Lecture

Control and Perception in Networked and Autonomous Vehicles

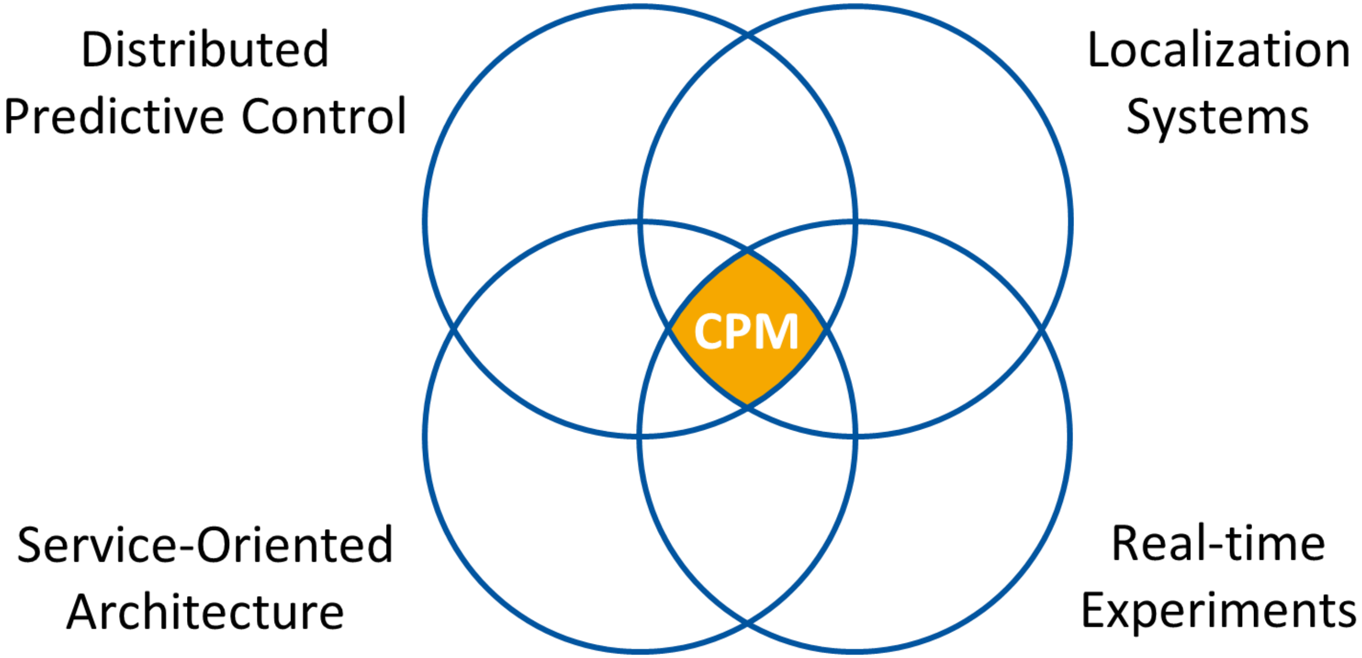
Dr. Bassam Alrifaee | Patrick Scheffe, M. Sc. | Simon Schäfer, M. Sc.
Winter Semester 2023/2024

Part 4

Network and Distribution

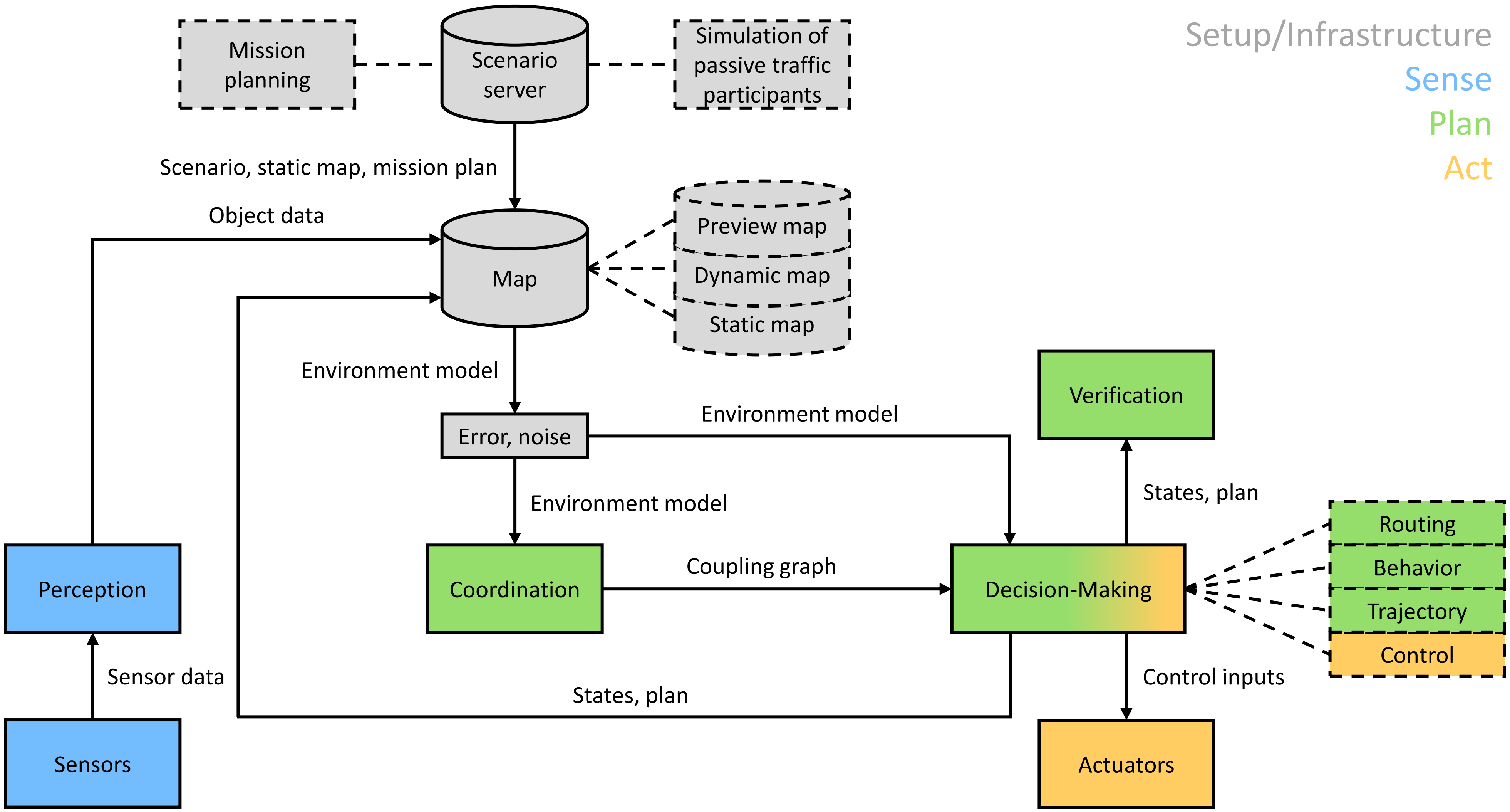
Course contents (CPM group course)

- Vehicle models
- Control and optimization
- Network and distribution
- Machine perception
- Software architectures and testing concepts
- Case study

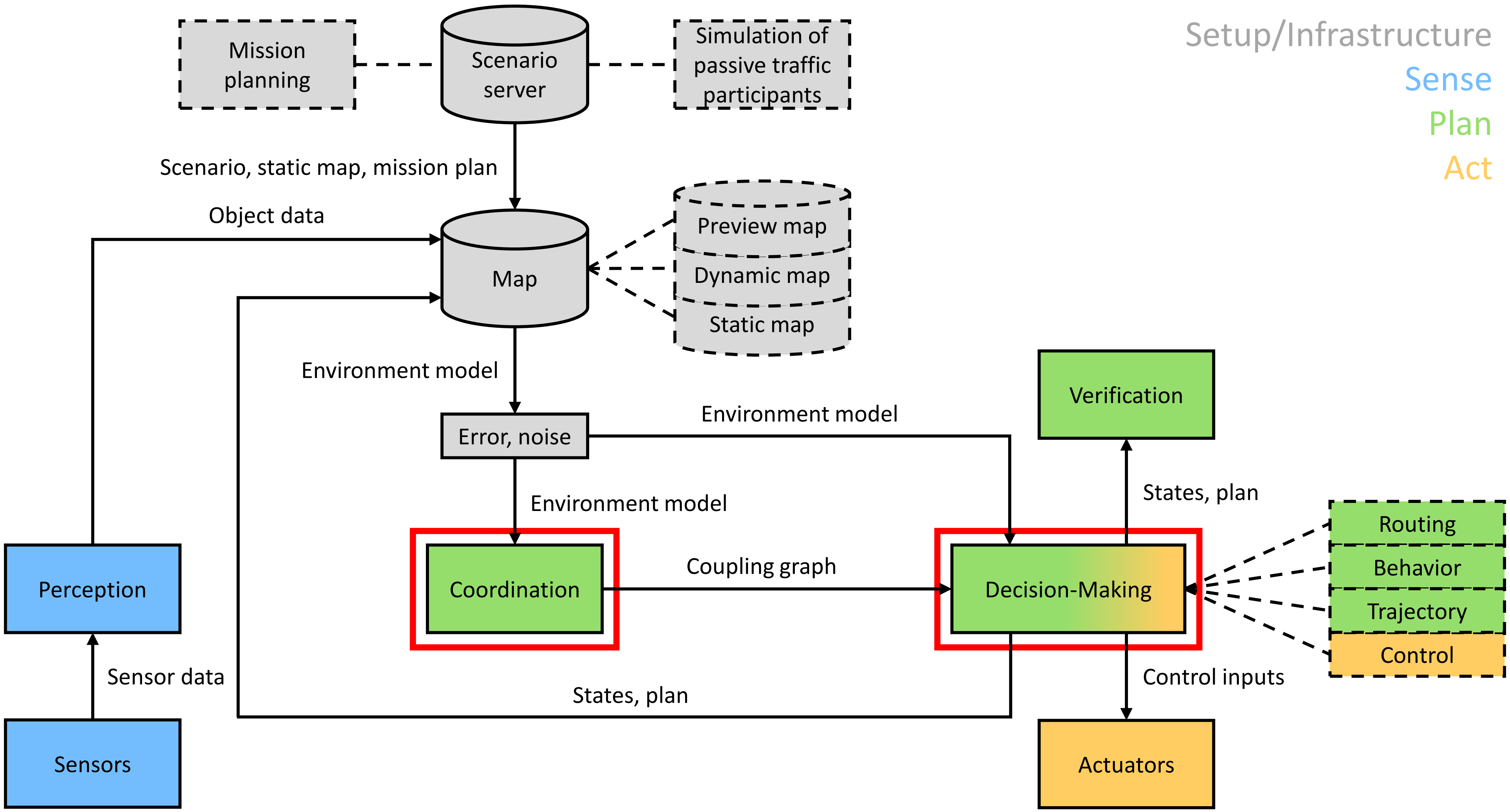


*CPM: Cyber-Physical Mobility

CPM Lab architecture



CPM Lab architecture



Setup/Infrastructure
Sense
Plan
Act

Literature

- J. Lunze. Control Theory of Digitally Networked Dynamic Systems. Springer, 2014.

Further literature (1)

- J. Lunze. Networked Control of Multi-Agent Systems. Bookmundo Direct, 2019

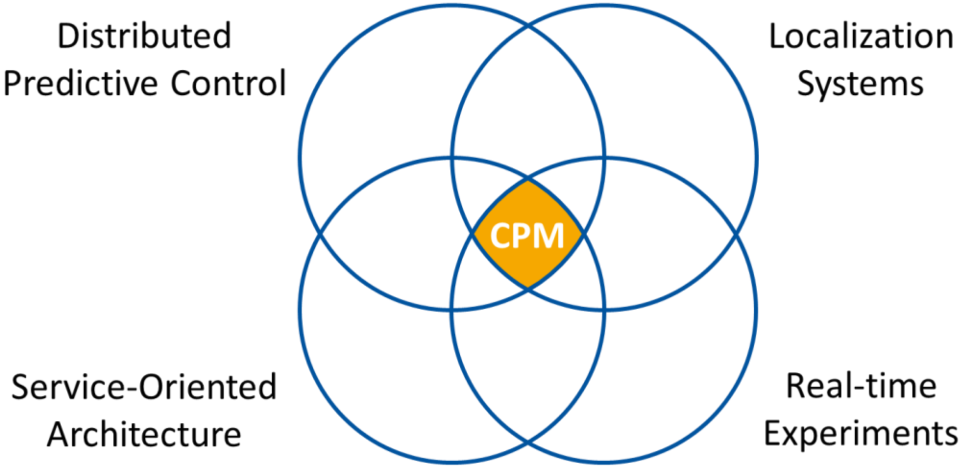
Further literature (2)

- B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
- B. Alrifaee. MATLAB Simulation of Networked Model Predictive Control for Vehicle Collision Avoidance, 2017. Available: https://doi.org/10.5281/zenodo.1252992

