Aalto University
School of Electrical
Engineering

ELEC-E8121: Networked Control of Multi-agent Systems

Communication structure design

Dominik Baumann

Office number: 2569, Maarintie 8
dominik.baumann@aalto.fi

Mar. 26, 2025

In the previous lecture…

We

- Described the synchronization problem
- Designed synchronizing controllers
- Derived the synchronization condition
- Analyzed whether or not agents are synchronizable

Follow-up from last week

- In the last lecture, we defined the Laplacian as

$$L = \begin{pmatrix} \ell_{11} & \ell_{12} \\ \ell_{21} & L_{22} \end{pmatrix}$$

- I then claimed that it can be shown that the eigenvalues of $\tilde{L}_{22} = L_{22} - \mathbb{1}\ell_{12}$ are equal to the eigenvalues $\lambda_2(L), \ldots, \lambda_{\mathrm{N}}(L)$
- I have uploaded a new version of the slides from Lecture 5 where I added a proof of this in the appendix
- We can go through this proof at the end of this lecture for those who are interested

Learning outcomes

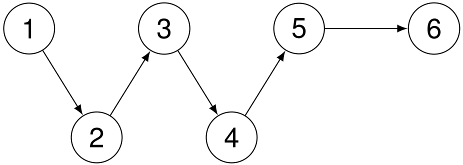
By the end of this lecture, you should be able to

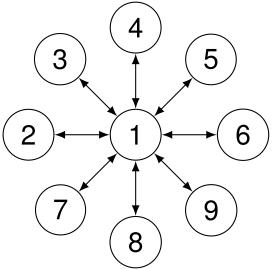
- Describe the problem of designing communication structures for networked controllers
- Design communication structures for networked controllers
- Describe design objectives of controllers for vehicle platooning
- Name conditions for controllers and communication structures that meet these objectives

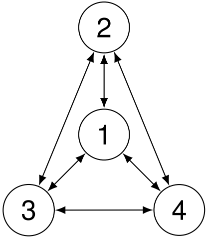
Reference

- Chapter 5 of J. Lunze. *Networked Control of Multi-Agent Systems*. Edition MoRa, 2019

Introduction







- In a networked system, we may have various communication structures
- We know that for synchronization and consensus, we need a spanning tree
- In this lecture, we discuss more in detail what communication structure is required to ensure a satisfactory overall system behavior
- We here consider leader-follower problems and platooning as a concrete example

General goals of our control design

- G1 **Stability.** The overall system should be asymptotically stable
- G2 **Asymptotic set-point following.** All agents should asymptotically reach the reference point (set by the leader)
- G3 **Quick transient behavior.** The transient behavior should be as short as possible

Problem

How does the behavior of the agents within the overall system depend upon the agent dynamics and the communication structure?

- Just trying out possible solutions is challenging:
  - Number of possible communication structures increases exponentially with the number of agents
  - The dynamical order of the overall system increases linearly with the number of agents
  - Also potential scenarios like combinations of initial states and set-points increases with the number of agents

$\rightarrow$ We need a more principled approach

Principled solution

- When there are many parameters, we often try to cast problems as optimization problems
- In this problem, we would have integer variables (is there a communication link or not), and continuous variables representing the local controllers of agents

$\rightarrow$ The complexity of such mixed-integer programming problems again increases exponentially with the number of agents...

- To circumvent this, we introduce an abstract model of the multi-agent system
- We have controlled agents $\bar{\Sigma}^i$, consisting of the agent P^i and a local controller K^i
- Each agent introduces a delay (Agent 1 sends information at time k, Agent 2 receives it at time $k+1$, Agent 3 at time $k+2, \dots$)
- We can then represent the model as a directed graph where the delay measures are the labels on the edges
- We can then rephrase the problem as searching the shortest path in a weighted directed graph

High-level algorithm

Input $P^i = (\Phi^i, \Gamma^i, C^i, K^i)$, $i = 0, 1, \ldots, N$, Goals **G1**, **G2**, **G3**

