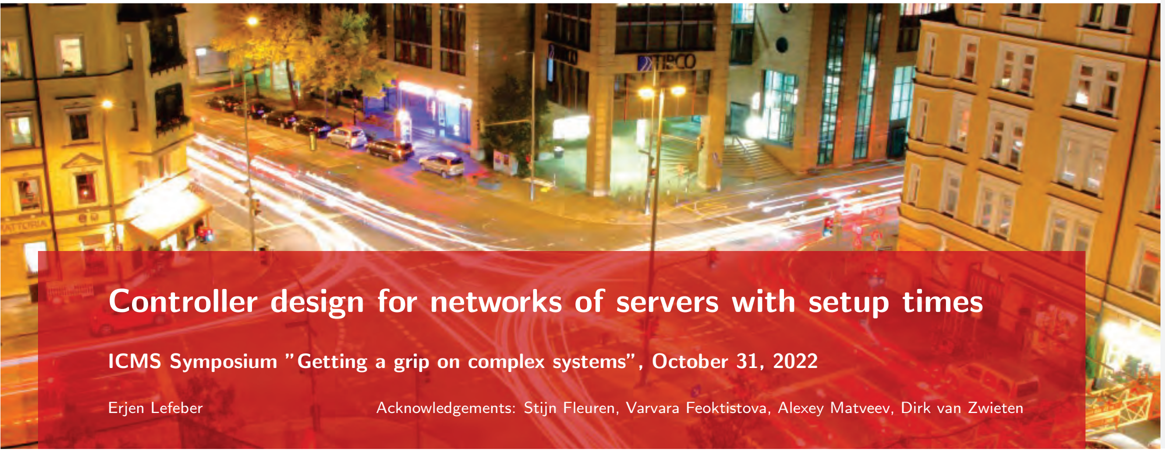


Controller design for networks of servers with setup times

ICMS Symposium "Getting a grip on complex systems", October 31, 2022

Erjen Lefeber

Acknowledgements: Stijn Fleuren, Varvara Feoktistova, Alexey Matveev, Dirk van Zwieten

Mechanical Engineering

Motivation

Problem

How to control these networks?

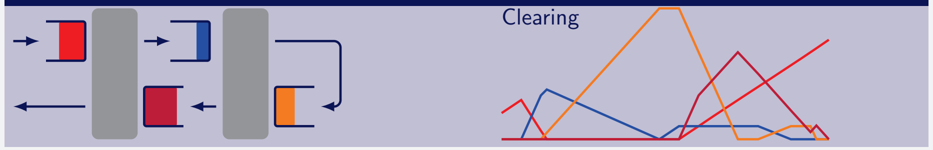
Decisions: When to switch, and to which job-type

Goals: Minimal number of jobs, minimal flow time

Current approach

Start from policy, analyze resulting dynamics

Kumar, Seidman (1990)



Problem

Current status (after three 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

Notions from control theory

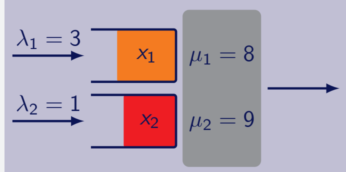
1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

Parallels with this problem

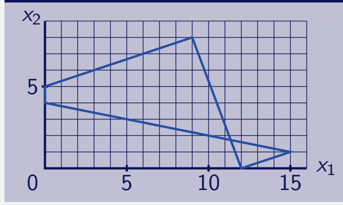
1. Determine desired system behavior
2. Derive non-distributed/centralized controller
3. Can state be reconstructed?
4. Derive distributed/decentralized controller

Problem 1: Generate feasible reference trajectory

Single machine



Optimal periodic behavior



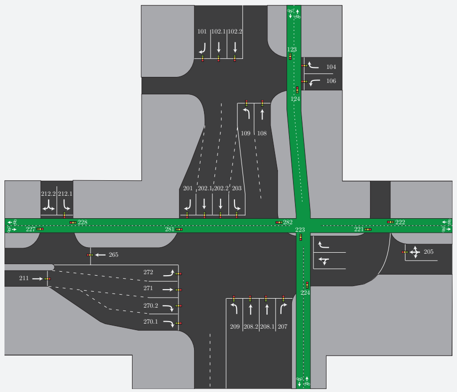
Remarks

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

Problem 1: Generate feasible reference trajectory

Question

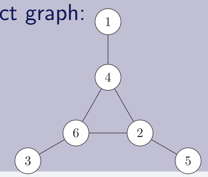
How to determine optimal periodic behavior for networks?