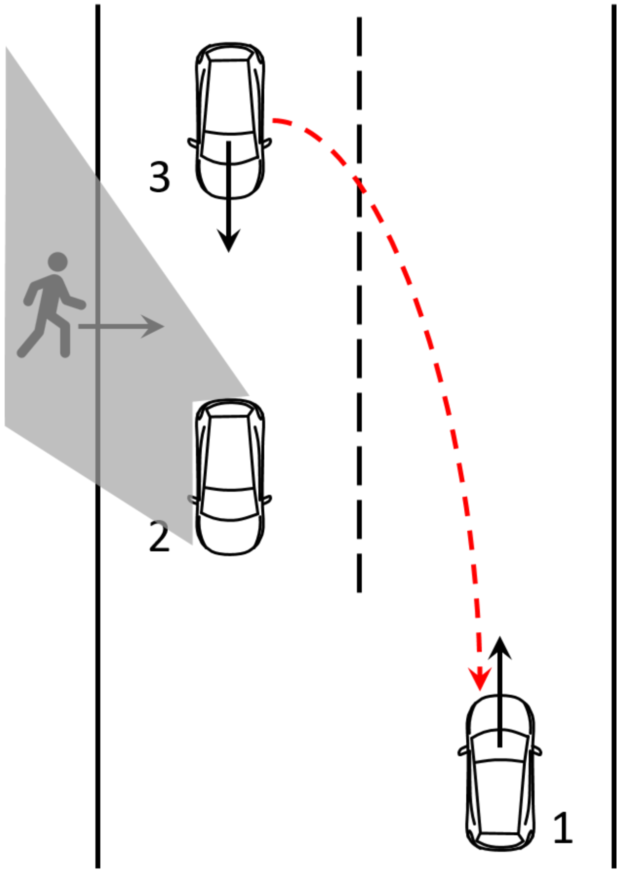
- Check out our website

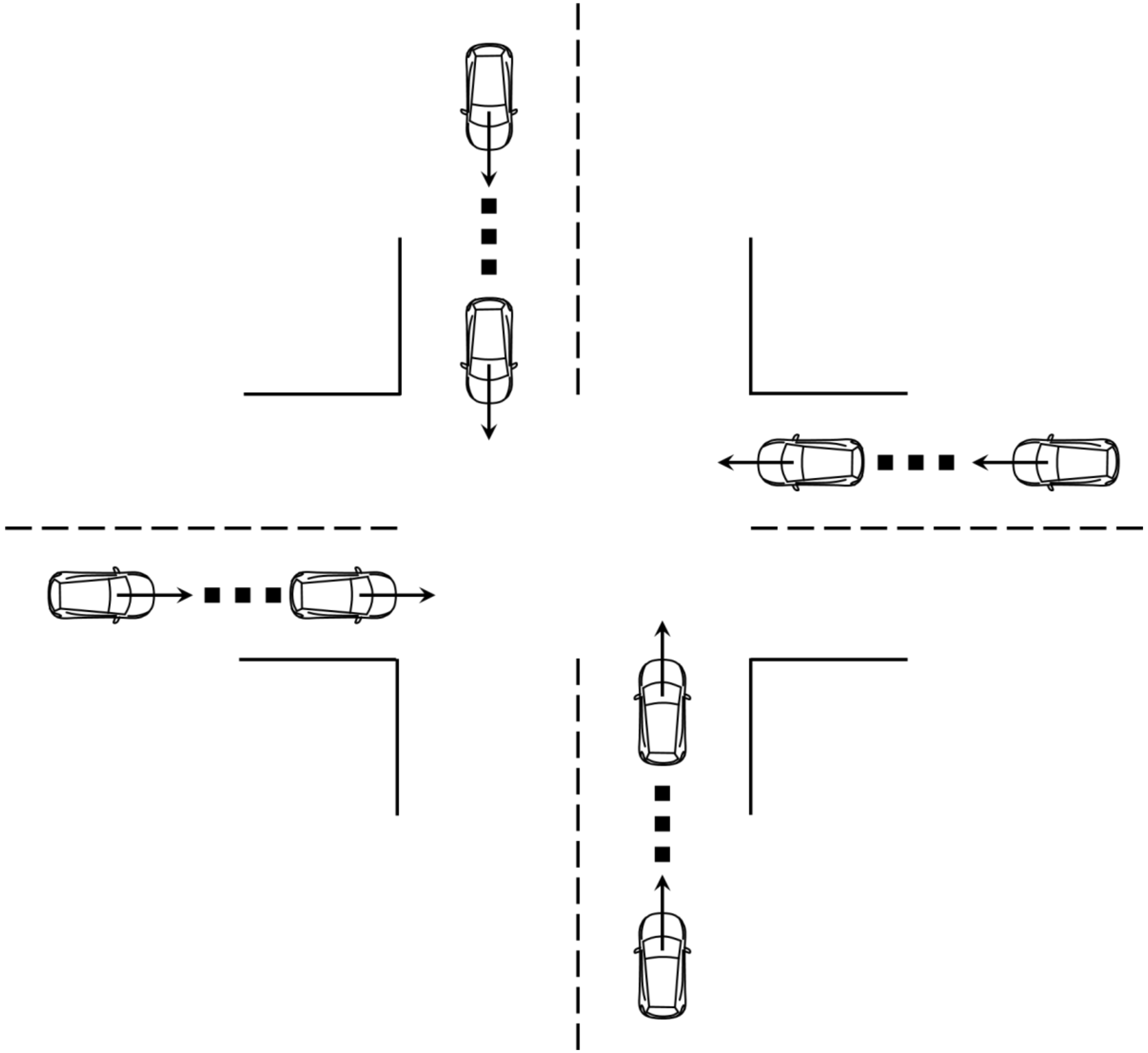
Definition of networked systems

- A.k.a. connected systems
- Communicate and interact
- Improve
  - Perception
  - Decision-making
- Impose challenges



Examples of improvement of perception and decision-making





Pedestrians walking

- Two pedestrians approach each other
- Avoid each other once
- Avoid each other twice
- .
- .
- .
- (Collide with each other)

- n-pedestrians

Shibuya crossing

Beauty contest game

- Setup
  - Choose number between 0 and 100
  - Winner = Closest to 1/2 of average

Shamma, course on game theory and distributed control, 2019

Beauty contest game

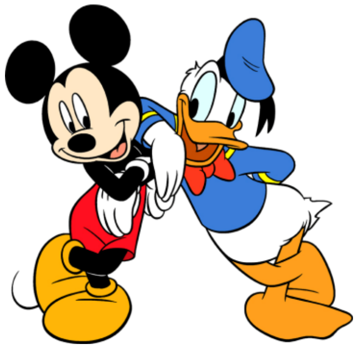
- Setup
  - Choose number between 0 and 100
  - Winner = Closest to 1/2 of average
- Decision quality of individuals affected by decisions of others

Shamma, course on game theory and distributed control, 2019

Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY

Beauty contest game

- Setup
  - Choose number between 0 and 100
  - Winner = Closest to 1/2 of average
- Decision quality of individuals affected by decisions of others
- Why “beauty contest”?


- Favorite characters = ???
- Compare: stock market. See [Keynesian beauty contest](). See [A Beautiful Mind (film)]()

Shamma, course on game theory and distributed control, 2019

Fisher problem

- Story

$$x(t_0) = 100\,\text{tons}$$

$$N = 5\,\text{number of players/groups}$$

$$i = 1, \dots, N\,\text{player/group}$$

$$u^{(i)}(t) \in \{0, \dots, 10\}\,\text{tons}$$

$$x(t+1) = 1.2 \cdot x(t) - \sum_{i=1}^{N} u^{(i)}(t)$$

$$\text{Task:}\quad \max_{u^{(i)}(\cdot)} \sum_t u^{(i)}(t),\ \text{s.t.}\ x(t) \geq 1\ ,\ \text{i.e., survive,}\ \forall t$$

Thanks to L. Dörschel for the discussion

Fisher problem – discussion

- Fisher problem as MPC
  - Discuss the effect of the prediction horizon
  - The MPC algorithm is greedy, if the prediction horizon is ...

This lecture

$$x(t_0) = 0\,\text{participants}$$

$$N = 30\,\text{number of participants}$$

$$i = 1, \ldots, N\,\text{participant}$$

$$u^{(i)}(t) \in \{0, 1\}$$

$$x(t+1) = \sum_{i=1}^{N} u^{(i)}(t)$$

$$\text{Task:} \quad \min_{u^{(i)}(\cdot)} \sum_{t} u^{(i)}(t), \text{ s.t. } x(t) \geq 18,\ \forall t$$

This lecture – discussion

- This lecture as MPC
  - How long is the prediction horizon?
  - What is the best strategy?

COVID-19 pandemic

$$x(t_0) = 0 \, \text{infected}$$

$$N = 80\text{M} \, \text{number of population}$$

$$i = 1, \dots, N \, \text{population}$$

$$u^{(i)}(t) \in \{0, 1\}$$

$$x(t+1) = x(t) + R_{in}x(t) - R_{out}x(t), \text{ where } R_{in}x(t) = \sum_{i=1}^{N} u^{(i)}(t)$$

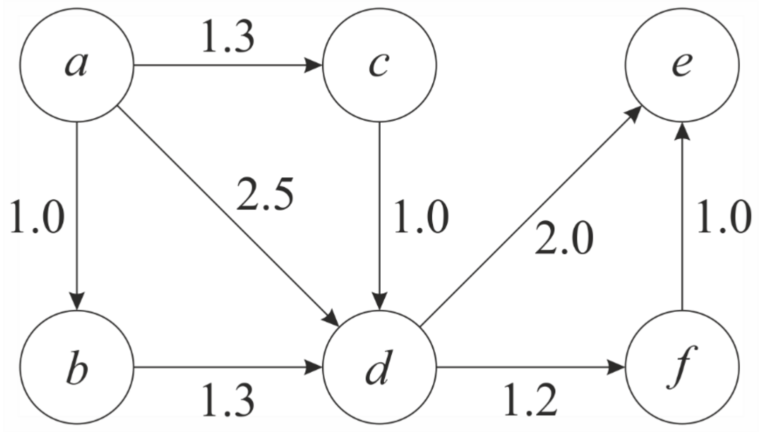
$$\text{Task: } \max_{u(t)} \sum_{i=1}^{N} u^{(i)}(t), \text{ s.t. } R_{intensive}x(t) \leq 3,000 \,, \forall t$$

COVID-19 pandemic – discussion

- COVID-19 pandemic as MPC
  - Discuss the effect of the prediction horizon

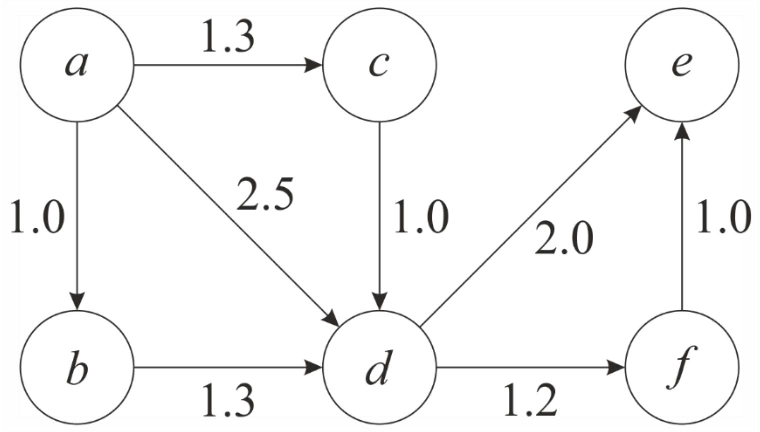
Graph theory

- Powerful tool for modeling and analyzing networked systems
- Reading
  - B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
    - Section 2.2, pages 5-7
- Definitions
- MATLAB exercise



Graph theory – example

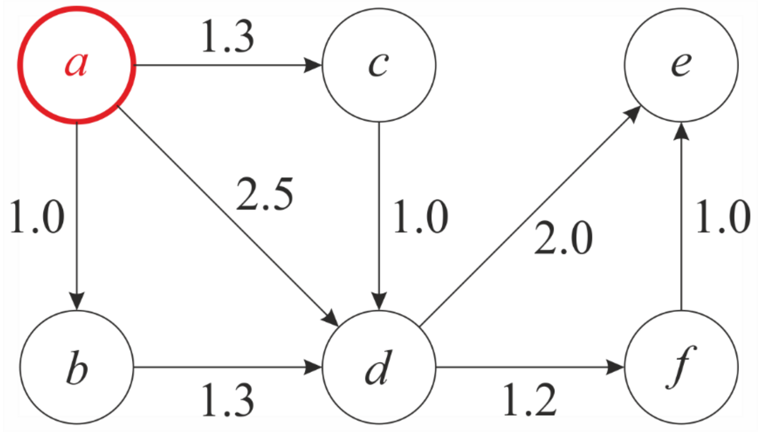
- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a



Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	No	∞	
c	No	∞	
d	No	∞	
e	No	∞	
f	No	∞	