Step 1 **Design of local controllers.** Design the local controllers K^i such that the controlled agents $\bar{\Sigma}^i$ achieve **G1** and **G2**

Step 2 **Communication structure design.** Determine the communication structure $A_{\mathcal{G}}$ such that the overall system achieves **G3**

Output Networked controller

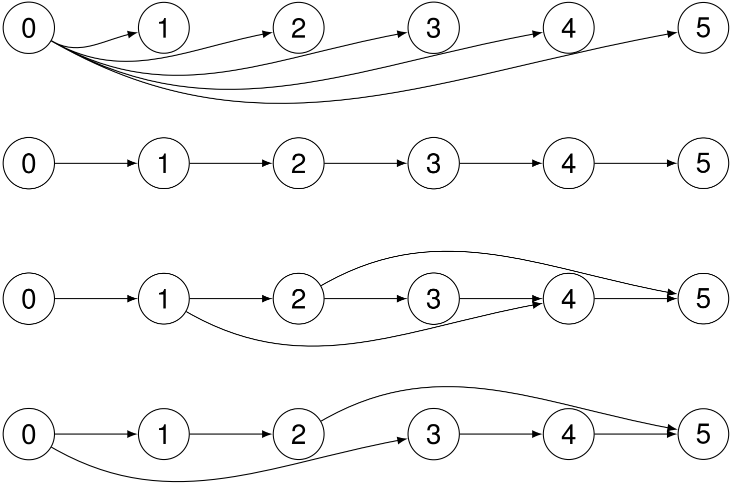
Communication structure

- As before, we represent the communication structure with a directed graph $G = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$
- We define Node 0 to be the leader P^0
- We assume a normalized adjacency matrix $\hat{A}_{\mathcal{G}}$, i.e.,

$$\sum_{j=0, j\neq i}^{N} \hat{a}_{ij} = 1, \quad i = 1, 2, \ldots, N$$

- We consider cycle-free graphs
- We assume that the agents are ordered such that $\hat{A}_{\mathcal{G}}$ is lower triangular (i.e., $\hat{a}_{ij} = 0$ if $i \leq j$)
- We further require that a spanning tree exists, which implies $\hat{a}_{10} = 1$ and for any i, we have some j such that $\hat{a}_{ij} \neq 0$

Which of the following is the best communication structure?



Model

- In general, we adopt the same model as we did before,

$$x^i_{k+1} = \Phi^i x^i_k + \Gamma^i u^i_k$$
$$y^i_k = C^i x^i_k$$

- We could now again use a proportional controller

$\rightarrow$ We already saw that this does not work for the platooning example

- Let's go directly for the PI-controller

$$u^i_k = -K^i_{\mathrm{I}} \sum_{\kappa=0}^{K} (y^i_\kappa - y^{\mathrm{s},i}_\kappa) - K^i_{\mathrm{P}} (y^i_k - y^{\mathrm{s},i}),$$

where

$$y^{\mathrm{s},i}_k = \sum_{j=0}^{i-1} \hat{a}_{ij} y^j_k$$

Unified model

- Defining $e_k^i := y_k^{\mathrm{s},i} - y_k^i$, we can rewrite this model as

$$\bar{x}_{k+1}^i = \underbrace{\begin{pmatrix} \Phi^i & K_{\mathrm{I}}^i \Gamma^i \\ 0 & 0 \end{pmatrix}}_{\Phi^{0,i}} \bar{x}_k^i + \underbrace{\begin{pmatrix} K_{\mathrm{P}}^i \Gamma^i \\ 1 \end{pmatrix}}_{\Gamma^{0,i}} e_k^i$$

$$y_k^i = \underbrace{\begin{pmatrix} C^i & 0 \end{pmatrix}}_{C^{0,i}} \bar{x}_k^i$$

- Here, we defined the extended state

$$\bar{x}_k^i = \begin{pmatrix} x_k^i \\ \sum_{\kappa=0}^{k} e_\kappa^i \end{pmatrix}$$

