Controller design for flow networks of switched servers with setup times and formation tracking for drones

Erjen Lefeber. Complex Friday, April 12, 2019, Eindhoven

Slides available via `https://dc.wtb.tue.nl/lefeber/`

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mechanical Engineering

Complex Friday

- Personal historic background (10 min)
- Recent work (40 min)
- Connecting with themes of GoC, starting point for future collaboration (discussion)

About Erjen Lefeber

- 1996: MSc in Applied Mathematics at University of Twente (Adaptive) control of chaotic and robot systems via bounded feedback control
- 2000: PhD in Applied Mathematics at University of Twente Tracking Control of Nonlinear Mechanical Systems
- Since 2000: Assistant Professor at TU/e: Mechanical Engineering

2000–2014	Systems Engineering Group (since 2011: Manufacturing Networks)
2015–now	Dynamics and Control

Complexity

- 2006: Organised EU Thematic Institute on Information and material flows in complex networks together with D.Armbruster, D.Helbing, A.Mikhailov.
- 2008–2013 VIDI: Controller design for flow networks of switched servers with setup times
- 2018: Organised 5th IFAC Conference on Analysis and Control of Chaotic Systems together with H.Nijmeijer, A.Pogromsky
- 2019–: At ICMS on Fridays

TU/e

Selected papers

- Tracking control of an underactuated ship / Way-point tracking control of ships
- Saturated stabilization and tracking of a nonholonomic mobile robot
- Information and material flows in complex networks
- Robust cyclic berth planning of container vessels
- Modeling and control of a manufacturing flow line using partial differential equations
- Robust optimal control of material flows in demand-driven supply networks
- Aggregate modeling of semiconductor equipment using Effective Process Times
- Designs of optimal switching feedback decentralized control policies for fluid queueing networks
- Stability of multi-class queueing networks with infinite virtual queues
- Optimal routeing in two-queue polling systems
- Design of a supervisory controller for Cooperative Intersection Control using Model Predictive Control
- A Spatial Approach to Control of Platooning Vehicles: Separating Path-Following from Tracking
- Structural Properties of Third-Party Logistics Networks
- Almost global decentralised formation tracking for multiple distinct UAVs

Controller design for flow networks of switched servers with setup times

Motivation

Problem

How to control these networks?

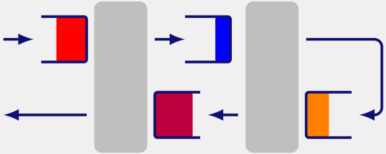
Decisions: When to switch, and to which job-type

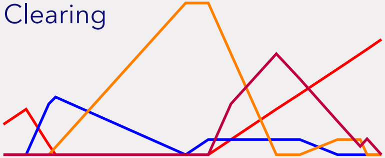
Goals: Minimal number of jobs, minimal flow time

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

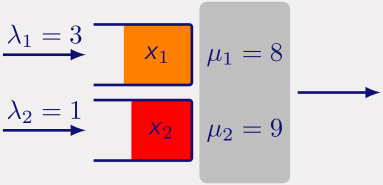
1. Generate feasible reference trajectory
2. Design (static) state feedback controller
3. Design observer
4. Design (dynamic) output feedback controller