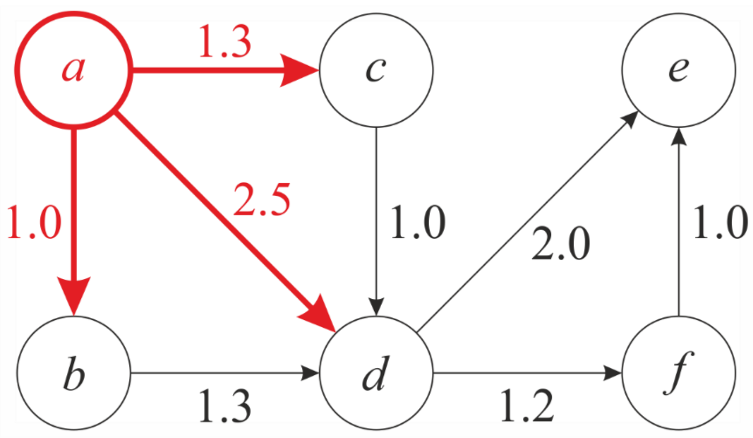


$Q = []$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	No	1.0	a
c	No	1.3	a
d	No	2.5	a
e	No	∞	
f	No	∞	

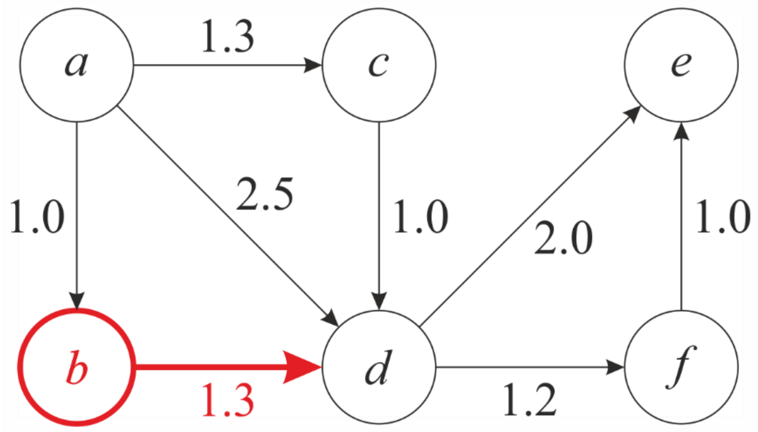


$Q = [b, c, d]$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	Yes	1.0	a
c	No	1.3	a
d	No	2.3	b
e	No	∞	
f	No	∞	

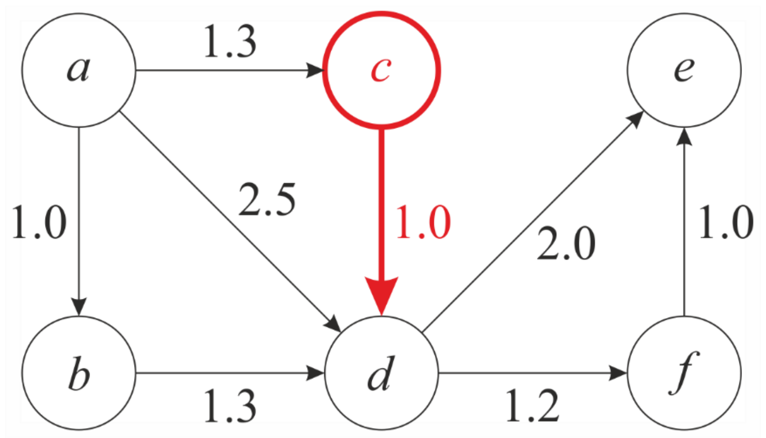


$Q = [\cancel{b}, c, d]$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	Yes	1.0	a
c	Yes	1.3	a
d	No	2.3	b
e	No	∞	
f	No	∞	

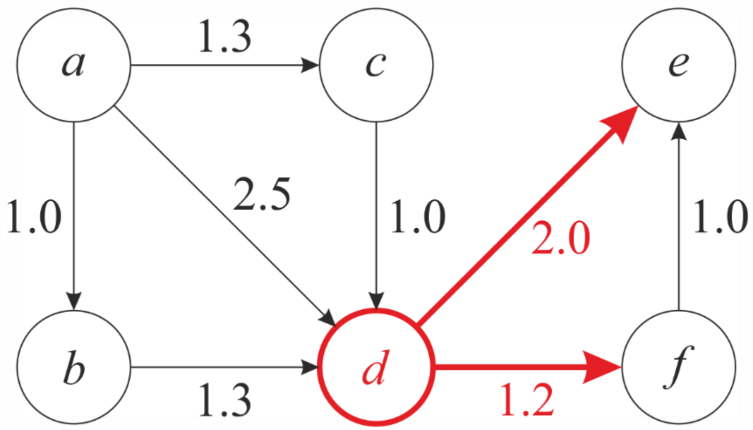


$Q = [\cancel{b}, \cancel{c}, d]$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	Yes	1.0	a
c	Yes	1.3	a
d	Yes	2.3	b
e	No	4.3	d
f	No	3.5	d

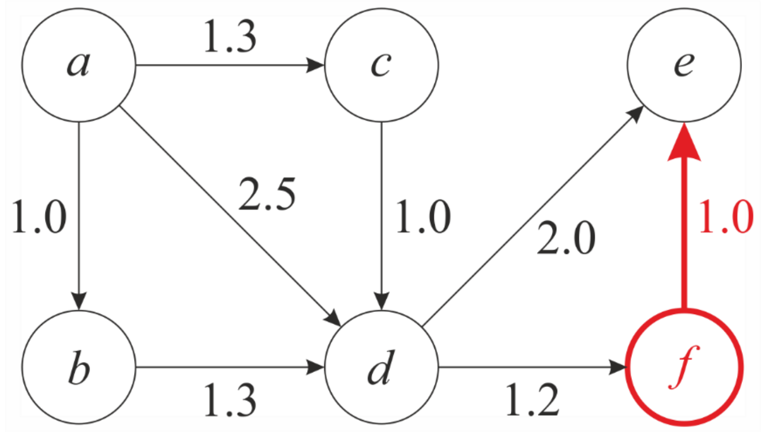


$Q = [\cancel{b}, \cancel{c}, \cancel{d}, e, f]$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	Yes	1.0	a
c	Yes	1.3	a
d	Yes	2.3	b
e	No	4.3	d
f	Yes	3.5	d

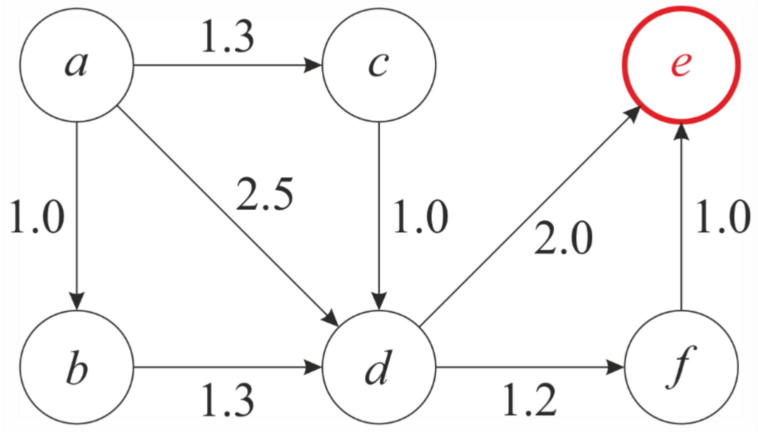


$Q = [\cancel{b}, \cancel{c}, \cancel{d}, e, \cancel{f}]$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	Yes	1.0	a
c	Yes	1.3	a
d	Yes	2.3	b
e	Yes	4.3	d
f	Yes	3.5	d

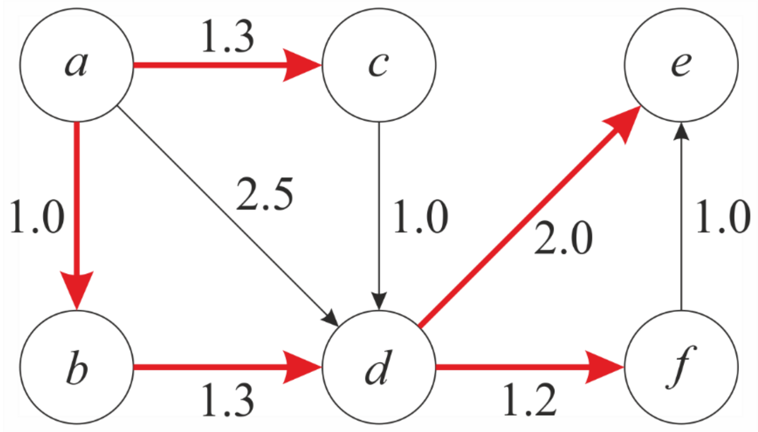


$Q = [\cancel{b}, \cancel{c}, \cancel{d}, \cancel{e}, \cancel{f}]$

Graph theory – example

- Dijkstra's algorithm for finding the shortest paths between nodes in a graph
- **Input:** graph with positive edge weights and a starting vertex a
- **Output:** shortest paths between a and all other reachable nodes from a

Vertex	Done	Distance	From
a	Yes	0	
b	Yes	1.0	a
c	Yes	1.3	a
d	Yes	2.3	b
e	Yes	4.3	d
f	Yes	3.5	d



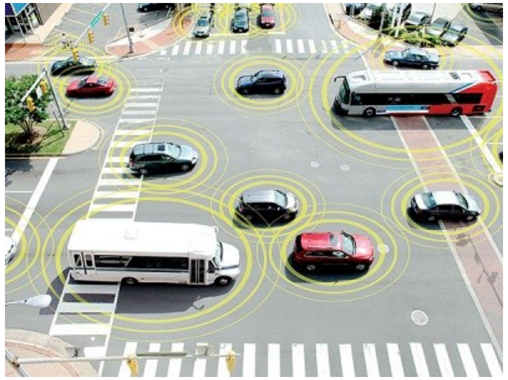
Computation: $O(N\log(N) + M)$

Networked model predictive control

- Flipped classroom
  - Group D should prepare a summary, ca. 15 minutes
- Reading
  - B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
    - Chapter 3, pages 20-51

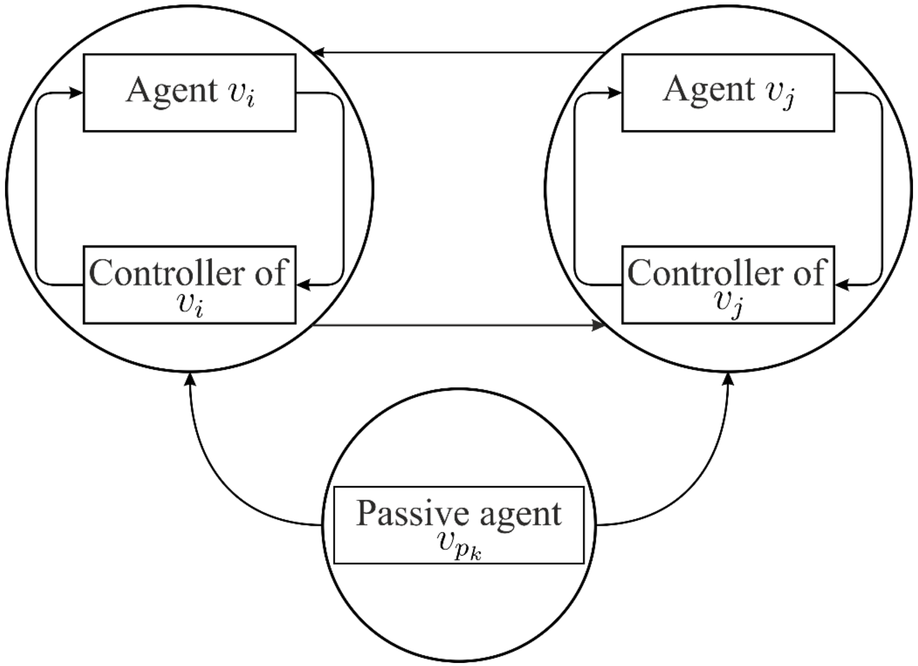
Networked control systems (NCS)

- NCS consist of interacting agents (dynamic subsystems)