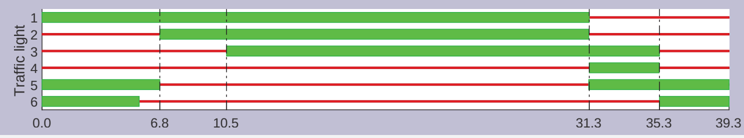
Problem 1: Generate feasible reference trajectory

Intersection

Conflict graph:

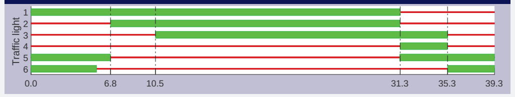


Optimal schedule (data from Sweco: A2/N279 [East; Evening])



Problem 1: Generate feasible reference trajectory

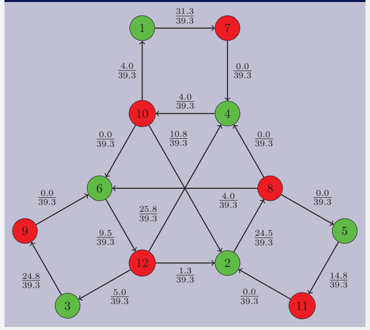
Optimal schedule / conflict graph



Event times

i	1	2	3	4	5	6
t(i)	0.0	6.8	10.5	31.3	31.3	35.3
t(i+6)	31.3	31.3	35.3	35.3	6.8	5.5
i+6	7	8	9	10	11	12

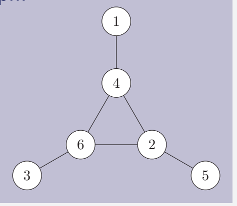
Extended graph



TU/e

Data

- Arrival rates: λ_i
- Service rates: μ_i
- Clearance times: $\sigma_{i,j}$
- Min/max green time: $g_i^{\min}$, $g_i^{\max}$.
- Min/max period: $T^{\min}$, $T^{\max}$.
- Conflict graph:



Design variables

- $x(i,j)$ fraction of period from event i to j
- $T' = 1/T$ reciprocal of duration of period

Constraints

- Stable system: $\rho_i = \lambda_i/\mu_i \leq x(i, i+n)$
- Clearance time: $\sigma_{i,j} T' \leq x(i,j)$
- Minimal/maximal green time
- Minimal/maximal period
- Conflict: $x(i,i+n)+x(i+n,j)+x(j,j+n)+x(j+n,i) = 1$
- Integer cycle: $\sum_{(i,j)\in C^+} x(i,j) - \sum_{(i,j)\in C^-} x(i,j) = z_C$.

TU/e

Objective

Minimize weighted average delay (Fluid, Webster, Miller, v.d. Broek):

$$\sum_{i=1}^{n} \frac{r_i}{2\lambda_i(1-\rho_i)T}\left(r_i\lambda_i + \frac{s_i^2}{1-\rho_i} + \frac{r_i\rho_i^2 s_i^2 T^2}{(1-\rho_i)(T-r_i)^2((1-\rho_i)T-r_i)}\right)$$

Concluding remarks for Problem 1

- Mixed integer convex optimization problem.
- Data (Sweco) of real intersection in the Netherlands with 29 directions:
  - Notebook Intel i5-4300U CPU 1.90GHZ with 16.0GB of RAM, Solver: SCIP 3.2.0
  - Standard implementation: 48 hours.
  - Our approach (plus advanced graph theoretical algorithms): 2 seconds.
- Network of intersections: (conflict) graph with components