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 reference

Single machine

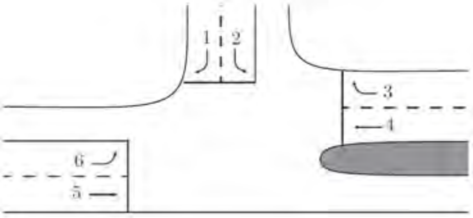


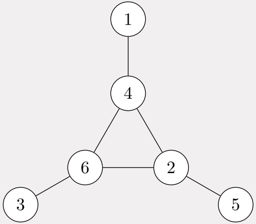
Optimal behavior

Remarks

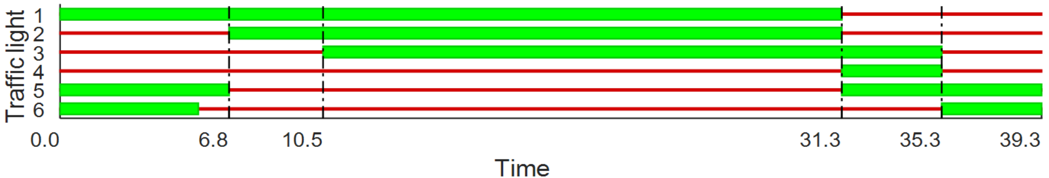
- Many existing policies assume non-idling a-priori
- Slow-mode optimal if $(\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2}) + (\lambda_2 - \lambda_1)(1 - \frac{\lambda_2}{\mu_2}) < 0.$
- Trade-off in wasting capacity: idle $\Leftrightarrow$ switch more often

Intersection



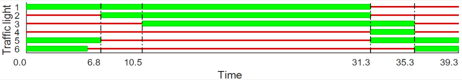


Optimal schedule (data from Grontmij (currently: Sweco): A2/N279)

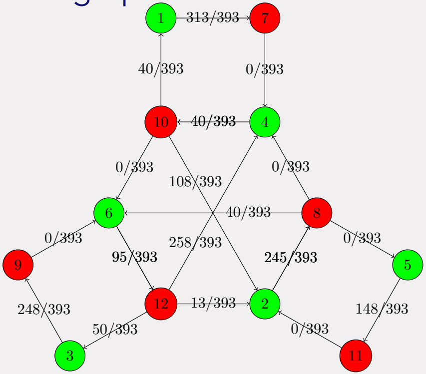


TU/e

Optimal schedule



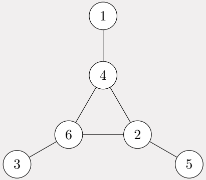
Extended graph



Event times

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

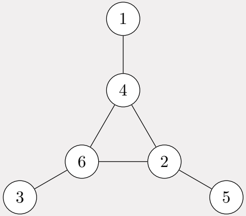
Conflict graph



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:
  $\sum_{(i,j)\in C+} x(i,j) - \sum_{(i,j)\in C-} x(i,j) = z_C$.

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 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

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

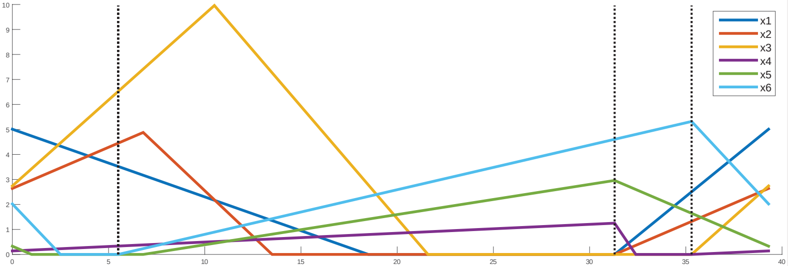
Parallels with this problem

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

Problem 2: Feedback design

Consider the following periodic schedule for a T-crossing:

Resulting steady state periodic wip evolution:



Problem 2: Feedback design

Steady state periodic wip evolution:

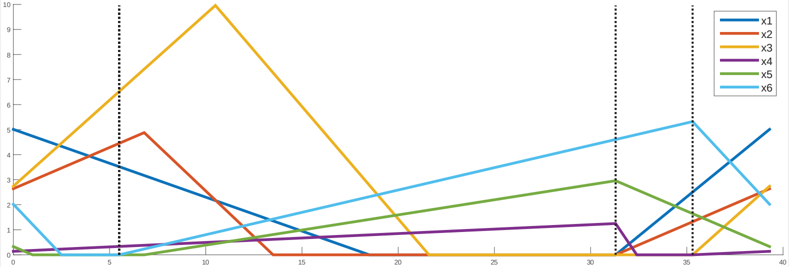
- Mode 1: directions 1, 2 and 3 served (steady state: 5.5 – 31.3)
- Mode 2: directions 3, 4 and 5 served (steady state: 31.3 – 35.3)
- Mode 3: directions 5, 6 and 1 served (steady state: 35.3 – 5.5)

NB: Some issues with properly defining modes:

- In mode 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 mode 2: direction 1 is served after setup.