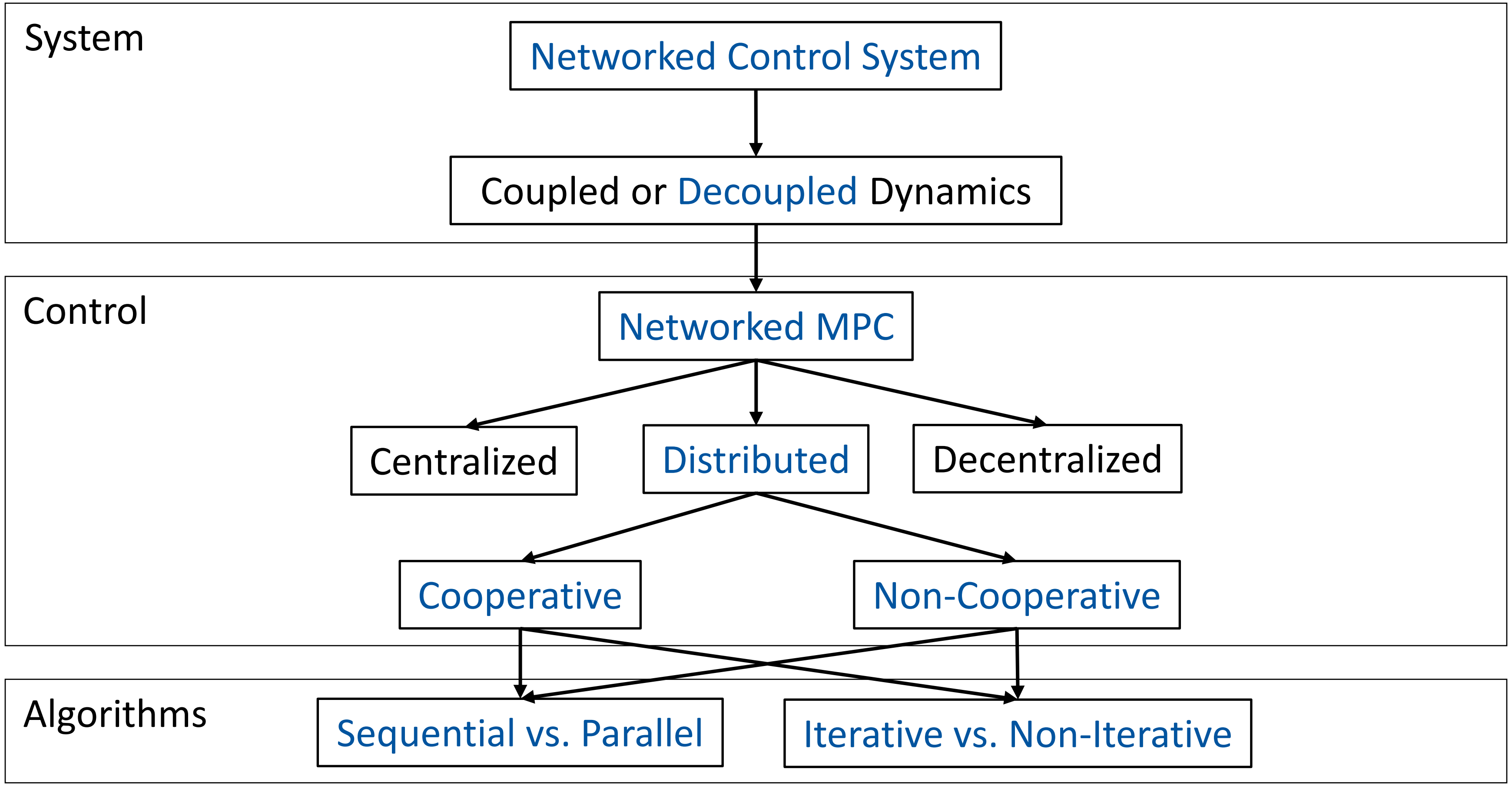


NCS

- Passive agents
  - Dynamic subsystems without networked control
  - Data communication to active agents
- (Active) Agents
  - Data exchange
  - Achieve their goals while taking the interaction with other agents into consideration
  - Full knowledge about passive agents and their future states
- Classification of agents as a step to full automation of NCS and consideration of non-automatable agents
- Communication restrictions, e.g., time delays, and computation time affect stability and performance
- Network: time-invariant or time-variant



NCS classification



NCS classification

- *Control strategy*: combination of a control method and the algorithm applied to it
- Selection of control strategy based on:
  - NCS categories in the system level
  - Available computation time
  - Communication requirements
- *Computation time*: time required for the whole NCS to reach a solution at a given time step, i.e.,
  - Measure the states
  - Formulate and solve the optimization problem
  - Apply the inputs to all agents
  - Communication of required data

  Sequence depends on the control strategy

Networked model predictive control

- Method
  - Enhancing feasibility (safety and efficiency)
  - Reducing computation time and communication

Formulation of Net-MPC: basic formulation

- Strength of Net-MPC:
  Making decisions while considering plans of agents

$$J^{(i)^\star} = \min_{\Delta \mathbf{u}^{(i)}(\cdot)}$$

$$\sum_{k=1}^{H_p-1} l_x^{(i)}(\mathbf{x}^{(i)}(t+k), \mathbf{r}^{(i)}(t+k)) + l_{xH_p}^{(i)}(\mathbf{x}^{(i)}(t+H_p), \mathbf{r}^{(i)}(t+H_p)) +$$

$$\sum_{k=0}^{H_u-1} l_u^{(i)}(\Delta \mathbf{u}^{(i)}(t+k)) + \boxed{\sum_{\substack{j \\ v_j \in \mathcal{V}^{(i)}}} \sum_{k=1}^{H_p} c_o^{(i,j)}(\mathbf{x}^{(i)}(t+k), \mathbf{x}^{(j)}(t+k))}$$

$$\text{subject to } (\forall v_j \in \mathcal{V}^{(i)},\ \forall v_p \in \mathcal{V}_p^{(i)}):$$

$$\mathbf{x}^{(i)}(t+1+k) = f^{(i)}(\mathbf{x}^{(i)}(t+k), \mathbf{u}^{(i)}(t+k)),\ k = 0, \ldots, H_p - 1$$

$$\mathbf{x}^{(i)}(t+k) \in \mathcal{X}^{(i)},\ k = 1, \ldots, H_p - 1$$

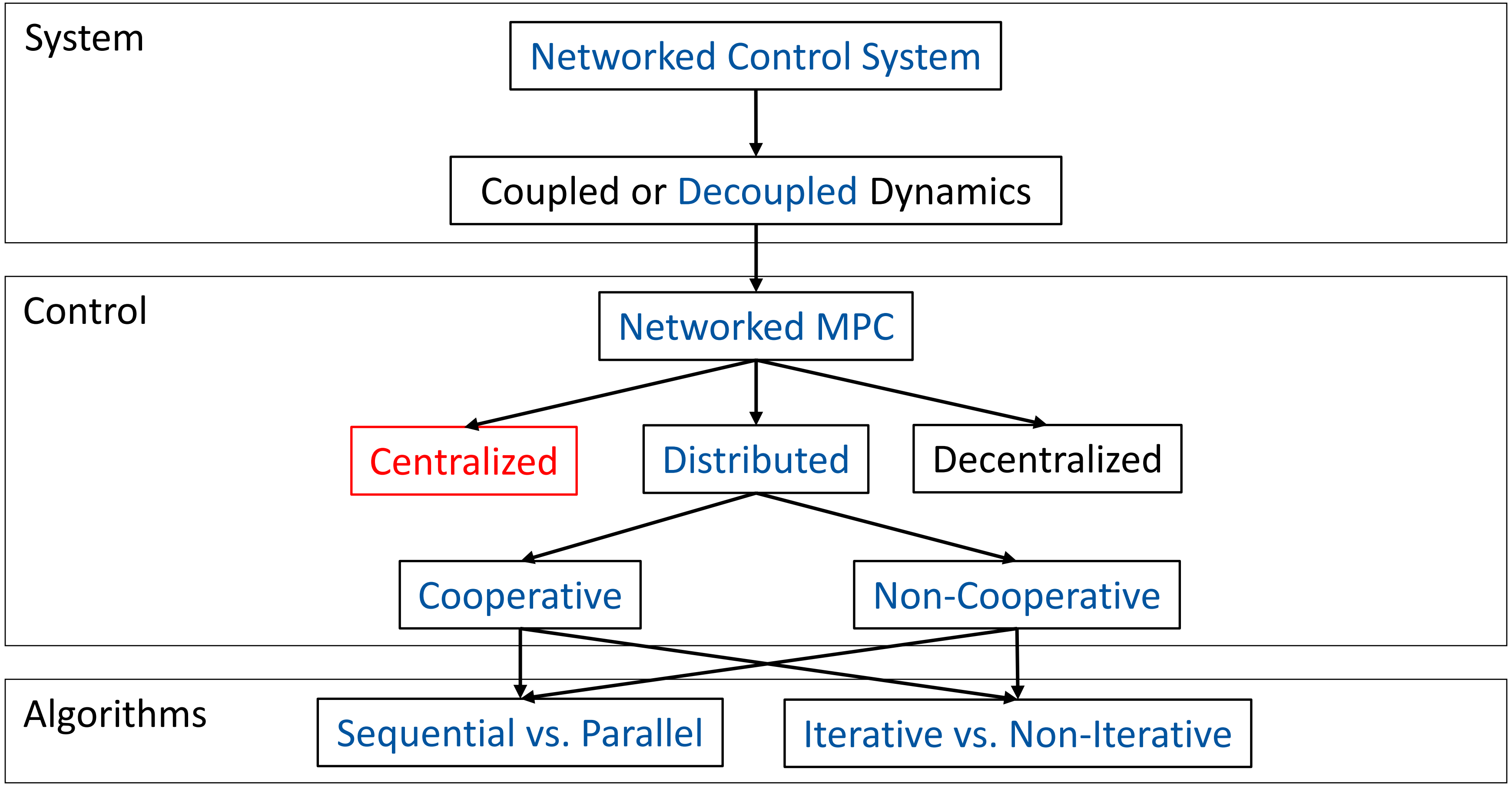
$$\mathbf{x}^{(i)}(t+H_p) \in \mathcal{X}_{H_p}^{(i)}$$

$$\mathbf{u}^{(i)}(t+k) \in \mathcal{U}^{(i)},\ k = 0, \ldots, H_u - 1$$

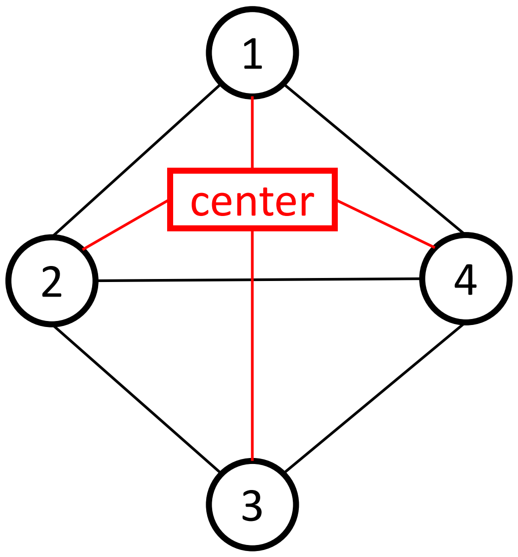
$$\Delta \mathbf{u}^{(i)}(t+k) \in \Delta \mathcal{U}^{(i)},\ k = 0, \ldots, H_u - 1$$

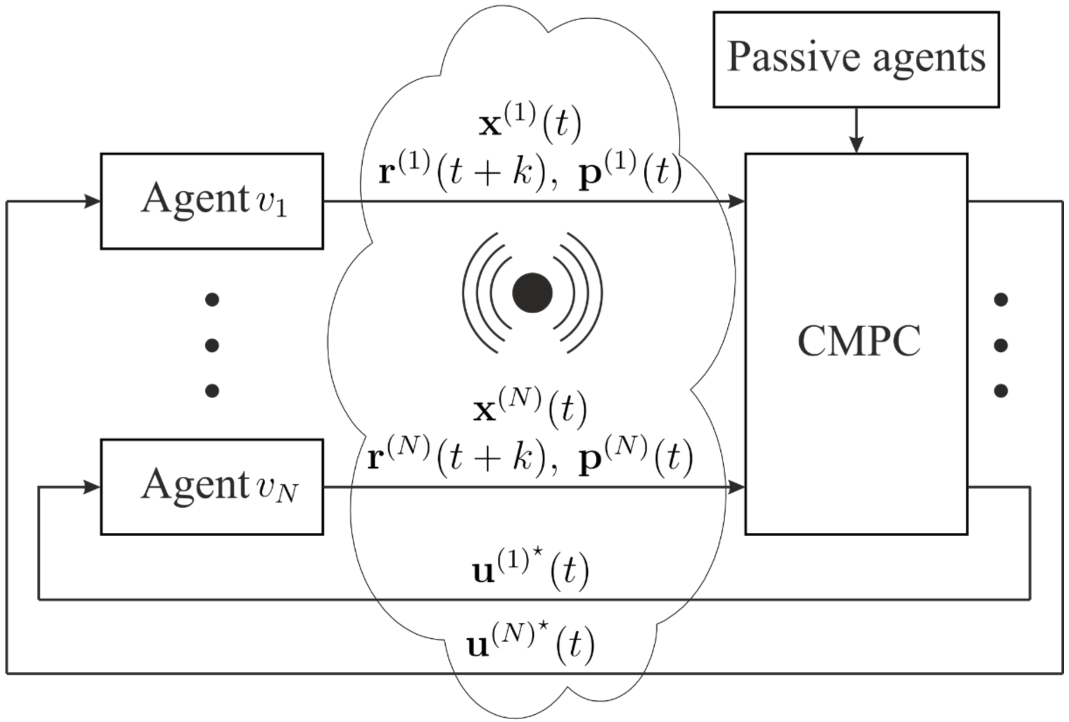
$$\boxed{\begin{aligned} &c_c^{(i,j)}(\mathbf{x}^{(i)}(t+k), \mathbf{x}^{(j)}(t+k)) \le 0,\ k = 1, \ldots, H_p \\ &c_c^{(i,p)}(\mathbf{x}^{(i)}(t+k), \mathbf{x}^{(p)}(t+k)) \le 0,\ k = 1, \ldots, H_p \end{aligned}}$$

