lambda_2 Q_1(t) - \lambda_1 Q_2(t)$$

Then we have

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 + \lambda_2 \\ 0 \end{bmatrix} - \begin{bmatrix} \lambda_1 + \lambda_2 & \lambda_1 + \lambda_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

In particular the variable $z_2(t)$ evolves independent of the policy chosen.

Example: uncontrollable case

Some words about case $\lambda_1 = \mu_1$, $\lambda_2 = \mu_2$, i.e., R not invertible
Uncontrollable dynamics

$$\begin{bmatrix} \dot{Q}_1(t) \\ \dot{Q}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} \lambda_1 & \lambda_1 \\ \lambda_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Define change of coordinates:

$$z_1(t) = Q_1(t) + Q_2(t) \qquad z_2(t) = \lambda_2 Q_1(t) - \lambda_1 Q_2(t)$$

Then we have

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 + \lambda_2 \\ 0 \end{bmatrix} - \begin{bmatrix} \lambda_1 + \lambda_2 & \lambda_1 + \lambda_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

In particular the variable $z_2(t)$ evolves independent of the policy chosen.

Example: uncontrollable case

Some words about case $\lambda_1 = \mu_1$, $\lambda_2 = \mu_2$, i.e., R not invertible

Uncontrollable dynamics

$$\begin{bmatrix} \dot{Q}_1(t) \\ \dot{Q}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} \lambda_1 & \lambda_1 \\ \lambda_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Define change of coordinates:

$$z_1(t) = Q_1(t) + Q_2(t) \qquad z_2(t) = \lambda_2 Q_1(t) - \lambda_1 Q_2(t)$$

Then we have

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 + \lambda_2 \\ 0 \end{bmatrix} - \begin{bmatrix} \lambda_1 + \lambda_2 & \lambda_1 + \lambda_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

In particular the variable $z_2(t)$ evolves independent of the policy chosen.

Example: uncontrollable case

Some words about case $\lambda_1 = \mu_1, \lambda_2 = \mu_2$, i.e., R not invertible

Uncontrollable dynamics

$$\begin{bmatrix} \dot{Q}_1(t) \\ \dot{Q}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} - \begin{bmatrix} \lambda_1 & \lambda_1 \\ \lambda_2 & \lambda_2 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Define change of coordinates:

$$z_1(t) = Q_1(t) + Q_2(t) \qquad z_2(t) = \lambda_2 Q_1(t) - \lambda_1 Q_2(t)$$

Then we have

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 + \lambda_2 \\ 0 \end{bmatrix} - \begin{bmatrix} \lambda_1 + \lambda_2 & \lambda_1 + \lambda_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

In particular the variable $z_2(t)$ evolves independent of the policy chosen.

Example: uncontrollable case

Leonardo Rojas-Nandayapa, Tom Salisbury, Yoni Nazarathy

The push-pull network with $\lambda_1 = \mu_1$, $\lambda_2 = \mu_2$ is non-stabilizable.
Proof uses the fact that $Z(Q(t)) = \lambda_2 Q_1(t) - \lambda_1 Q_2(t)$ is a martingale

Example: controller design

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \leq 1$$

Basic idea

Decouple state from input, i.e. what does u_i control?

Define change of coordinates $z(t) = R^{-1}Q(t)$:

Transformed system

$$\dot{z}(t) = R^{-1}\alpha - u(t) = u^* - u(t) \qquad z(0) = z_0 = R^{-1}Q_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \leq 1$$

Technische Universiteit Eindhoven University of Technology

Example: controller design

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \leq 1$$

Basic idea

Decouple state from input, i.e. what does u_i control?

Define change of coordinates $z(t) = R^{-1}Q(t)$:

Transformed system

$$\dot{z}(t) = R^{-1}\alpha - u(t) = u^* - u(t) \qquad z(0) = z_0 = R^{-1}Q_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \leq 1$$

Example: controller design

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \leq 1$$

Basic idea

Decouple state from input, i.e. what does u_i control?