Model in terms of the reference trajectory

- For analysis purposes, it is convenient to rewrite the model in terms of the reference trajectory
- We had the equation

$$\bar{x}^i_{k+1} = \Phi^{0,i}\bar{x}^i_k + \Gamma^{0,i}e^i_k$$

- Inserting $e^i_k = y^{\mathrm{s},i}_k - y^i_k = y^{\mathrm{s},i}_k - C^{0,i}\bar{x}^i_k$, we get

$$\bar{x}^i_{k+1} = \underbrace{\left(\Phi^{0,i} - \Gamma^{0,i}C^{0,i}\right)}_{\bar{\Phi}^i}\bar{x}^i_k + \underbrace{\Gamma^{0,i}}_{\bar{\Gamma}^i}y^{\mathrm{s},i}_k$$

$$y^i_k = \underbrace{C^{0,i}}_{\bar{C}^i}\bar{x}^i_k$$

Set-point following

- Let us assume that the local controllers are designed such that the controlled agents are stable and have integrator dynamics

Lemma: set-point following in cycle-free networked systems

Consider controlled agents $\bar{\Sigma}^i$ under the assumption above. The overall system satisfies the control goals **G1** and **G2** if the communication graph $\mathcal{G}$ is cycle-free and includes a spanning tree with root node 0.

Proof by induction

- The first agent directly gets the reference trajectory and is asymptotically stable, i.e., we have $\lim_{k\to\infty}\left|y_k^1 - y_k^{\mathrm{ref}}\right| = 0$
- Assume this holds for the first n agents. We need to prove it also holds for $n+1$
- We assumed a lower triangular adjacency matrix. Thus,

$$y_k^{s,n+1} = \sum_{j=1}^{n} \hat{a}_{n+1,j} y_k^j + \hat{a}_{n+1,0} y_k^{\mathrm{ref}}$$

- Because of the induction assumption, all agents until n are synchronized. Thus,

$$\begin{aligned}\lim_{k\to\infty}\left|y_k^{s,n+1} - y_k^{\mathrm{ref}}\right| &= \lim_{k\to\infty}\left|\sum_{j=0}^{n} \hat{a}_{n+1,j} y_k^{\mathrm{ref}} - y_k^{\mathrm{ref}}\right| \\ &= \lim_{k\to\infty}\left|\left(\sum_{j=0}^{n} \hat{a}_{n+1,j} - 1\right) y_k^{\mathrm{ref}}\right| = 0\end{aligned}$$

- This holds because the adjacency matrix is normalized, i.e., rows sum to 1

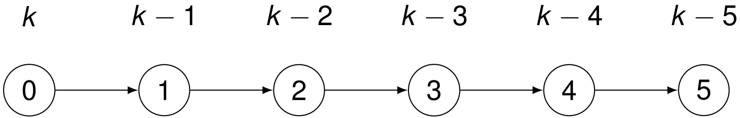
Implications

- The lemma has two important consequences
  - The controlled agents need to be asymptotically stable and have integrator dynamics. This we can achieve for each agent individually using classical controller design methods
  - Under this condition, any communication structure fulfills **G1** and **G2** as long as the graph is cycle-free and has a spanning tree

$\rightarrow$ We still have a lot of freedom in how to design the communication structure

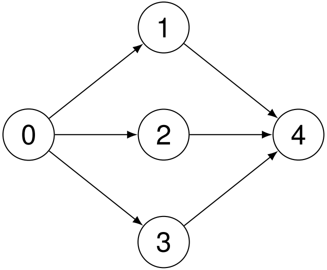
- This freedom we will need to satisfy **G3**

Delays: series connection



- Transferring information in a networked system introduces a delay
- For a series connection, the delay increases by 1 for each node
- The delay corresponds to the time constant of the physical system and could also include communication delay
- Defining the delay Δ as the time $k \to k+1$, we can say that for N agents in series, the delay until the last agent learns about the reference trajectory is $N\Delta$