NCS Classification



Net-MPC: centralized MPC



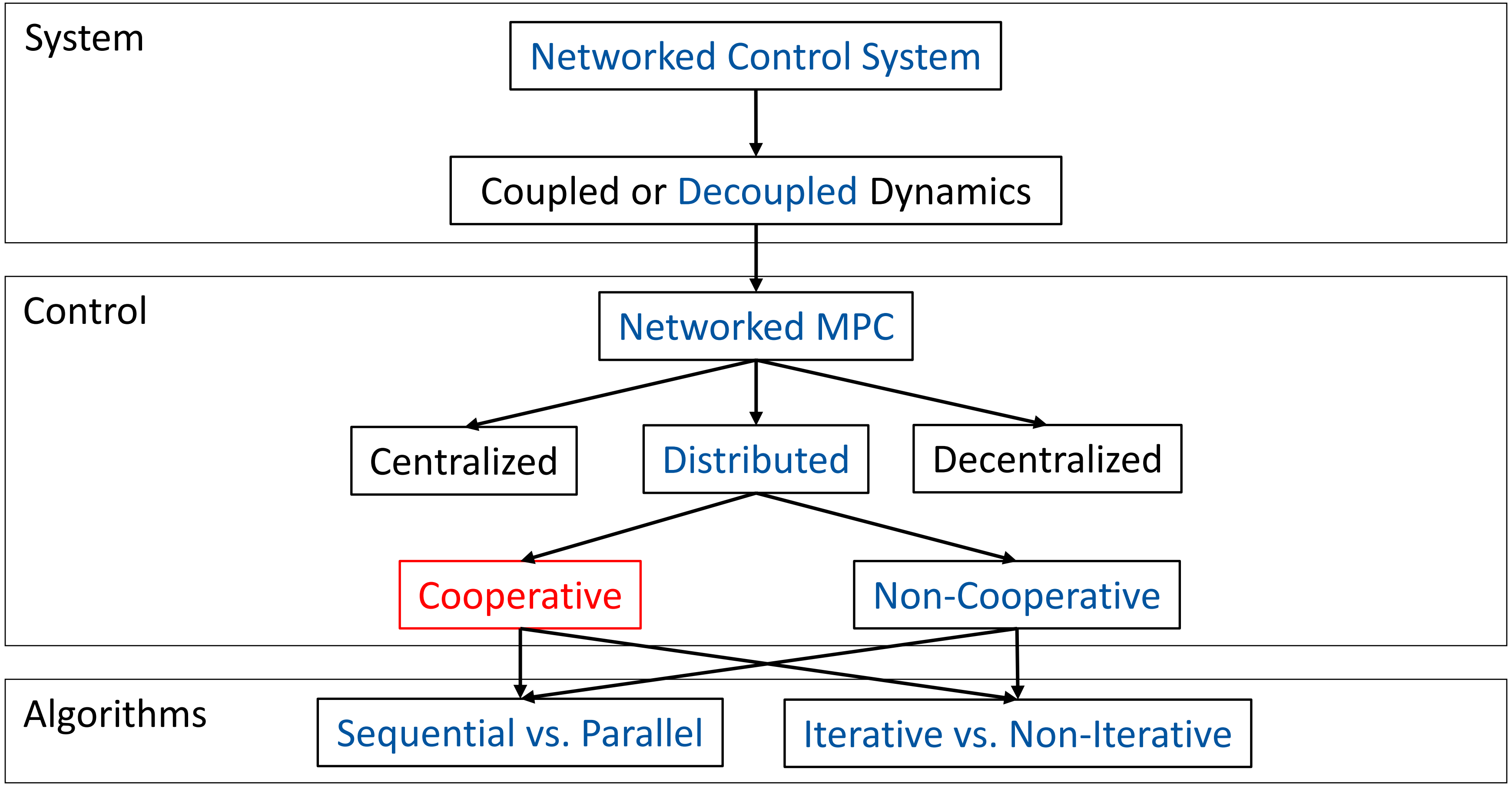


- Not applicable in practice due to
  - High computation time
  - Safety hazards
- Benchmark for comparing different distributed MPC strategies

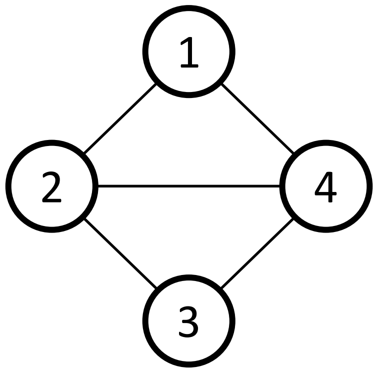
Net-MPC: terms

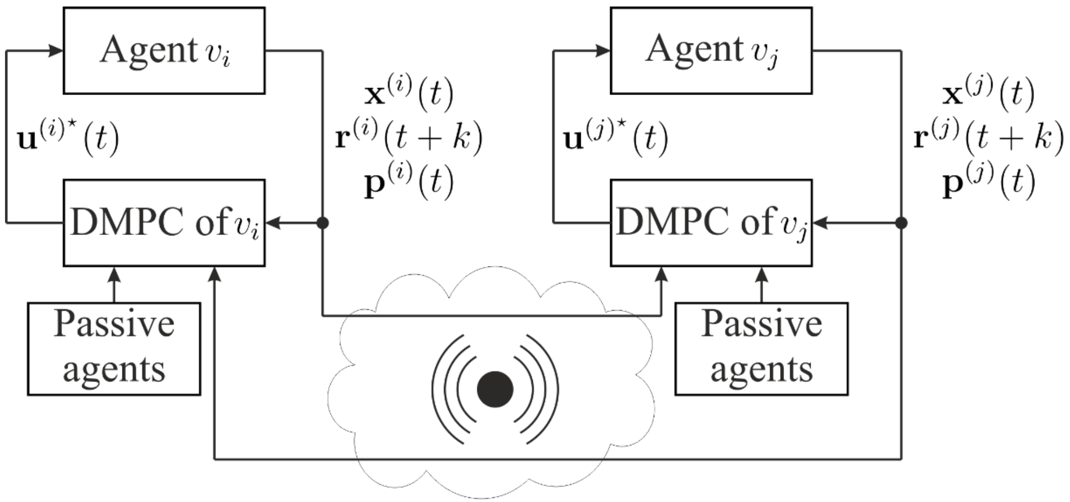
- NCS performance
  - NCS stability, solution feasibility, solution optimality, solution quality
- A NCS is ***stable*** if each of its agents is stable in the network
- A solution is ***agent-feasible*** if each agent's controller generates a feasible solution in terms of its own optimization problem
- A solution is ***NCS-feasible*** if it is feasible in terms of a corresponding CMPC
- A solution is ***agent-optimal*** if each single agent's controller generates an optimal solution in terms of its own optimization problem
- A solution is ***NCS-optimal*** if it is optimal in terms of a corresponding CMPC
- The ***NCS-quality*** is defined as the quality of a solution compared with the solution of CMPC
- Assumption: a solution to CMPC exists and it is NCS-stable, -feasible, and -optimal

NCS Classification



Net-MPC: cooperative distributed MPC





- Decomposition of centralized MPC into smaller optimization problems
- Each agent just considers hypothetical plans of its neighbors

Net-MPC: cooperative distributed MPC

Prediction consistency

- Prediction consistency means that the predictions $x^{(j)}(t+k)$, $k=1,\ldots,H_p$ used or computed in the optimization problem of an agent v_i at a time instance t for an agent v_j coincide with the predictions computed by agent v_j itself at the same time instance t
- Without satisfaction of this property, no guarantee for NCS-stability and -feasibility
- Coop. DMPC does not satisfy the prediction consistency property
  - Exception: if the NCS is fully connected
  - Coop. DMPC becomes CMPC except the communication structures
  - High computation time

Informatik 11 Embedded Software

RWTH AACHEN UNIVERSITY

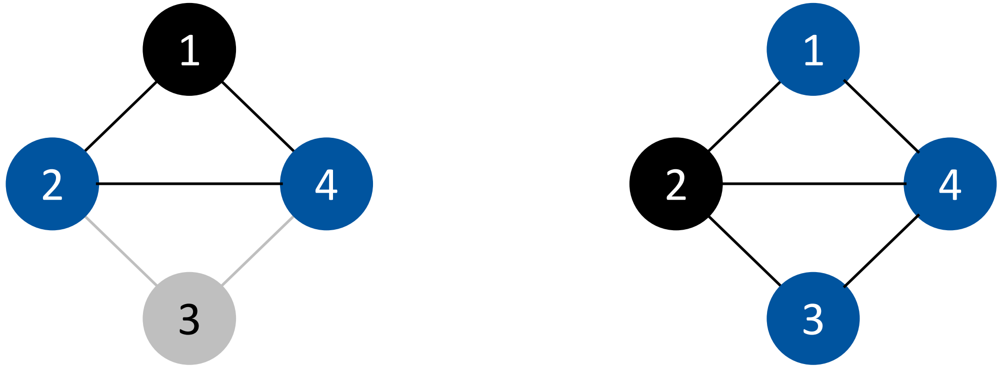
Net-MPC: cooperative distributed MPC

Local System Knowledge

- Agent 1 considers (hypothetical) plans of
  - Agents 1, 2, 4, ~~3~~
- Agent 2 considers (hypothetical) plans of
  - Agents 2, 1, 3, 4

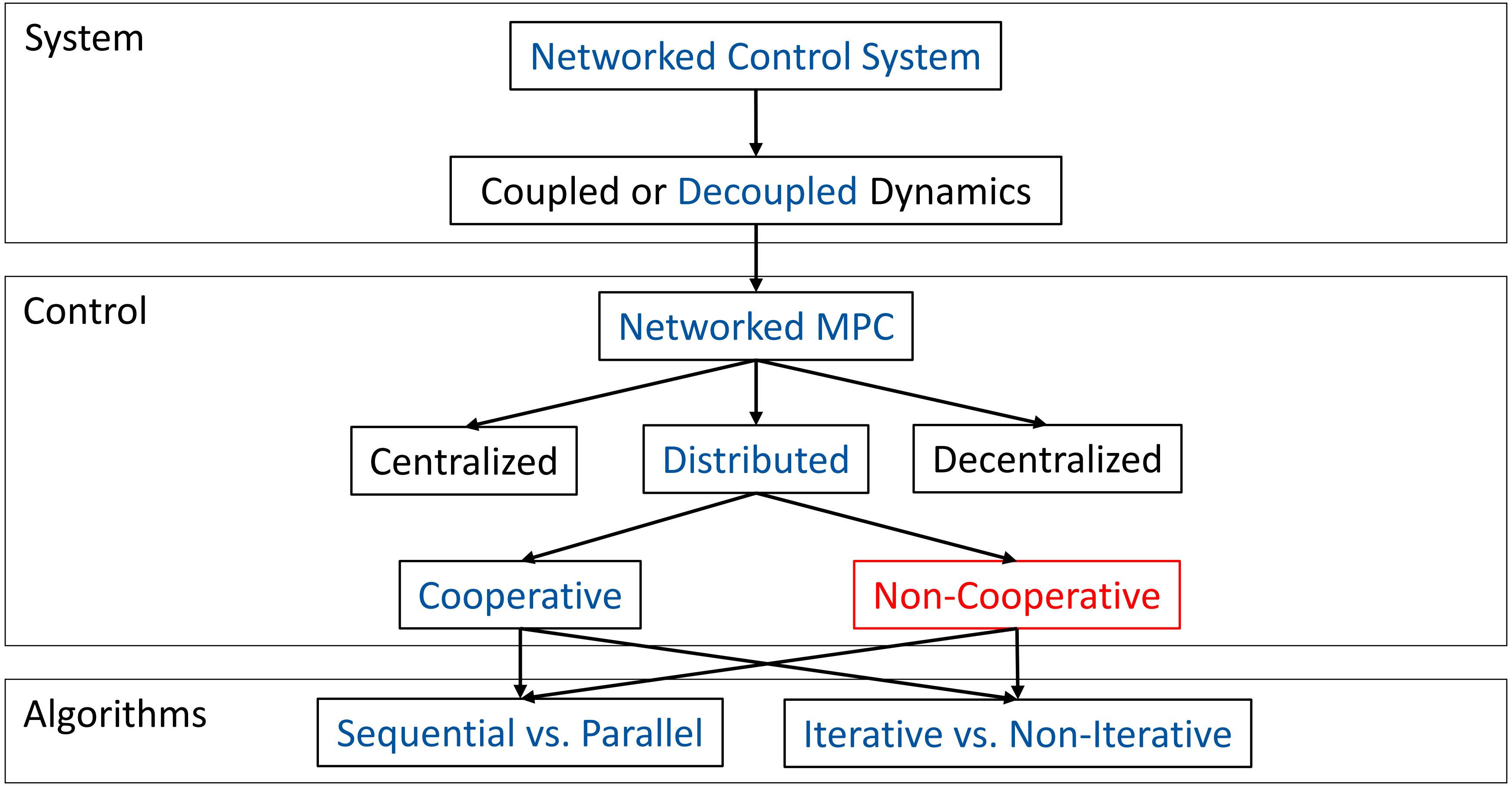
Perspective of agent 1

Perspective of agent 2



- Prediction consistency of plans is essential property

NCS Classification



Non-cooperative DMPC – pedestrian walking

- Objective J
  - Follow the line (minimize distance of your positions to markers on rope)

- Constraints
  - Model: $f(\mathbf{x}, \mathbf{u})$ (position is integral of velocity)
    - Position: states $\mathbf{x}$
    - Velocity:
      - Forwards: constant, one step per time step
      - Sideways: given by input change $\Delta\mathbf{u}$
  - Input change: max. one step per time step to either side
  - States: must be collision-free (with obstacles, other pedestrians’ predicted positions)