TU/e

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

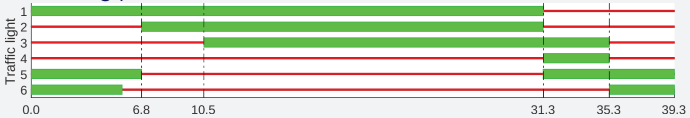
Parallels with this problem

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

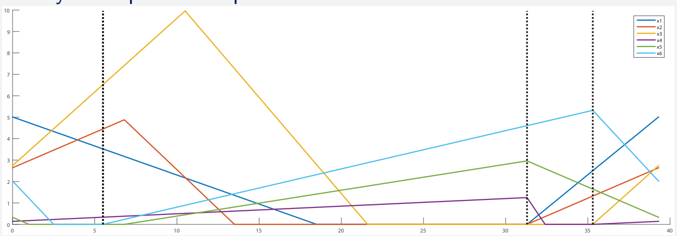
TU/e

Problem 2: Design (static) state feedback

Consider the following periodic schedule:

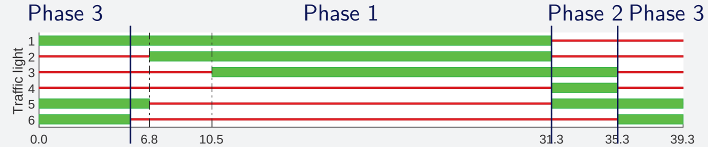


Resulting in steady state periodic wip evolution:



Problem 2: Design (static) state feedback

Steady state periodic wip evolution:



- Phase 1: directions 1, 2 and 3 served (steady state: 5.5 – 31.3)
- Phase 2: directions 3, 4 and 5 served (steady state: 31.3 – 35.3)
- Phase 3: directions 5, 6 and 1 served (steady state: 35.3 – 5.5)

NB: In Phase 1: directions 2 and 3 are served after setup,
and 5 is still served for the first 6.8-5.5=1.3 seconds.
In Phase 3: direction 1 is served after setup.

Problem 2: Design (static) state feedback

Consider non-empty buffers at start of Phase 1: $X_1 = [x_1, x_2, x_3, x_4]^T$,
and non-empty buffers at end of Phase 1: $X_2 = [x_4, x_5, x_6]^T$.

A policy (phase control rule) relates the two by means of a mapping $\mathcal{T}_1$: $\mathcal{T}_1 X_1 = X_2$.

Similarly by defining $X_3 = [x_1, x_2, x_5, x_6]^T$ we also have mappings $\mathcal{T}_2$ and $\mathcal{T}_3$.

We are interested in the monodromy operator $\mathcal{M} = \mathcal{T}_3 \circ \mathcal{T}_2 \circ \mathcal{T}_1$ (Poincaré map). Ideally we want this to have the state of the desired periodic orbit as globally attractive fixed point.

Question

How to design a policy (phase control rule) to achieve desired behavior?

NB: Typically a phase control rule results in a piecewise affine mapping $\mathcal{T}_i$.

Problem 2: Design (static) state feedback

Stability analysis of monodromy operator $\mathcal{M} = \mathcal{T}_n \circ \mathcal{T}_{n-1} \circ \cdots \circ \mathcal{T}_1$ is cumbersome.

Useful result by Feoktistova, Matveev, Lefeber, Rooda (2012)

Let $\mathcal{M}$ be an operator which:
- is piecewise affine, i.e. $\mathcal{M}x = A_i x + b_i$ for $x \in \{P_i x \leq q_i\}$,
- is continuous,
- is monotone, i.e. $A_i \geq 0$,
- is strictly dominated, i.e. $b_i > 0$,
- has a fixed point, i.e. there exists x^* such that $x^* = \mathcal{M}x^*$,

then
- the fixed point is unique, and
- attracts all solutions of $x_{k+1} = \mathcal{M}x_k$; $x_0 \in \mathbb{R}^n_+$, i.e. $\lim_{k\to\infty} x_k = x^*$.

Problem 2: Design (static) state feedback

Useful Lemma's

Composition: $\mathcal{T}_2 \circ \mathcal{T}_1 : A_2(A_1 x + b_1) + b_2 = \underbrace{A_2 A_1}_{A} x + \underbrace{A_2 b_1 + b_2}_{b}$.