Define change of coordinates $z(t) = R^{-1}Q(t)$:

Transformed system

$$\dot{z}(t) = R^{-1}\alpha - u(t) = u^* - u(t) \qquad z(0) = z_0 = R^{-1}Q_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \leq 1$$

Example: controller design

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \leq 1$$

Basic idea

Decouple state from input, i.e. what does u_i control?

Define change of coordinates $z(t) = R^{-1}Q(t)$:

Transformed system

$$\dot{z}(t) = R^{-1}\alpha - u(t) = u^* - u(t) \qquad z(0) = z_0 = R^{-1}Q_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \leq 1$$

Technische Universiteit Eindhoven University of Technology

Example

Change of coordinates

$$z_1(t) = \frac{\mu_2}{\mu_1\mu_2 - \lambda_1\lambda_2} Q_1(t) - \frac{\lambda_1}{\mu_1\mu_2 - \lambda_1\lambda_2} Q_2(t)$$
$$z_2(t) = \frac{-\lambda_2}{\mu_1\mu_2 - \lambda_1\lambda_2} Q_1(t) + \frac{\mu_1}{\mu_1\mu_2 - \lambda_1\lambda_2} Q_2(t)$$

Resulting control problem

$$\dot{z}_1(t) = u_1^* - u_1(t) \qquad 0 \leq u_1(t) \leq 1$$
$$\dot{z}_2(t) = u_2^* - u_2(t) \qquad 0 \leq u_2(t) \leq 1$$

while making sure that

$$0 \leq \begin{bmatrix} \mu_1 & \lambda_1 \\ \lambda_2 & \mu_2 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix}$$

Example

Neglecting the latter constraint, the problem of controlling

$$\dot{z}_1(t) = u_1^* - u_1(t) \qquad 0 \leq u_1(t) \leq 1$$
$$\dot{z}_2(t) = u_2^* - u_2(t) \qquad 0 \leq u_2(t) \leq 1$$

becomes easy:

$$u_1(t) = \begin{cases} 1 & \text{if } z_1(t) > 0 \\ u_1^* & \text{if } z_1(t) = 0 \\ 0 & \text{if } z_1(t) < 0 \end{cases} \qquad u_2(t) = \begin{cases} 1 & \text{if } z_2(t) > 0 \\ u_2^* & \text{if } z_2(t) = 0 \\ 0 & \text{if } z_2(t) < 0 \end{cases}$$

Observations

- Above controller also solves problem with constraint
- Optimal controller for minimizing $\int_0^\infty \|z(t)\|_1 \, dt$.
- Minimal time controller

Example

Neglecting the latter constraint, the problem of controlling

$$\dot{z}_1(t) = u_1^* - u_1(t) \qquad 0 \le u_1(t) \le 1$$
$$\dot{z}_2(t) = u_2^* - u_2(t) \qquad 0 \le u_2(t) \le 1$$

becomes easy:

$$u_1(t) = \begin{cases} 1 & \text{if } z_1(t) > 0 \\ u_1^* & \text{if } z_1(t) = 0 \\ 0 & \text{if } z_1(t) < 0 \end{cases} \qquad u_2(t) = \begin{cases} 1 & \text{if } z_2(t) > 0 \\ u_2^* & \text{if } z_2(t) = 0 \\ 0 & \text{if } z_2(t) < 0 \end{cases}$$

Observations

- Above controller also solves problem with constraint
- Optimal controller for minimizing $\int_0^\infty \|z(t)\|_1 dt$.
- Minimal time controller

Technische Universiteit Eindhoven University of Technology

Example: Controller

Controller for stochastic queueing network

$$u_1(t) = \begin{cases} 1 & \text{if } \frac{\mu_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) > \frac{\lambda_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ and } Q_1(t) > 0 \\ 0 & \text{if } \frac{\mu_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) < \frac{\lambda_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ or } Q_1(t) = 0 \end{cases}$$

$$u_2(t) = \begin{cases} 1 & \text{if } \frac{\lambda_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) < \frac{\mu_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ and } Q_2(t) > 0 \\ 0 & \text{if } \frac{\lambda_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) > \frac{\mu_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ or } Q_2(t) = 0 \end{cases}$$