Non-cooperative DMPC – pedestrian walking

- Parameters
  - Prediction horizon $H_p = 3$
  - Control horizon $H_u = 1$
  - Time step duration T_s

Non-cooperative DMPC – pedestrian walking

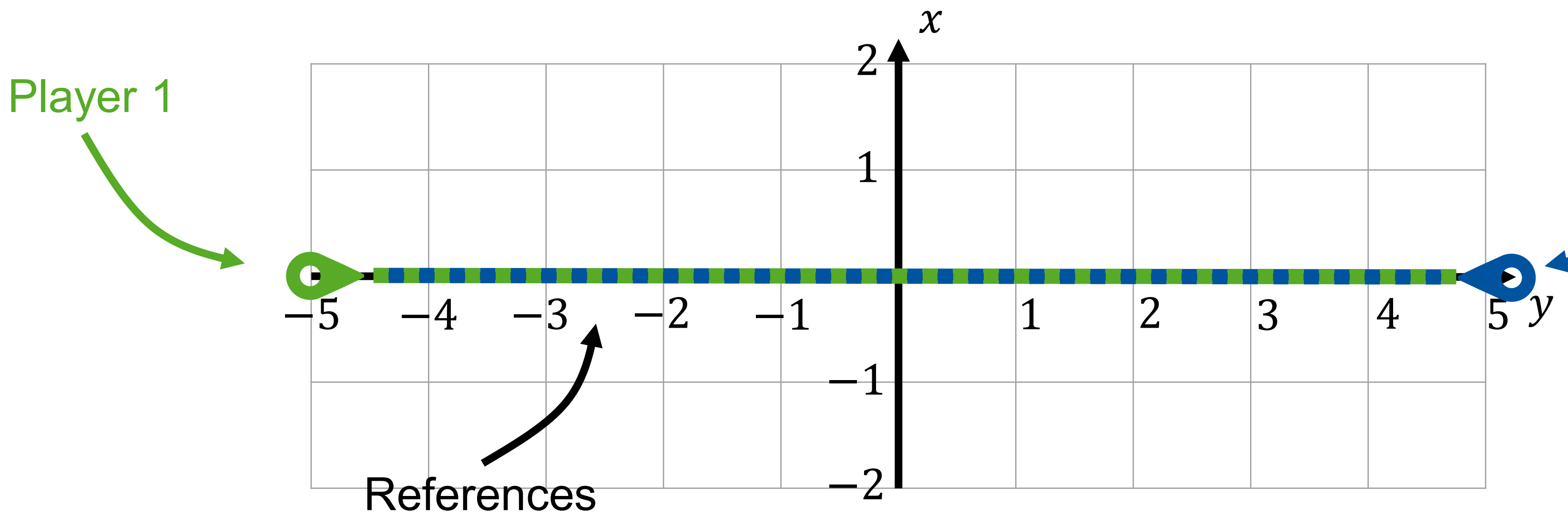
- Process for agent i:
  1. Form MPC optimization problem (coupling constraints)
  2. Optimize (generate plan): $\mathbf{x}_{\cdot|k}^{(i)}$
  3. Communicate plan
  4. Act (according to the first step of the plan): $\mathbf{u}_{k}^{(i)}$

Non-cooperative DMPC – pedestrian walking

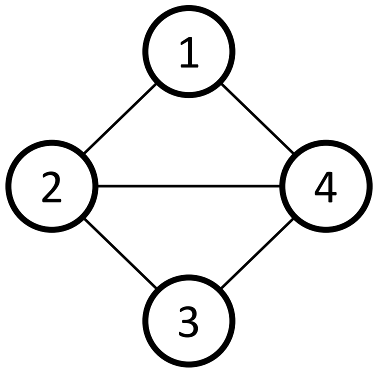
- How to get other pedestrians' (j) predicted positions $\mathbf{x}_{\cdot|k}^{(j)}$
- Communicated plan from previous time step $\mathbf{x}_{\cdot|k-1}^{(j)}$
  - First entry is from the past $(k-1)$
  - Entry for end of prediction horizon ($k+H_p$) is missing
- Predict using model, assume input $\Delta\mathbf{u}_{k+H_p-1}=0$

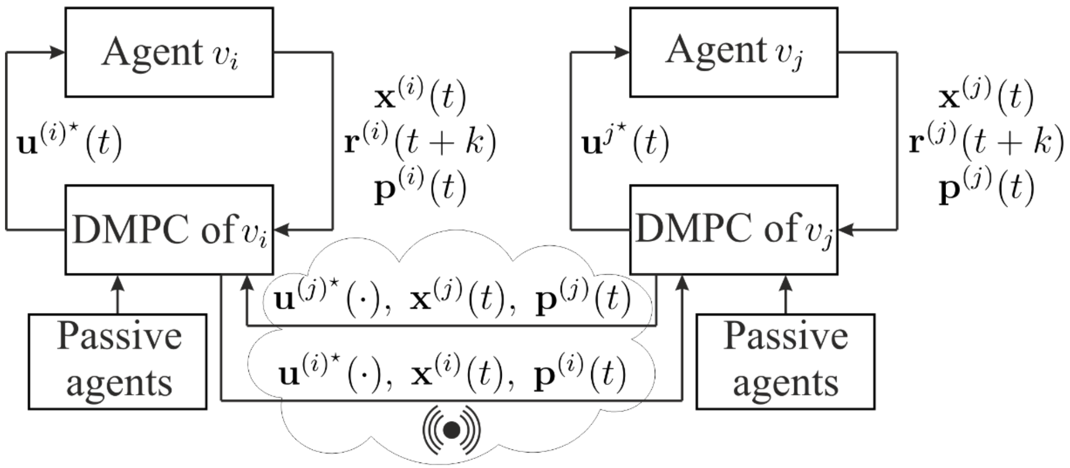
Non-cooperative DMPC – pedestrian walking

- Process for agent i:
  1. Form MPC optimization problem (coupling constraints)
  2. Optimize (generate plan): $\mathbf{x}^{(i)}_{\cdot|k}$
  3. Communicate plan
  4. Act (according to the first step of the plan): $\mathbf{u}^{(i)}_{k}$



Net-MPC: non-cooperative distributed MPC





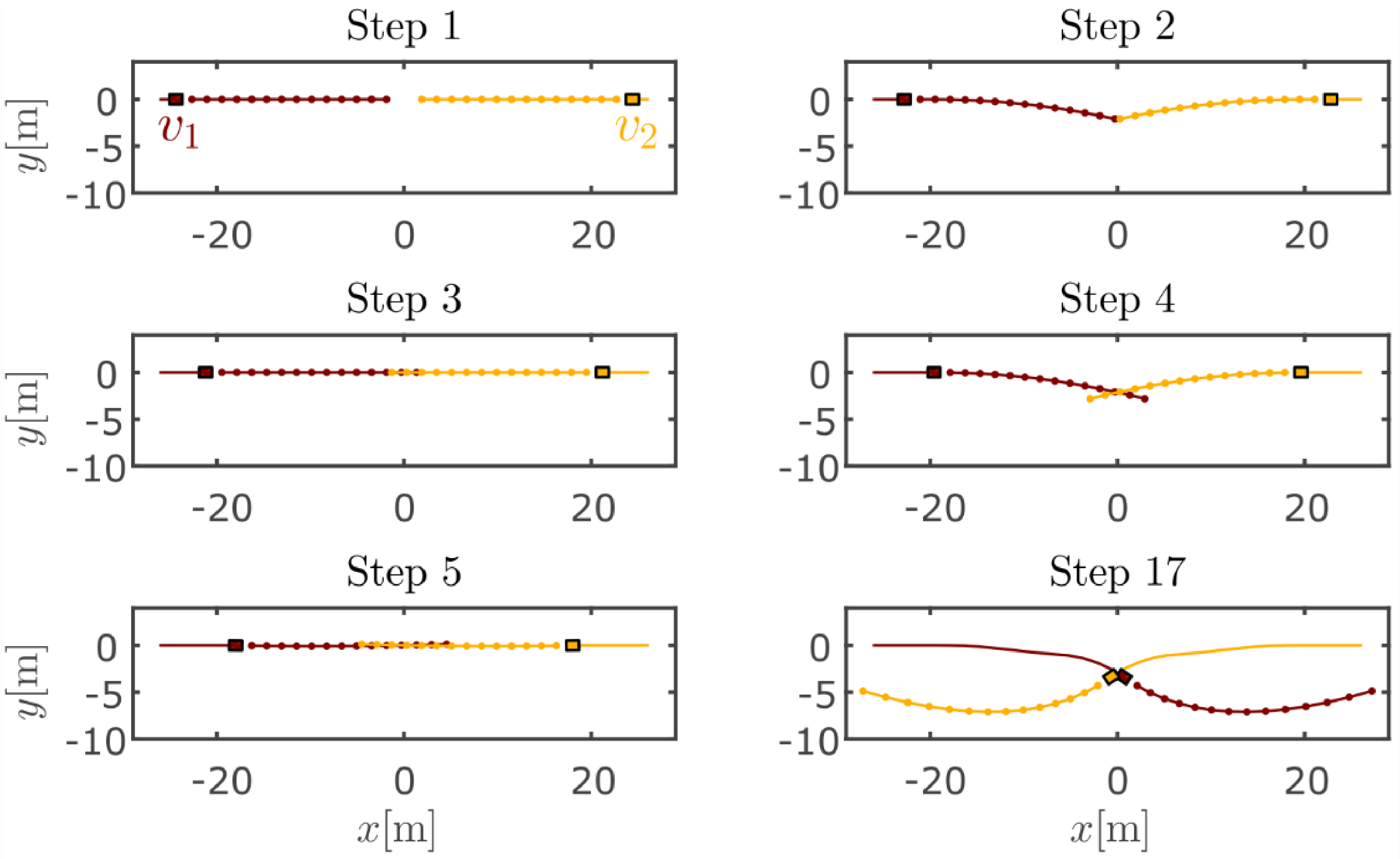
- Decomposition of centralized MPC into smaller optimization problems
- Consideration only of the own objective function, own constraints, and the coupling objectives and constraints with neighbors (greedy algorithm)
- Use of communicated optimized predictions from neighbors → time delay

Net-MPC: non-cooperative distributed MPC

Prediction consistency

- Prediction consistency means that the predictions $x^{(j)}(t+k)$, $k=1,\ldots,H_p$ used or computed in the optimization problem of an agent v_i at a time instance t for an agent v_j coincide with the predictions computed by agent v_j itself at the same time instance t
- Without satisfaction of this property, no guarantee for NCS-stability and -feasibility
- Non-Coop. DMPC does not satisfy the prediction consistency property due to the time delay in the communication

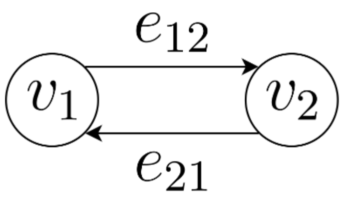
Net-MPC: non-cooperative distributed MPC



Net-MPC: non-cooperative distributed MPC

Coupling graph

- Coupling graph contains cycles
  - Consideration of exactly the same coupling
- Cycles in coupling graph lead to loss of prediction consistency property

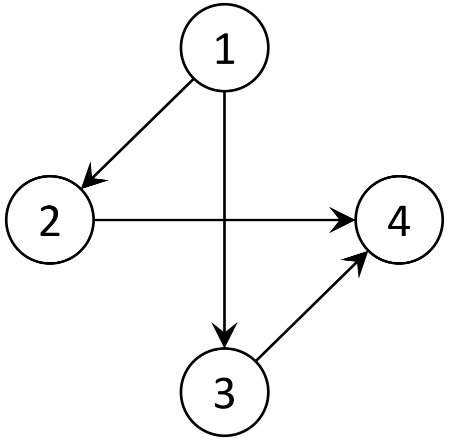


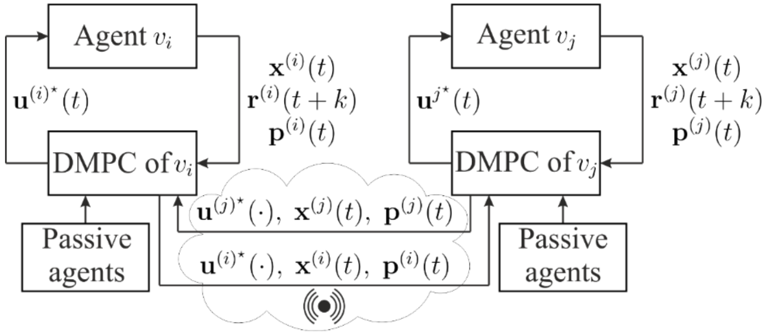
- Solutions:
  - Solve in sequence and iterate → high computation time
  - **Priority-Based Non-Cooperative Distributed MPC**

Networked model predictive control

- Flipped classroom
  - Group E should prepare a summary, ca. 15 minutes
- Reading
  - B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
    - Chapter 3, pages 51-77