Delays: parallel connection

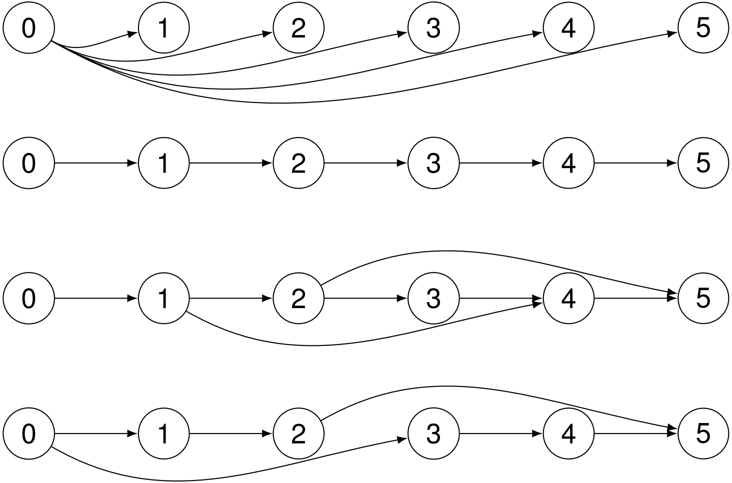


- Parallel connections are a bit more tricky
- We have defined $y_k^{\mathrm{s},i} = \sum_{j=0}^{i-1} \hat{a}_{ij} y_k^j$
- In the example, Node 4 computes $y_k^{\mathrm{s},i}$ as a weighted sum of its inputs

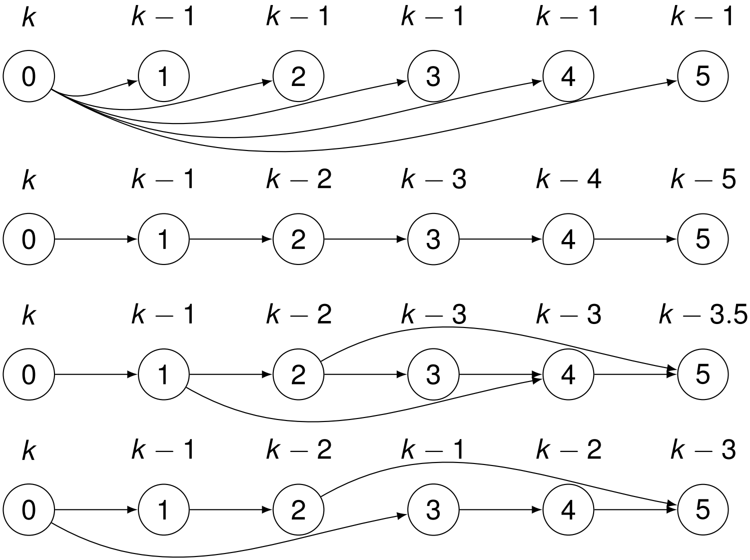
$\rightarrow$ Also the delay gets weighted by $\hat{a}_{ij}$

$\rightarrow$ In general, the total delay at Node i for a parallel connection is $\Delta^i + \sum_{j=1}^{N} \hat{a}_{ij} \Delta_{\mathrm{ref}}^j$, where Δ_{ref}^j is the delay with which Node j receives the reference signal

Exercise: compute the delays for each node



Solution



k
$k-1$
$k-1$
$k-1$
$k-1$
$k-1$
0
1
2
3
4
5
k
$k-1$
$k-2$
$k-3$
$k-4$
$k-5$
0
1
2
3
4
5
k
$k-1$
$k-2$
$k-3$
$k-3$
$k-3.5$
0
1
2
3
4
5
k
$k-1$
$k-2$
$k-1$
$k-2$
$k-3$
0
1
2
3
4
5

Communication structure design

- We can now collect the individual and cumulative delays in vectors

$$\Delta_{\mathrm{cum}} = \begin{pmatrix} \Delta_{\mathrm{ref}}^{1} \\ \Delta_{\mathrm{ref}}^{2} \\ \vdots \\ \Delta_{\mathrm{ref}}^{N} \end{pmatrix}, \quad \Delta_{\mathrm{ind}} = \begin{pmatrix} \Delta^{1} \\ \Delta^{2} \\ \vdots \\ \Delta^{N} \end{pmatrix}$$