Problem 2: Feedback design

Steady state periodic wip evolution:



- Define δ_i as duration of $x_i^* = 0$ for $i = 1, 2, 4, 6$,
- Define $\delta_3 + 4$ as duration of $x_3^* = 0$, define $\delta_5 + 1.3$ as duration of $x_5^* = 0$,
- Define $\theta_1 = x_1^*(5.5)/[x_1^*(35.3) + 4\lambda_1]$, define $\theta_5 = x_5^*(31.3)/x_5^*(35.3)$.

Problem 2: Feedback design

State at start of mode 1: $X = [x_4, x_3, x_2, x_1]^T(t_0)$.

State at end of mode 1: $\mathcal{T}_1 X = [x_6, x_5, x_4]^T(t_1)$.

Stay in mode until $x_1(t_1 - \delta_1) = 0$, $x_2(t_1 - \delta_2) = 0$ and $x_3(t_1 - \delta_3) = 0$.

$$\mathcal{T}_1 X = \begin{cases} \begin{bmatrix} 0 & 0 & 0 & \frac{\lambda_6}{\mu_1 - \lambda_1} \\ 0 & 0 & 0 & \frac{\lambda_5}{\mu_1 - \lambda_1} \\ 1 & 0 & 0 & \frac{\lambda_4}{\mu_1 - \lambda_1} \end{bmatrix} X + \begin{bmatrix} \lambda_6 \delta_1 \\ \lambda_5 \delta_1 \\ \lambda_4 \delta_1 \end{bmatrix} & \text{if } \frac{x_1(t_0)}{\mu_1 - \lambda_1} + \delta_1 \geq \frac{x_2(t_0)}{\mu_2 - \lambda_2} + \delta_2 \vee \frac{x_3(t_0)}{\mu_3 - \lambda_3} + \delta_3 \\ \begin{bmatrix} 0 & 0 & \frac{\lambda_6}{\mu_2 - \lambda_2} & 0 \\ 0 & 0 & \frac{\lambda_5}{\mu_2 - \lambda_2} & 0 \\ 1 & 0 & \frac{\lambda_4}{\mu_2 - \lambda_2} & 0 \end{bmatrix} X + \begin{bmatrix} \lambda_6 \delta_2 \\ \lambda_5 \delta_2 \\ \lambda_4 \delta_2 \end{bmatrix} & \text{if } \frac{x_2(t_0)}{\mu_2 - \lambda_2} + \delta_2 \geq \frac{x_1(t_0)}{\mu_1 - \lambda_1} + \delta_1 \vee \frac{x_3(t_0)}{\mu_3 - \lambda_3} + \delta_3 \\ \begin{bmatrix} 0 & \frac{\lambda_6}{\mu_3 - \lambda_3} & 0 & 0 \\ 0 & \frac{\lambda_5}{\mu_3 - \lambda_3} & 0 & 0 \\ 1 & \frac{\lambda_4}{\mu_3 - \lambda_3} & 0 & 0 \end{bmatrix} X + \begin{bmatrix} \lambda_6 \delta_3 \\ \lambda_5 \delta_3 \\ \lambda_4 \delta_3 \end{bmatrix} & \text{if } \frac{x_3(t_0)}{\mu_3 - \lambda_3} + \delta_3 \geq \frac{x_1(t_0)}{\mu_1 - \lambda_1} + \delta_1 \vee \frac{x_2(t_0)}{\mu_2 - \lambda_2} + \delta_2 \end{cases}$$

Problem 2: Feedback design

Similarly, $\mathcal{T}_2 X$ and $\mathcal{T}_3 X$ can be determined. We need to investigate stability of the monodromy operator $M = \mathcal{T}_3 \circ \mathcal{T}_2 \circ \mathcal{T}_1$ (or $M = \mathcal{T}_2 \circ \mathcal{T}_1 \circ \mathcal{T}_3$, or $M = \mathcal{T}_1 \circ \mathcal{T}_3 \circ \mathcal{T}_2$).