Lyapunov function: cost-to-go from optimal control problem

$$V(z) = \begin{cases} z_1^2/(1-u_1^*) + z_2^2/(1-u_2^*) & \text{if } z_1 \geq 0 \text{ and } z_2 \geq 0 \\ z_1^2/u_1^* + z_2^2/(1-u_2^*) & \text{if } z_1 \leq 0 \text{ and } z_2 \geq 0 \\ z_1^2/(1-u_1^*) + z_2^2/u_2^* & \text{if } z_1 \geq 0 \text{ and } z_2 \leq 0 \\ z_1^2/u_1^* + z_2^2/u_2^* & \text{if } z_1 \leq 0 \text{ and } z_2 \leq 0 \end{cases}$$

Example: Controller

Controller for stochastic queueing network

$$u_1(t) = \begin{cases} 1 & \text{if } \frac{\mu_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) > \frac{\lambda_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ and } Q_1(t) > 0 \\ 0 & \text{if } \frac{\mu_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) < \frac{\lambda_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ or } Q_1(t) = 0 \end{cases}$$

$$u_2(t) = \begin{cases} 1 & \text{if } \frac{\lambda_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) < \frac{\mu_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ and } Q_2(t) > 0 \\ 0 & \text{if } \frac{\lambda_2}{\mu_1\mu_2-\lambda_1\lambda_2} Q_1(t) > \frac{\mu_1}{\mu_1\mu_2-\lambda_1\lambda_2} Q_2(t) \text{ or } Q_2(t) = 0 \end{cases}$$

Lyapunov function: cost-to-go from optimal control problem

$$V(z) = \begin{cases} z_1^2/(1-u_1^*) + z_2^2/(1-u_2^*) & \text{if } z_1 \geq 0 \text{ and } z_2 \geq 0 \\ z_1^2/u_1^* + z_2^2/(1-u_2^*) & \text{if } z_1 \leq 0 \text{ and } z_2 \geq 0 \\ z_1^2/(1-u_1^*) + z_2^2/u_2^* & \text{if } z_1 \geq 0 \text{ and } z_2 \leq 0 \\ z_1^2/u_1^* + z_2^2/u_2^* & \text{if } z_1 \leq 0 \text{ and } z_2 \leq 0 \end{cases}$$

Technische Universiteit Eindhoven University of Technology

Controller design: general case

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Change of coordinates: $z(t) = R^{-1}Q(t)$

Transformed system

$$\dot{z}(t) = u^* - u(t) \qquad z(0) = z_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Objective

$$\min_{u(t)} \int_0^\infty \|z(t)\|_1 \, dt$$

TU/e Technische Universiteit Eindhoven University of Technology

Controller design: general case

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Change of coordinates: $z(t) = R^{-1}Q(t)$

Transformed system

$$\dot{z}(t) = u^* - u(t) \qquad z(0) = z_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Objective

$$\min_{u(t)} \int_0^\infty \|z(t)\|_1 \, dt$$

TU/e Technische Universiteit Eindhoven University of Technology

Controller design: general case

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Change of coordinates: $z(t) = R^{-1}Q(t)$

Transformed system

$$\dot{z}(t) = u^* - u(t) \qquad z(0) = z_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Objective

$$\min_{u(t)} \int_0^\infty \|z(t)\|_1 \, dt$$

TU/e Technische Universiteit Eindhoven University of Technology

Controller design: general case

System

$$\dot{Q}(t) = \alpha - Ru(t) \qquad Q(0) = Q_0$$

$$0 \leq Q(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Change of coordinates: $z(t) = R^{-1}Q(t)$

Transformed system

$$\dot{z}(t) = u^* - u(t) \qquad z(0) = z_0$$

$$0 \leq Rz(t) \qquad 0 \leq u(t) \qquad Cu(t) \leq 1$$

Objective

$$\min_{u(t)} \int_0^\infty \|z(t)\|_1 dt$$

Technische Universiteit Eindhoven University of Technology

SCLP

Let $z(t) = x^+(t) - x^-(t)$, and let T be large enough.