- Defining $\hat{A}_{\mathcal{G}\mathrm{F}}$ as the normalized adjacency matrix where we deleted the row for the leader (as the leader does not have any delay), we can write the delay as

$$\begin{aligned} \Delta_{\mathrm{cum}} &= \hat{A}_{\mathcal{G}\mathrm{F}} \Delta_{\mathrm{cum}} + \Delta_{\mathrm{ind}} \\ &= (I - \hat{A}_{\mathrm{GF}})^{-1} \Delta_{\mathrm{ind}} \end{aligned}$$

- Now, we can formulate the communication structure design as an optimization problem, e.g., minimize the overall delay in the network:

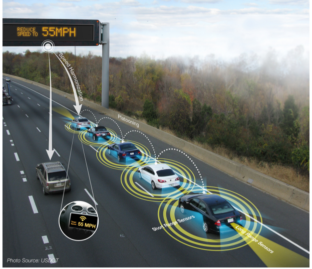
$$\min J(\hat{A}_{\mathcal{G}\mathrm{F}}) = \min \sum \Delta_{\mathrm{cum}} = \min \mathbb{1}^{\mathrm{T}} (I - \hat{A}_{\mathrm{GF}})^{-1} \Delta_{\mathrm{ind}}$$

Solving the optimization problem

- In the simplest case, we could solve the minimization problem by letting the leader directly communicate with all agents
- In practice, we may now need to include constraints (certain agents may not be able to communicate with each other for technical reasons)
- Then, we are left with a *static* optimization problem
- The dynamics of the agents are not part of the optimization

→ This drastically reduces the complexity

Example: distance control of vehicle platoons



- Modern vehicles often have adaptive cruise controllers (ACC)
- Should keep distance between vehicles in a platoon constant and avoid collisions in (ideally) all possible traffic situations
- Measurements are typically distance to and potentially velocity of the preceding vehicle
- By introducing car-to-car communication, we can extend this to cooperative adaptive cruise controllers (CACC)

Problem setting

- We have the controlled vehicles V^i, consisting of the vehicles P^i and the local controllers K^i
- All vehicles have a local reference velocity $v_k^{\mathrm{s},i}$ as input and measure their current velocity v_k^i
- An additional output for each vehicle is the distance d_k^i from its predecessor
- For ACC, the reference signal is always the velocity of the preceding vehicle, $v_k^{\mathrm{s},i} = v_k^{i-1}$, where $v_k^{\mathrm{s},0} = v_k^{\mathrm{ref}}$
- For CACC, the local reference depends on the communication structure:

$$v_k^{\mathrm{s},i} = \sum_{j=0}^{i-1} \hat{a}_{ij} v_k^j$$

$\rightarrow$ We could interpret ACC as a networked controller where vehicles can only communicate with their physical neighbors

Performance requirements for platooning

R1 Stability of the overall system

- For this, we need that each individual vehicle is stable

R2 Asymptotic synchronization

$$\lim_{k\to\infty} \left| v_k^{\mathrm{ref}} - v_k^i \right| = 0, \quad i = 0, 1, \ldots, N$$

R3 Time-headway spacing (inter-vehicle distance should increase if velocity increases)

$$d_k^{\mathrm{ref},i} = \alpha^i + \beta^i v_k^i, \quad i = 1, 2, \ldots, N$$

R4 Collision avoidance

$$d_k^i \geq \alpha^i \quad \forall k \geq 0, \quad i = 1, 2, \ldots, N$$

R5 Continuous progression (no backward driving)

$$v_k^i \geq 0 \quad \forall k \geq 0, \quad i = 0, 1, \ldots, N$$

Discussion of the requirements

- Requirements **R1** and **R2** are basically the same as the control objectives **G1** and **G2**

$\rightarrow$ We can satisfy them using the PI-controller we already introduced

- Here, we will focus mainly on **R3** to **R5**
- We will see that **R3** is directly related to **G3**, i.e., the communication structure design
- **R4** and **R5** are additional specifications, that extend the control objectives we stated before