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

Let $\mathcal{T}$ be an operator which:

- is piecewise affine, i.e. $\mathcal{T}x = A_i x + b_i$ for $x \in \{x \in \mathbb{R}_+^{n_i} \mid P_i x \leq q_i\}$,
- 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 $x^* = \mathcal{T}x^*$,

then

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

Problem 2: Feedback design

Observation

Since $\mathcal{T}_3$ is strictly dominated, we have $M = \mathcal{T}_3 \circ \mathcal{T}_2 \circ \mathcal{T}_1$ is strictly dominated.
Furthermore, the desired periodic behavior is a fixed point.
Therefore, global convergence towards desired periodic behavior.

Even more

Under conditions such as (show only 2 of 18 expressions):

$$(1-\rho_1)(1-\rho_3)(1-\rho_5) > \rho_1\rho_3\rho_5(1-\theta_1)(1-\theta_5)$$
$$(1-\rho_1)(1-\rho_2)(1-\rho_5) > \rho_2\rho_5(1-\theta_5)(1-\rho_1\theta_1)$$

we can show that a fixed point for M exists.

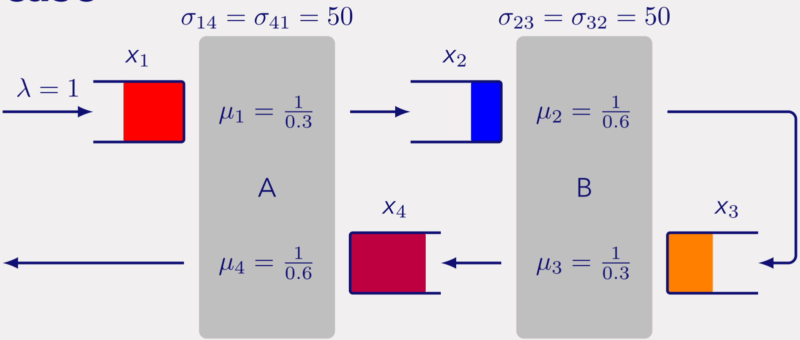
This guarantees robustness against changes in parameters.

Problem 2: Feedback design

Concluding remarks

- Switching rules determine (mode)operators. Can relatively easily be chosen to be piecewise affine continuous monotone dominated
- Only need to show that M is strictly dominated (chose initial/final mode cleverly) and has a fixed point.
- Robustness against parameters (only requirement on parameters: existence of fixed point).

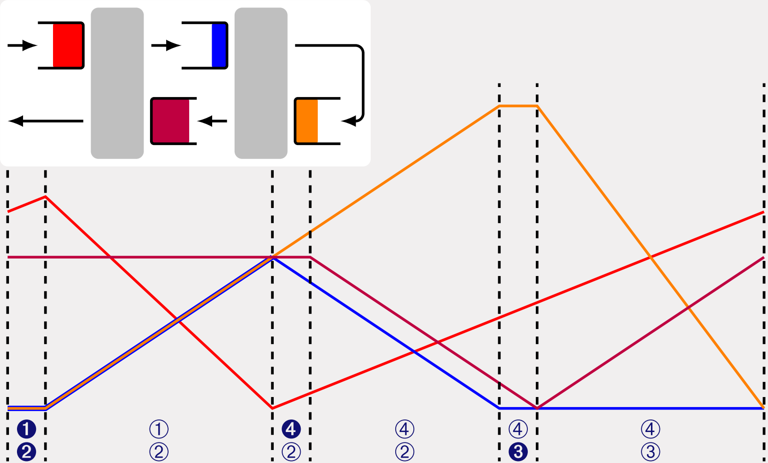
Illustration problems 3 and 4: Kumar-Seidman case



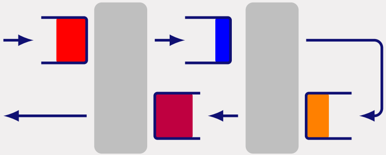
Observation

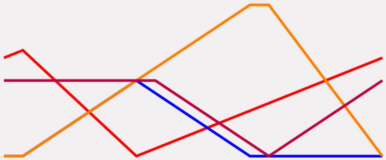
Sufficient capacity (consider period of at least 1000).