Net-MPC: priority-based non-cooperative distributed MPC

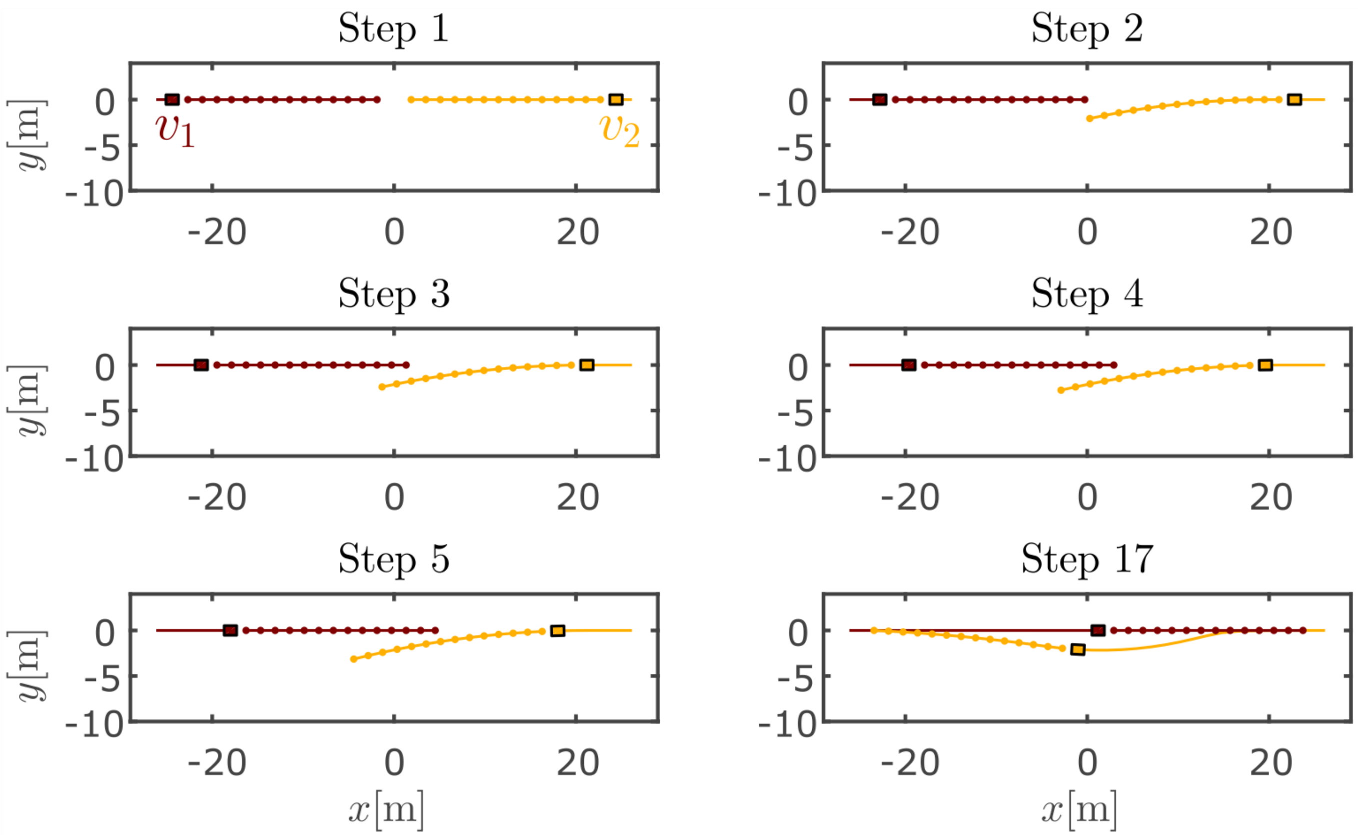




- Assign agents distinct priorities
- Lower priority value corresponds to a higher priority
- Passive agents have the highest priority
- Consideration of the own objective function, constraints, and only the coupling objectives and constraints with higher priority agents

Net-MPC: priority-based non-cooperative distributed MPC

- Example: Non-Coop. DMPC vs. PB-Non-Coop. DMPC



Informatik 11 Embedded Software

Net-MPC: priority-based non-cooperative distributed MPC

Coupling graph

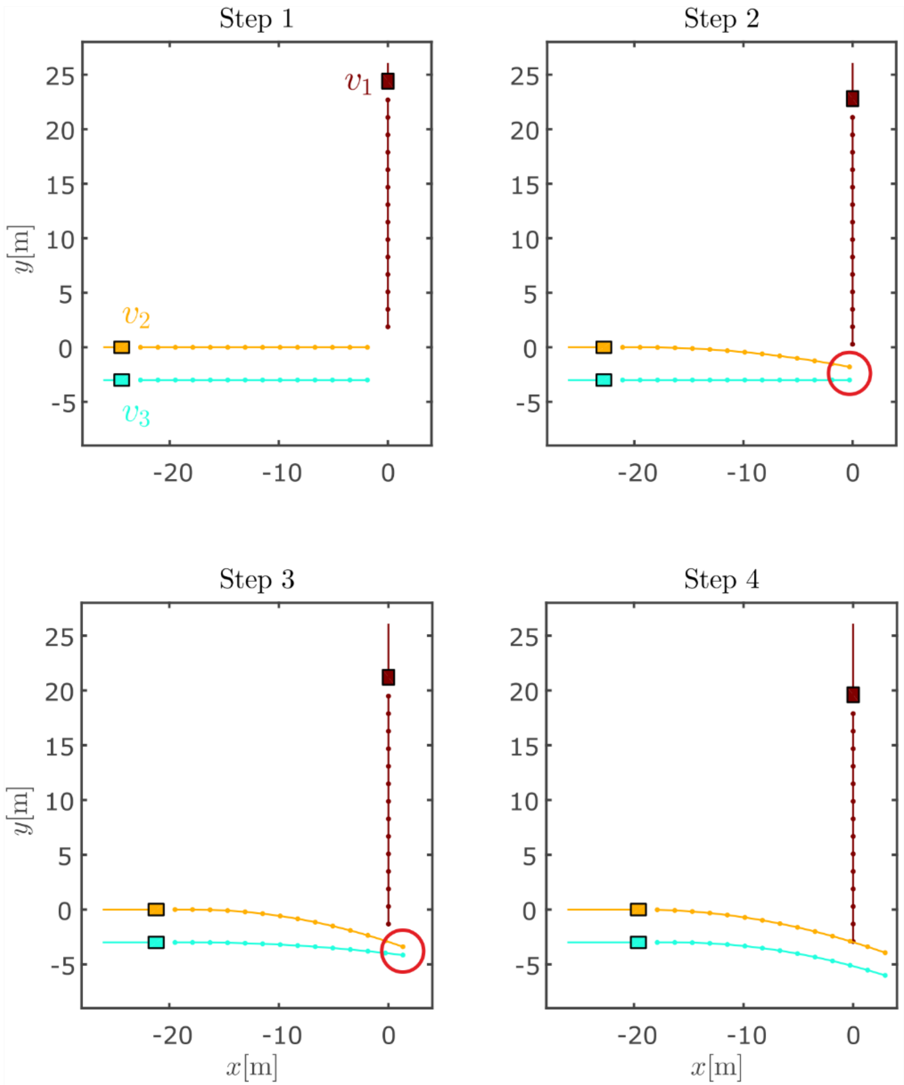
- Directed acyclic coupling graph (DAG) → proof using adjacency matrix

$$v_1 \xrightarrow{e_{12}} v_2$$

- Time delay of predictions of higher priority neighbors
  - Done in the case of time-invariant coupling topology and assuming bounded disturbances in higher priority neighbors
  - Loss of the prediction consistency property in the case of a time-variant coupling topology

Net-MPC: priority-based non-cooperative distributed MPC

- Vehicles move with the same velocity
- Priorities
  - v_1 First
  - v_2 Second
  - v_3 Third
- Time delay of one time step



- Solutions:
  - Infinite prediction horizon → not implementable in real-time
  - Incorporating a sequential algorithm into the PB-Non-Coop. DMPC strategy

Net-MPC: priority-based non-cooperative distributed MPC

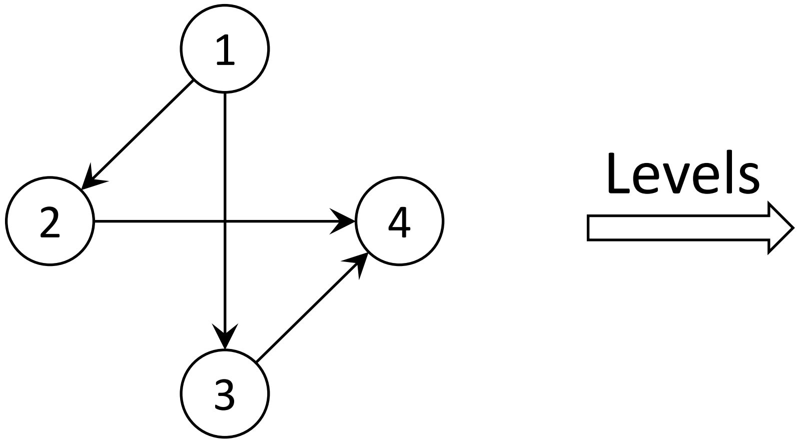
Sequential algorithm based on coupling graph

- Priorities generate a partial order
- Converting the partial order into a topological order
  - Renumbering the vertices of the coupling graph with their corresponding priorities
  - Valid sequence for solving the optimization problems
  - Possible if the coupling graph is DAG (proven)
- Our topological order is not unique → parallelization of subsets of the optimization problems possible

Net-MPC: priority-based non-cooperative distributed MPC

Algorithm to determine parallelizable vertices

- **Input:** adjacency matrix of a DAG
- **Output:** parallelizable vertices saved in a matrix $\mathbf{L}$

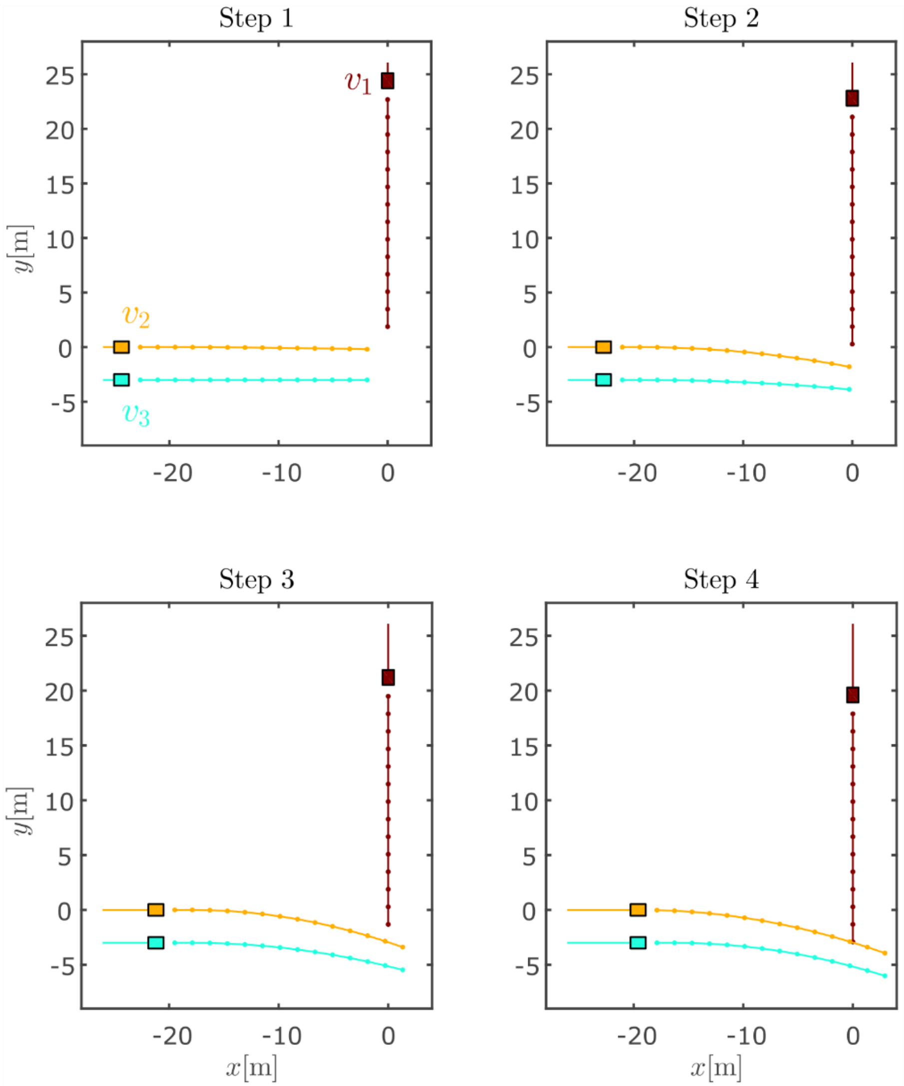


$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Rules of parallel and sequential computation:
  - Agents on the same level solve in parallel
  - If disturbances of agents on the first level are negligible, agents on the first and second level solve in parallel
  - Agents on level $3 \leq i \leq N_l$ solve sequentially after agents on level $i - 1$
  - The algorithm is executed repeatedly in the case of a change in the coupling graph

Net-MPC: priority-based non-cooperative distributed MPC

- Vehicles v_1 and v_2 solve in parallel
- Vehicle v_3 solves after v_1 and v_2
- Satisfaction of the prediction consistency property