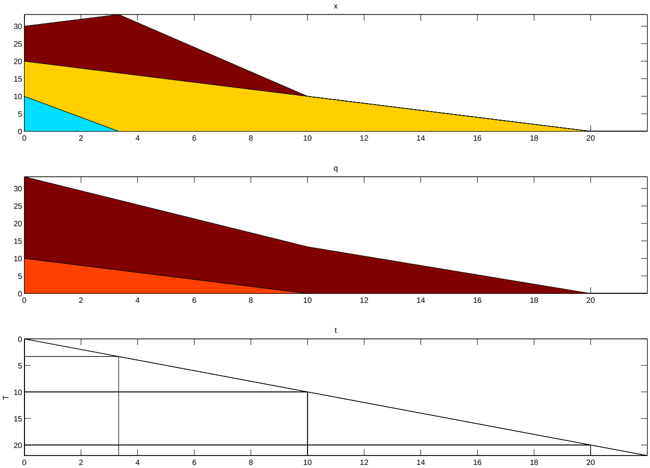
Problem

$$\max \int_0^T \mathbf{1}^\top \big((T-t)u(t) - 2x^-(t)\big)\,\mathrm{d}\,t$$
$$\text{s.t.} \int_0^t u(s)\,\mathrm{d}\,s - x^-(t) + x^+(t) = x_0 + u^* t$$
$$\int_0^t Ru(s)\,\mathrm{d}\,s + Q(t) = Q_0 + Ru^* t$$
$$Cu(t) + u^{\mathrm{IVQ}}(t) = \mathbf{1}$$
$$Q(t), x^+(t), x^-(t), u(t), u^{\mathrm{IVQ}}(t) \geq \mathbf{0}$$

Weiss, 2008

Simplex-like algorithm for solving SCLP.

mpSCLP: structure



Solution (conjecture)

Rates LP

$$\begin{aligned} \max \quad & \mathbf{1}^\top (u - 2\dot{x}^-) \\ \text{s.t.} \quad & u - \dot{x}^- + \dot{x}^+ = u^* \\ & Ru + \dot{Q} = Ru^* \\ & Cu + u^{\mathrm{IVQ}} = \mathbf{1} \end{aligned}$$

Additional constraints

If $z_i = 0$: $u_i = u_i^*$

If $Q_i = 0$: $\dot{Q}_i = 0$

Solution (conjecture)

Properties of solution

- it empties in minimum time
- it has minimum $\|z(t)\|_1$ for all t (pathwise optimality)
- if $z_i(t) < 0$ it will increase for all t until it hits 0.
- if $z_i(t) > 0$ it is monotonically non-increasing
- once $z_i(t) = 0$ it stays 0
- once $Q_i(t)$ hits zero it stays 0
- $Q_i(t)$ may not be monotonically non-increasing

Resulting policy (for stochastic MCQN-IVQ)

Modified priority discipline

- For each server: make priority list of standard queues
- For each server: at time t:
  - Remove Q_i from list if $Q_i(t) = 0$
  - Remove Q_i from list if $z_i(t) < 0$
  - If list nonempty: serve Q_i on top of list
  - If list empty: serve IVQ

Remark

In specific cases: small modifications are required to ensure $\dot{Q}_i = 0$ once $Q_i = 0$.

Resulting policy (for stochastic MCQN-IVQ)

Modified priority discipline

- For each server: make priority list of standard queues
- For each server: at time t:
  - Remove Q_i from list if $Q_i(t) = 0$
  - Remove Q_i from list if $z_i(t) < 0$
  - If list nonempty: serve Q_i on top of list
  - If list empty: serve IVQ

Remark

In specific cases: small modifications are required to ensure $\dot{Q}_i = 0$ once $Q_i = 0$.