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

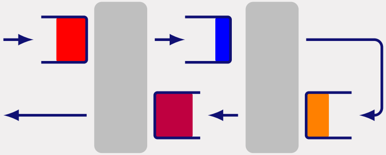
Mode (4,3): to (1,2) when $x_3 = 0$

Remarks

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

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

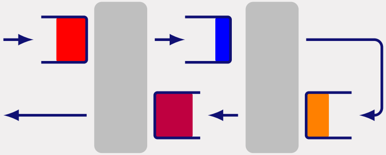
- $x_3(t_2) = 0$, and $x_3(t_1 - 50) = x_3(t_1) = (t_2 - t_1)/0.6$
- $x_2(t_1 - 50) = 0$, and $x_2(t_2) = \int_{t_1-50}^{t_2} u_1(\tau)\,\mathrm{d}\,\tau$

Observation

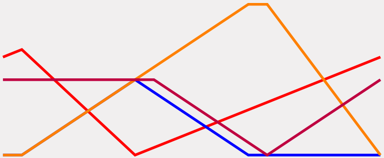
Observablity determined by network topology

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 (flow network of switched servers)

Control theory inspired approach

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

Advantage

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

Formation tracking for drones

Drone dynamics

Dynamics of drone i and its (formation) reference dynamics:

$$\begin{aligned} \dot{\rho}_i &= R_i \nu_i & \dot{\rho}_{r,i} &= R_{r,i} \nu_{r,i} \\ \dot{\nu}_i &= -S(\omega_i)\nu_i + gR_i^T e_3 - \frac{f_i}{m_i} e_3 & \dot{\nu}_{r,i} &= -S(\omega_{r,i})\nu_{r,i} + gR_{r,i}^T e_3 - \frac{f_{r,i}}{m_i} e_3 \\ \dot{R}_i &= R_i S(\omega_i) & \dot{R}_{r,i} &= R_{r,i} S(\omega_{r,i}) \\ J_i \dot{\omega}_i &= S(J_i \omega_i)\omega_i + \tau_i & J_i \dot{\omega}_{r,i} &= S(J_i \omega_{r,i})\omega_{r,i} + \tau_{r,i} \end{aligned}$$

We have formation frame $\mathcal{F}$ with position ρ_f and orientation $R_f \in \mathrm{SO}(3)$.

Reference position expressed in formation frame: $p_{r,i} = R_f^T(\rho_{r,i} - \rho_f)$.

Similarly: (individual) position and velocity tracking errors expressed in the formation frame $\mathcal{F}$:

$$\begin{aligned} p_{e,i} &= p_{r,i} - p_i = R_f^T(\rho_{r,i} - \rho_i) \\ v_{e,i} &= R_f^T(R_{r,i}\nu_{r,i} - R_i \nu_i). \end{aligned}$$

Communication topology

- Let $a_{ij} = a_{ji} = 1$ when drone i and j exchange position information, and 0 otherwise.
- Let $b_{ij} = b_{ji} = 1$ when drone i and j exchange velocity information, and 0 otherwise.

NB: Usually we have $a_{ij} = b_{ij}$, but this is not required.

Assumption 1: The (undirected) graphs with incidence matrices $A = [a_{ij}]$, respectively $B = [b_{ij}]$ are connected.

Assumption 2: The angular velocity of the formation frame, ω_f, as well as $\dot{\omega}_f$ are bounded.