Net-MPC: priority-based non-cooperative distributed MPC

Stability and feasibility discussion

- Main assumptions
  - A centralized MPC would generate a solution in each sample time that is NCS-stable, -feasible, and -optimal
  - Considering each agent as isolated from NCS, the solutions of PB-Non-Coop. DMPC are agent-stable, -feasible and -optimal
- Satisfaction of the prediction consistency property in any NCS even with time delays and time-variant coupling topology → Proof using mathematical induction
- Convergence after only one iteration of sequence

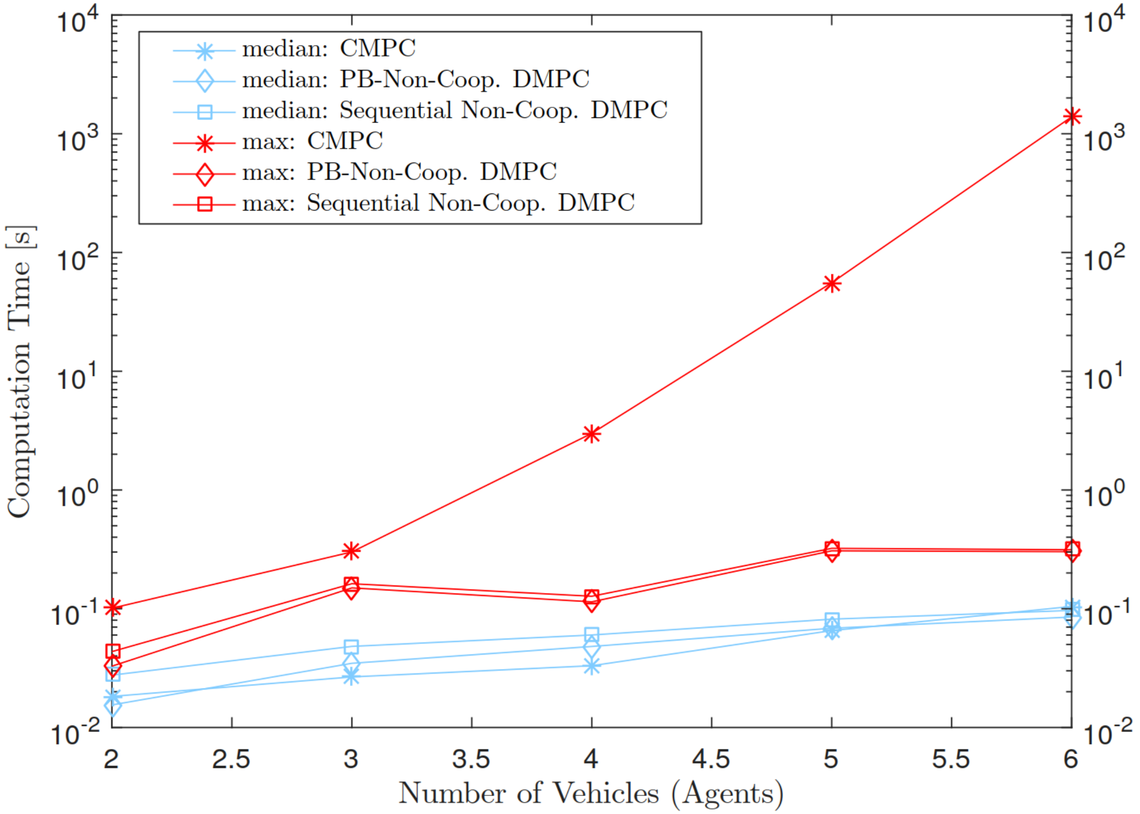
Net-MPC: comparison

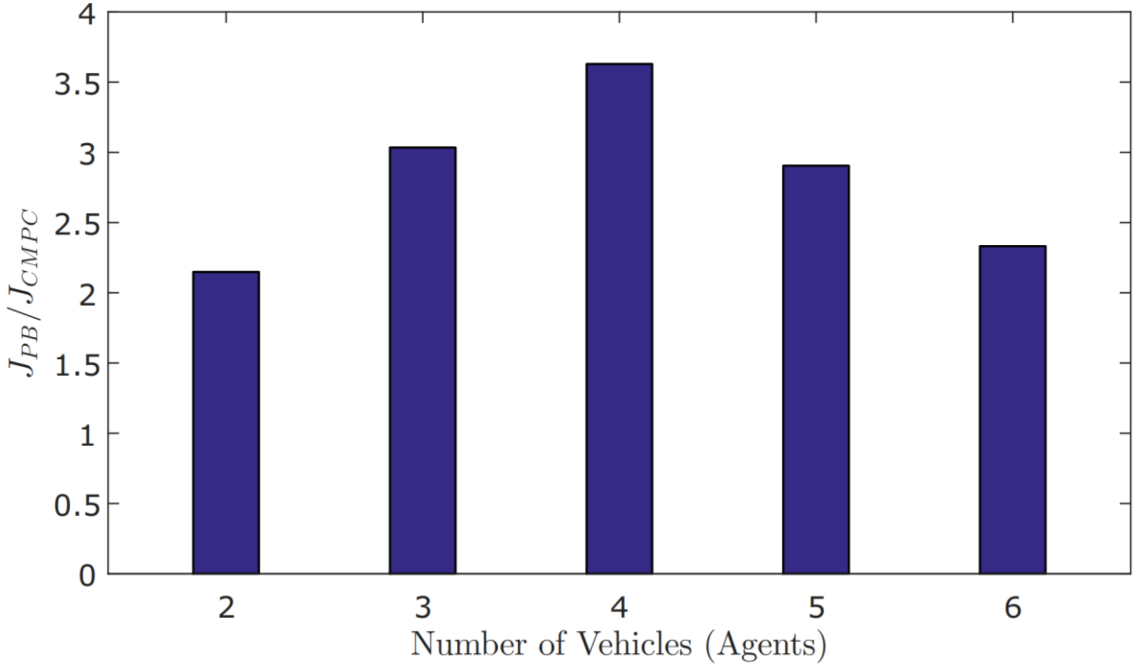
Optimization time of NCS in different Net-MPC strategies

Control Strategy	Optimization Time
CMPC	T_{ot}
Coop. DMPC	$T_{ot} = \max T_{ot}(v_i), \ \forall v_i \in \mathcal{V}$
PB-Non-Coop. DMPC	$T_{ot} = \max_{i \in level_{1,2}} T_{ot}(v_i) + \sum_{j=3}^{N_l} \max_{i \in level_j} T_{ot}(v_i)$
Sequential iterative DMPC	$T_{ot} = \sum_{1}^{N_{iter}} \sum_{i=1}^{N} T_{ot}(v_i)$

Net-MPC: comparison

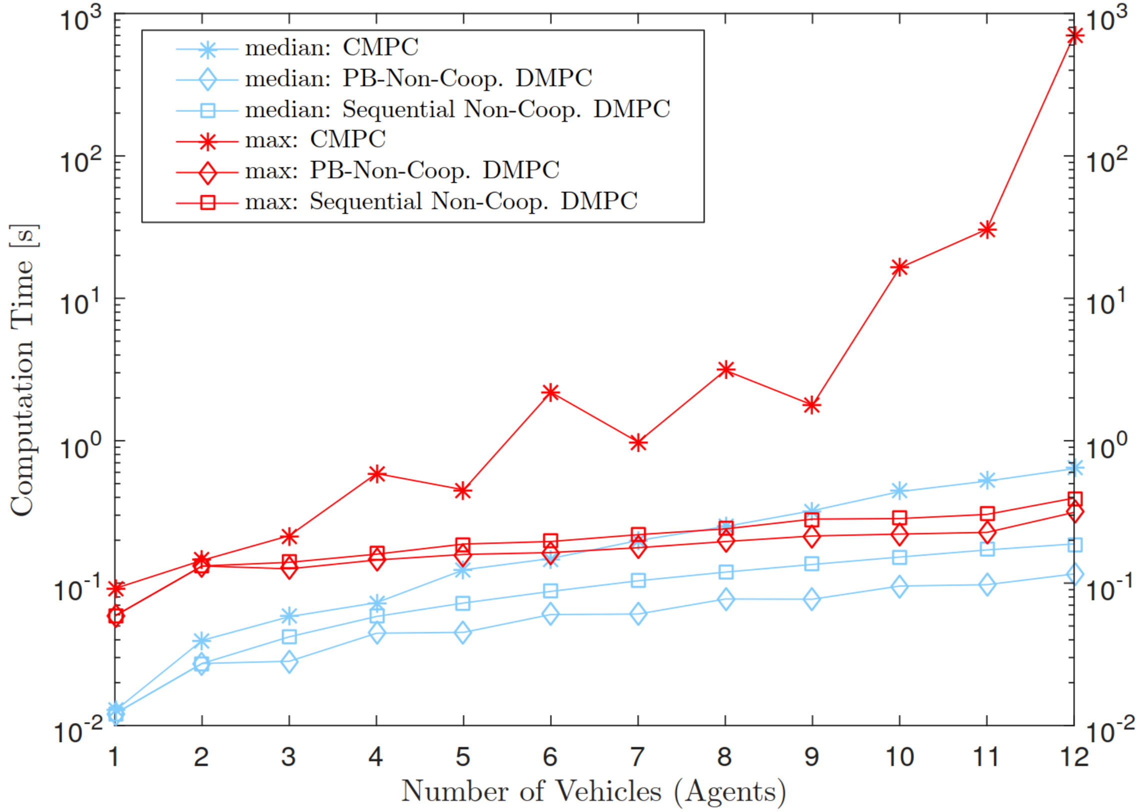
- Testing scenario N-circle, mean objective values normalized

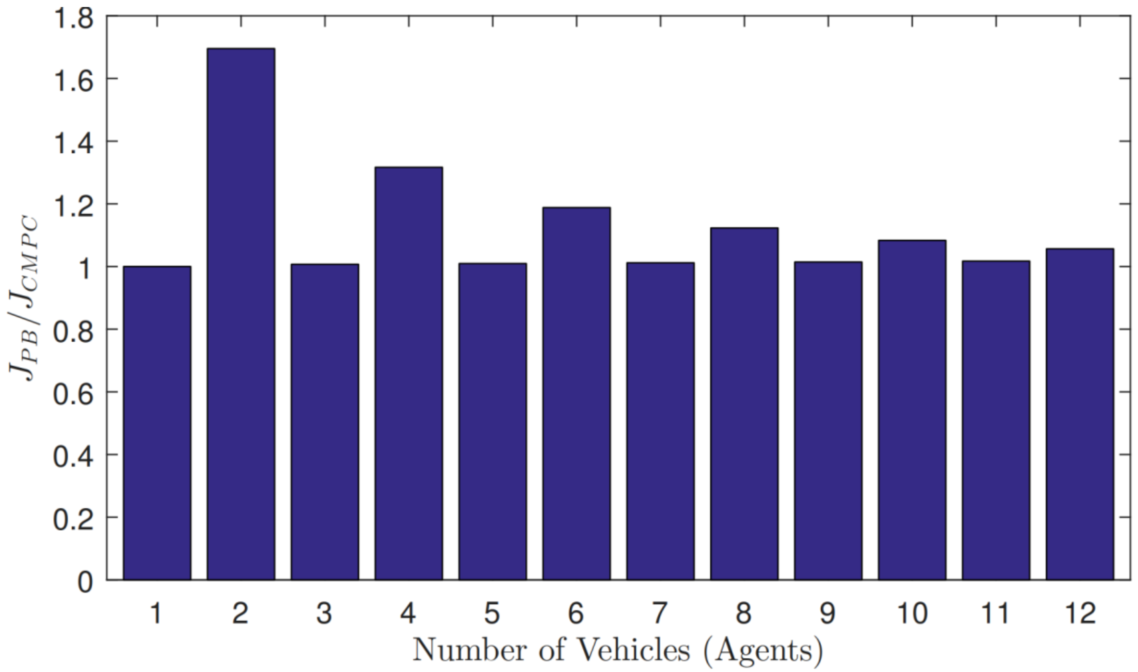




Net-MPC: comparison

- Testing scenario N-parallel, mean objective values normalized





Net-MPC: application to vehicle decision-making

- The application to vehicle decision-making combines high-level control methodologies and requirements of autonomous vehicles within a networked framework
- Networked vehicles are dynamically decoupled agents
- Couplings are functions of a subset of the states of a vehicle and a subset of the states of its neighbors
- (Active) agents are vehicles and passive agents are obstacles

- Net-MPC unifies collision detection, avoidance trajectory planning, and trajectory following in one optimization problem
- Efficient method to find a solution to the (in general non-convex) optimization problem of vehicle decision-making

Net-MPC: priority-based non-cooperative distributed MPC

Vehicle examples

- Video of simulation results
  - https://youtu.be/zS3UBx09O6M
- Video of experimental results
  - https://youtu.be/X2syxG5GI6g
- Further simulation results
  - https://youtu.be/XGql8FrjW6I
  - https://youtu.be/7sq3N8vwusA
  - https://youtu.be/kbooJFK52Fg

Net-MPC: priority-based non-cooperative distributed MPC

Summary

- Developed for NCS consisting of dynamically decoupled agents
- Coupling in the objective function or in the constraints
- Consideration of time-variant coupling

- Satisfaction of prediction consistency property
- NCS-stable and -feasible
- Reduction of the computation time

Moral machine

- Discussion of moral machine

Next Part

Guest lecture, then machine perception

- Guest lecture with Patrick Scheffe
- Machine perception with Alexandru Kampmann and Simon Schäfer