- Composition of piecewise affine operators is piecewise affine.
- Composition of continuous operators is continuous.
- Composition of monotone dominated ($b_i \geq 0$) operators is monotone dominated.

Consequence

It suffices to have that

- $\mathcal{T}_i$ are piecewise affine continuous monotone dominated (not necessarily strictly dominated)
- M is strictly dominated (e.g. $\mathcal{T}_i$ for some i, or $\mathcal{T}_i \circ \mathcal{T}_{i-1}$), and has a fixed point.

Problem 2: Design (static) state feedback

Policy for single server example

- Clearing with slow-mode for Phase 1 (e.g. serve for 1 at arrival rate, or minimal phase duration of 4).
- M has fixed point provided that $\rho_1 + \rho_2 = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1$.
- NB: For arbitrary arrival rates and service rates, so robustness.

Concluding remarks

- Phase control rules determine operators. Can relatively easily be chosen to be piecewise affine continuous monotone dominated.
- Only need to show that $\mathcal{M}$ is strictly dominated and has a fixed point.
- Robustness against parameters (different inflow rates, service rates, clearance times).

Approach

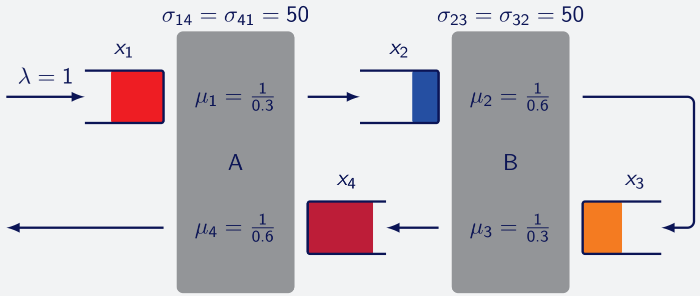
Notions from control theory

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

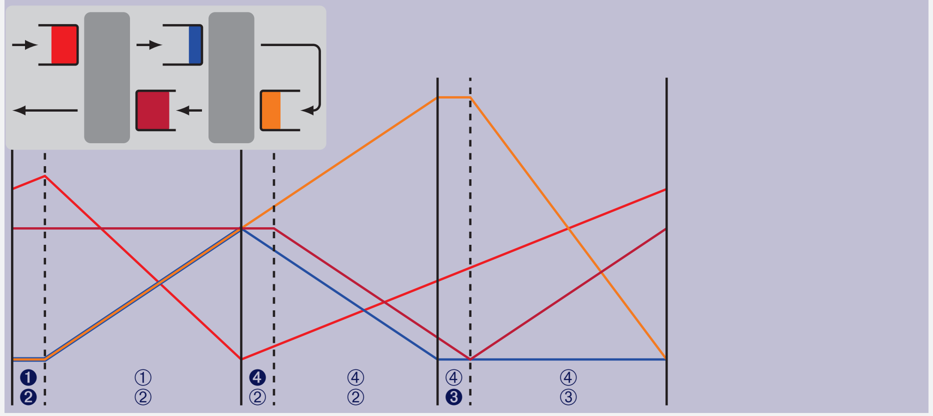
Illustration Problems 3 and 4: Kumar-Seidman



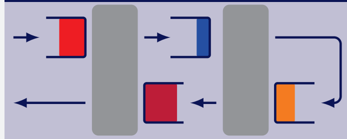
Observation

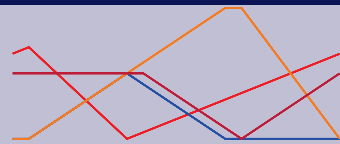
Sufficient capacity (consider period of at least 1000).