R3: time-headway spacing

- We want to achieve that, for all $k \geq 0$ and all vehicles

$$d_k^i = \alpha^i + \beta^i v_k^i$$

- If we assume that it is satisfied at $k = 0$, we only need that

$$d_{k+1}^i = d_k^i + \beta^i (v_{k+1}^i - v_k^i) \quad \forall k > 0$$

- We know that $d_k^i = s_k^{i-1} - s_k^i$. Therefore,

$$d_{k+1}^i \approx d_k^i + (v_k^{i-1} - v_k^i) T_{\mathrm{s}}$$

- Thus, we need

$$v_{k+1}^i = v_k^i + \frac{1}{\beta^i}(v_k^{i-1} - v_k^i) T_{\mathrm{s}}$$

$\rightarrow$ This we can achieve with an ACC already!

Impact of delays

- We already discussed that the network structure introduces a delay
- In an acceleration maneuver, it takes some time until following vehicles become aware of the change and increase their speed
- Thus, the distance between vehicles will increase during the transient
- In a braking maneuver, the distance will decrease, which might lead to an accident

$\rightarrow$ For reaching asymptotic time-headway spacing with ACC, we need $\beta^i = \Delta^i$ for all vehicles

R3 with CACC

- What happens if $\Delta^i > \beta^i$? Can we improve the situation with CACC?
- If we already know in advance that the leader is braking, not only when we see the delayed response of the preceding vehicle, we might be able to react faster
- Let us investigate the asymptotic distance between vehicles
- As the entire system is stable, we have $\lim_{k\to\infty} v_k^0 = \bar{v}$
- Then, we can look at the distance of vehicle i from the leader:

$$\lim_{k\to\infty} d_k^{i,0} = \sum_{j=1}^{i} \alpha^i + \Delta^{i,0}\bar{v}$$

- For the distance between Vehicles i and $i-1$, we can now write

$$\lim_{k\to\infty} \tilde{d}_k^i = \lim_{k\to\infty} (d_k^{i,0} - d_k^{i-1,0}) = (\Delta^{i,0} - \Delta^{i-1,0})\bar{v}$$

$\rightarrow$ We need $\Delta^{i,0} - \Delta^{i-1,0} = \beta^i$

Delay for CACC

- We discussed delays for different communication structures and found that

$$\Delta^{i,0} = \Delta^{i} + \sum_{j=0}^{i-1} \hat{a}_{ij} \Delta^{j,0}$$

- Thus, we have

$$\lim_{k \to \infty} \tilde{d}_k^i = \underbrace{\left(\Delta^{i} - \Delta^{i-1} + \sum_{j=0}^{i-1} \hat{a}_{ij} \Delta^{j,0} - \sum_{j=0}^{i-2} \hat{a}_{i-1,j} \Delta^{j,0} \right) \bar{v}}_{\overset{!}{=} \beta^i}$$

Time-headway spacing with CACC

- We can now design the communication structure (i.e., the normalized adjacency matrix $\hat{A}_{\mathcal{G}}$) such that all vehicles satisfy **R3** even if for some *i* we have $\Delta^i > \beta^i$
- The design implies that

$$\Delta^{i,0} = \sum_{j=1}^{i} \beta^j$$

- Using this and $\Delta^{i,0} = \Delta^i + \sum_{j=0}^{i-1} \hat{a}_{ij}\Delta^{j,0}$ from before, we can write

$$\sum_{j=1}^{i} \beta^j - \Delta^i \overset{!}{=} \sum_{j=1}^{i-1} \hat{a}_{ij}\Delta^{j,0}, \quad i = 1, 2, \ldots, N$$