Problem 1

Formation forming: Mutual synchronisation

All drones have the SAME formation tracking error, i.e., for all $i, j \in \{1, 2, \dots, n\}$:

$$\lim_{t \to \infty} \|p_{e,i}(t) - p_{e,j}(t)\| = 0 \qquad \lim_{t \to \infty} \|v_{e,i}(t) - v_{e,j}(t)\| = 0.$$

Problem 2

Formation tracking

All drones have ZERO formation tracking error, i.e., for all $i, j \in \{1, 2, \ldots, n\}$:

$$\lim_{t\to\infty} \|p_{e,i}(t)\| = \lim_{t\to\infty} \|v_{e,i}(t)\| = 0,$$

and the reference attitude is also tracked, i.e. for errors $R_{e,i} = R_i^T R_{r,i}$, $\omega_{e,i} = \omega_{r,i} - R_{e,i}^T \omega_i$:

$$\lim_{t\to\infty} R_{e,i} = I_3 \qquad \lim_{t\to\infty} \omega_{e,i} = 0.$$

Results I

Consider virtual input $u_{e,i} = -R_f^T(\frac{f_{r,i}}{m_i}R_{r,i} - \frac{f_i}{m_i}R_i)e_3$, and define

$$u_{e,i} = -\sum_{j=1}^{n}\left[a_{ij}\sigma_{p_{ij}}(p_{e,i} - p_{e,j}) + b_{ij}\sigma_{v_{ij}}(v_{e,i} - v_{e,j})\right] - \alpha_i\sigma_{p_i}(p_{e,i}) - \beta_i\sigma_{v_i}(v_{e,i}).$$

Then we can prove the following two propositions:

For $\sigma_{p_{ij}}(x) = \sigma_{p_{ji}}(x)$, $\alpha_i = 0$, and $\beta_i = 0$ the resulting closed-loop system with states $p_{e,i} - p_{e,j}$ and $v_{e,i} - v_{e,j}$ is UGAS, i.e., we solve Problem 1.

For $\sigma_{p_{ij}}(x) = \sigma_{p_{ji}}(x)$, $\alpha_i \geq 0$, $\beta_i \geq 0$, $\sum_{i=1}^{n}\alpha_i > 0$, and $\sum_{i=1}^{n}\beta_i > 0$ the resulting closed-loop system with states $p_{e,i}$ and $v_{e,i}$ is UGAS and we solve (the first part of) Problem 2.

Results II

Next step: we need to realize virtual input with actual inputs.

- Determine desired direction of thrust vector
- Determine corresponding desired attitude
- Calculate corresponding desired angular velocities and desired torques.
- Stabilize drone towards desired trajectory
- Prove convergence towards the desired trajectory, and show convergence to reference trajectory.

Simulation results (no cooperation)

Simulation results (cooperation)

Simulation results (more complex movement)

Connecting with themes of GoC, starting point for future collaboration (discussion)

- Program 1: Networks
- Program 2: Dynamical systems and control
- Program 3: Engineering complexity

Program 1: Networks

The goal is to further develop mathematical network theory and advance network science, and to apply novel theories for understanding and designing complex networks, including biological networks, molecular networks, quantum networks, communication networks, social networks and the brain.

- Traffic control (mobility and logistics) would be suitable bridge-builder between "Networks" and "Dynamical Systems and Control"
  - relatively easy dynamics
  - networks have fixed topology, but particles are free to move
  - challenges
    - from centralized/global policy to decentralized/local policy
    - move away from fixed sequence of modes
    - guarantee network stability
- My favorite question: Why that policy?
- …

Program 2: Dynamical systems and control

The goal is to analyze, control and optimize the dynamics of complex systems by designing new theoretical and computational modeling methodologies.

- Challenges in formation control of drones
  - Use only position measurements in feedback
  - Make drones cooperatively accomplish tasks (e.g. lifting and moving object together)
- …

Program 3: Engineering complexity

The goal is to design and develop new biological computing programs in mammalian cells using a combination of molecular biology and dynamical system theory and investigate their use in diverse cell-therapy applications.

- …