Solution to Problem 1: Desired behavior



Resulting controller (solving Problem 2)





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 $x_4 \leq 83\frac{1}{3}$

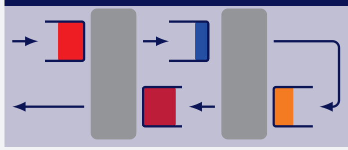
Mode (4,3): to (1,2) when $x_3 = 0$

Remarks

- Non-distributed/centralized controller
- Can be implemented using only synchronisation signals between servers

Problem 3: Reconstructing the state is possible

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

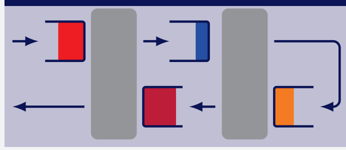
- $x_3(t_2) = 0$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\,\mathrm{d}\tau$

Observation

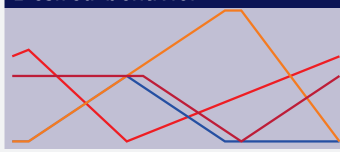
Observability determined by network topology

Solution to Problem 4: 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 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 2: Serve at least 1000 jobs until $x_2 = 0$, then switch.

Serving 3: Empty buffer, then switch.

Conclusions

Control theory inspired approach

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

Advantage

Problems can be considered separately.

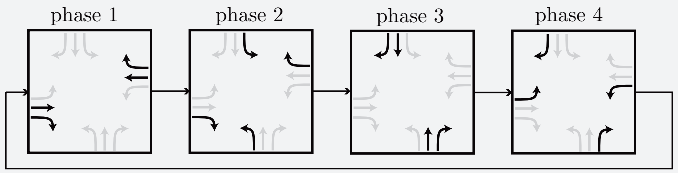
Centralized control

Can deal with: Arbitrary networks, Finite buffers, Transportation delays.

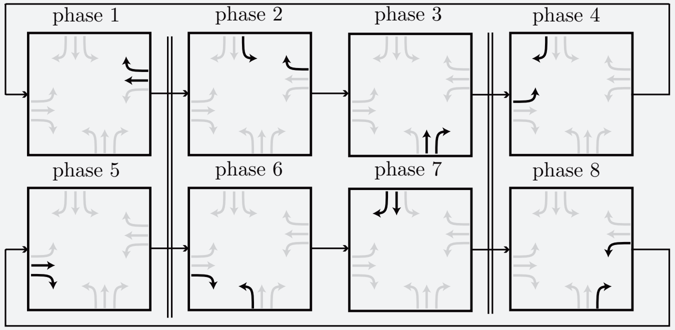
Decentralized control

- Observer based approach results in new, tailor-made controllers.

Extra: Single ring



Extra: Double ring



Extra: Phase operator

$$\mathcal{T}_1 X_1 = \begin{cases} \begin{bmatrix} \frac{\lambda_2}{\mu_1-\lambda_1} & 0 \\ \frac{\lambda_4}{\mu_1-\lambda_1} & 0 \end{bmatrix} X_1 + \begin{bmatrix} \lambda_2\delta_1 \\ \lambda_4(\delta_1-\sigma_{23}) \end{bmatrix} & \text{if } \frac{x_1}{\mu_1-\lambda_1} + \delta_1 \geq \frac{x_3+\mu_3\sigma_{23}}{\mu_3-\lambda_3} \wedge g^{\min} + \sigma_{23} - \sigma_{12} \\ \begin{bmatrix} 0 & \frac{\lambda_2}{\mu_3-\lambda_3} \\ 0 & \frac{\lambda_4}{\mu_3-\lambda_3} \end{bmatrix} X_1 + \begin{bmatrix} \frac{\lambda_2\sigma_{23}}{1-\rho_3} \\ \frac{\lambda_4\rho_3\sigma_{23}}{1-\rho_3} \end{bmatrix} & \text{if } \frac{x_3+\mu_3\sigma_{23}}{\mu_3-\lambda_3} \geq \frac{x_1}{\mu_1-\lambda_1} + \delta_1 \wedge g^{\min} + \sigma_{23} - \sigma_{12} \\ \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right] X_1 + \begin{bmatrix} \lambda_2(g^{\min}+\sigma_{23}-\sigma_{12}) \\ \lambda_4(g^{\min}-\sigma_{12}) \end{bmatrix} & \text{if } g^{\min} + \sigma_{23} - \sigma_{12} \geq \frac{x_1}{\mu_1-\lambda_1} + \delta_1 \wedge \frac{x_3+\mu_3\sigma_{23}}{\mu_3-\lambda_3} \end{cases}$$