- Then, we end up with two conditions for the delays:

$$\beta^1 = \Delta^1$$

$$\beta^i \leq \Delta^i \leq \sum_{j=1}^{i} \beta^j, \quad i = 2, 3, \ldots, N$$

Time-headway spacing with CACC

- From these equations, we can get an algorithm for designing the adjacency matrix
- We start with $\hat{a}_{10} = 1$
- Next, we set

$$
\begin{aligned}
\hat{a}_{i\ell^i} &= \frac{\sum_{j=\ell^i}^{i} \beta^j - \Delta^i}{\beta^{\ell^i}}, & i &= 2, 3, \ldots, N \\
\hat{a}_{i,\ell^i-1} &= 1 - \hat{a}_{i\ell^i}, & i &= 2, 3, \ldots, N \\
\hat{a}_{ij} &= 0, & j &\neq \ell^i, j \neq \ell^i - 1
\end{aligned}
$$

- Where we choose the indices ℓ^i such that we satisfy

$$
0 \leq \sum_{j=\ell^i}^{i} \beta^j - \Delta^i \leq \beta^{\ell^i}, \quad i = 2, 3, \ldots, N
$$

$\rightarrow$ We can satisfy the conditions using only two communication links to each vehicle

- The last inequality tells us from which other vehicle each vehicle needs information

Collision avoidance and positive velocities

- We can define the vehicle model as one with two outputs: the current velocity and the distance to the vehicle in front of it
- Both should be positive at all times
- This is captured by the notion of externally positive systems

Definition: externally positive systems

A linear system is said to be *externally positive* if for zero initial state its output is non-negative for every non-negative input:

$$u_k \geq 0,\ k \geq 0 \quad \Longrightarrow \quad y_k \geq 0,\ k \geq 0.$$

- If each vehicle is externally positive with respect to the output v_k^i, it is also externally positive with respect to $\tilde{d}_k^i$
- If we use ACC and all vehicles are externally positive, we satisfy **R4** and **R5**
- For CACC, the condition is a bit more involved: we need that also the transfer function of the platoon represents an externally positive system

Example: platoon with identical vehicles

- Let's try to design a suitable controller and communication structure for a platoon with identical vehicles
- For identical vehicles, we can assume $\beta^i = \beta$ and $\Delta^i = \Delta$ for all i
- We assume $\beta < \Delta \leq 2\beta$

$\rightarrow$ We need CACC to make this happen

- Before, we stated that to satisfy **R3**, we need

$$0 \leq \sum_{j=\ell^i}^{i} \beta^j - \Delta^i \leq \beta^{\ell^i}, \quad i = 2, 3, \ldots, N$$

- For our example, choosing $\ell^i = i - 1$, this translates to

$$0 \leq 2\beta - \Delta \leq \beta$$

Example: platoon with identical vehicles

- We then get the local reference signals

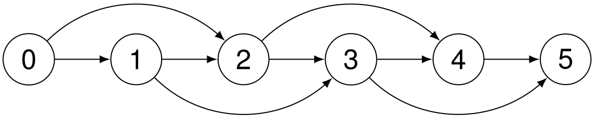
$$
\begin{aligned}
v_k^{\mathrm{s},1} &= v_k^0 \\
v_k^{\mathrm{s},i} &= \hat{a}_1 v_k^{i-1} + \hat{a}_2 v_k^{i-2}, \quad i = 2, 3, \ldots, N
\end{aligned}
$$

- Where the coefficients are

$$
\begin{aligned}
\hat{a}_1 &= \hat{a}_{i,i-1} = 2 - \frac{\Delta}{\beta} \\
\hat{a}_2 &= \hat{a}_{i,i-2} = \frac{\Delta}{\beta} - 1
\end{aligned}
$$

- Note that this satisfies also our normalization condition: $\hat{a}_1 + \hat{a}_2 = 1$
- Given a proper controller, this suffices to meet all our control objectives

Example: platoon with identical vehicles



Learning outcomes

By the end of this lecture, you should be able to

- Describe the problem of designing communication structures for networked controllers
- Design communication structures for networked controllers
- Describe design objectives of controllers for vehicle platooning
- Name conditions for controllers and communication structures that meet these objectives

Feedback

Feedback

Please leave some feedback for today's lecture: `https://presemo.aalto.fi/nmas`