EECI-IGSC Course

Networked Model Predictive Control for Multi-Vehicle Decision-Making

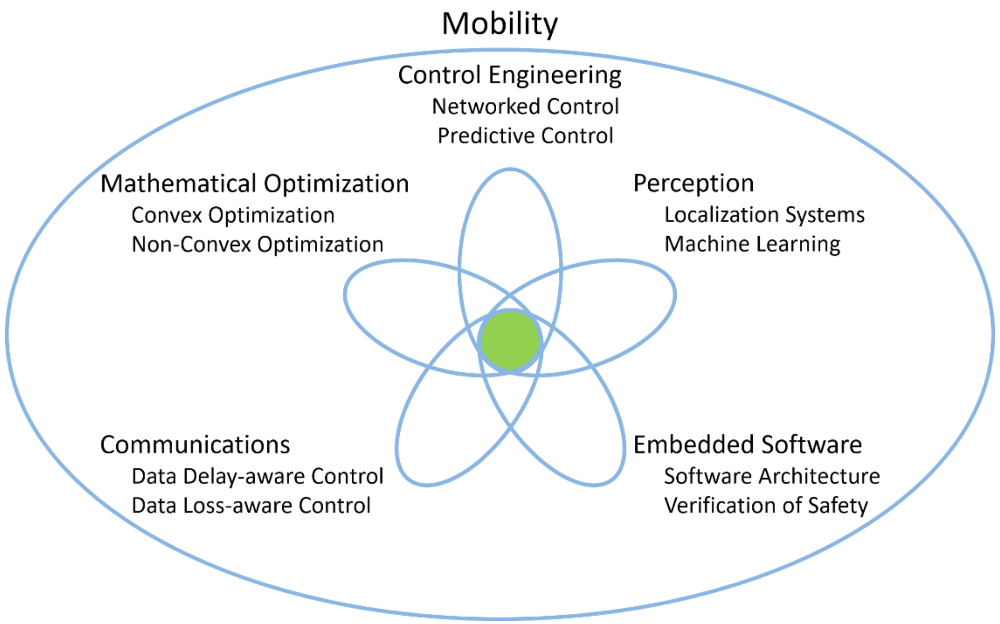
Dr.-Ing. Bassam Alrifaee | Patrick Scheffe, M. Sc.
2021

Part 3

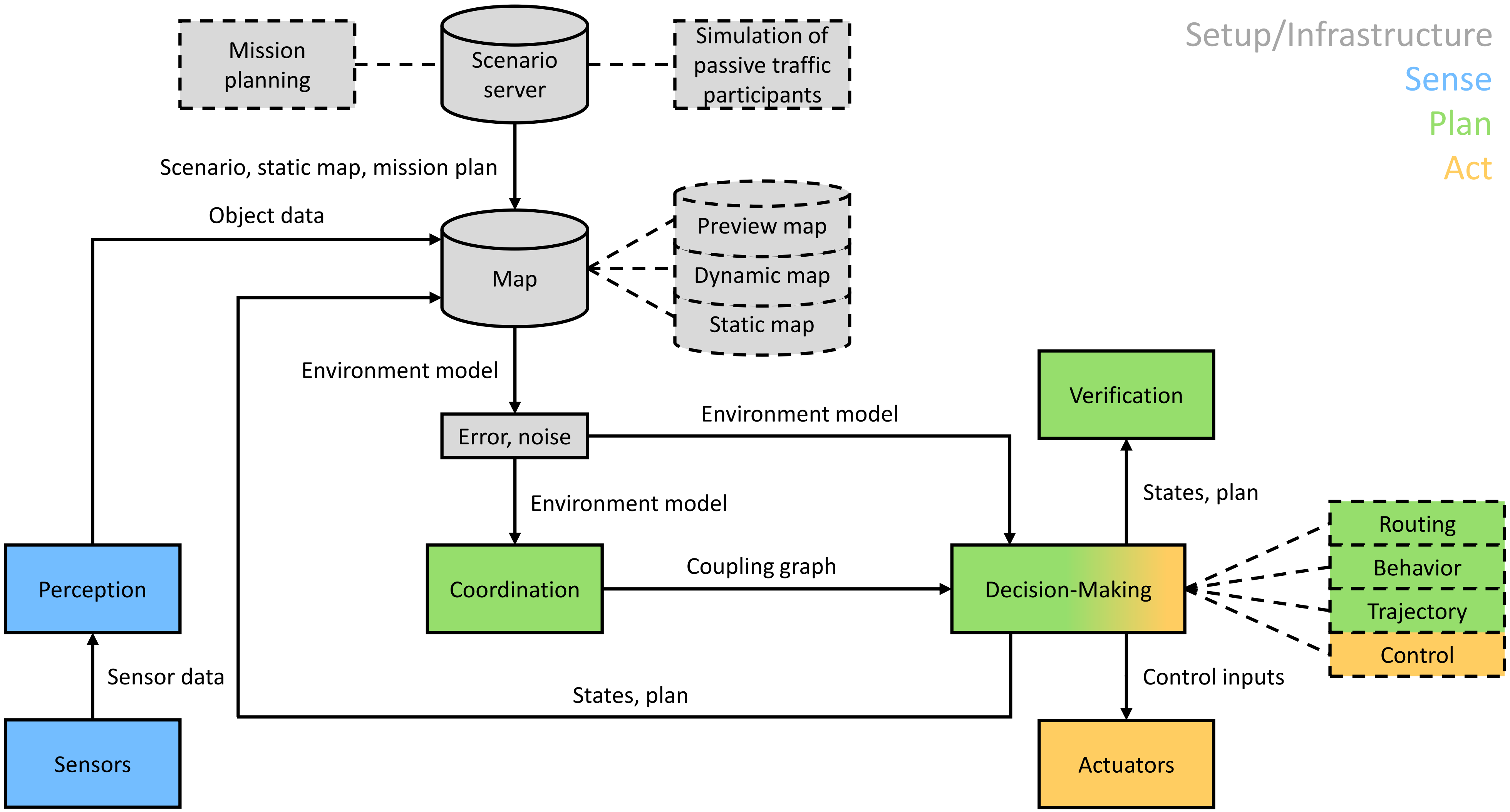
Control Engineering and Optimization

Course contents

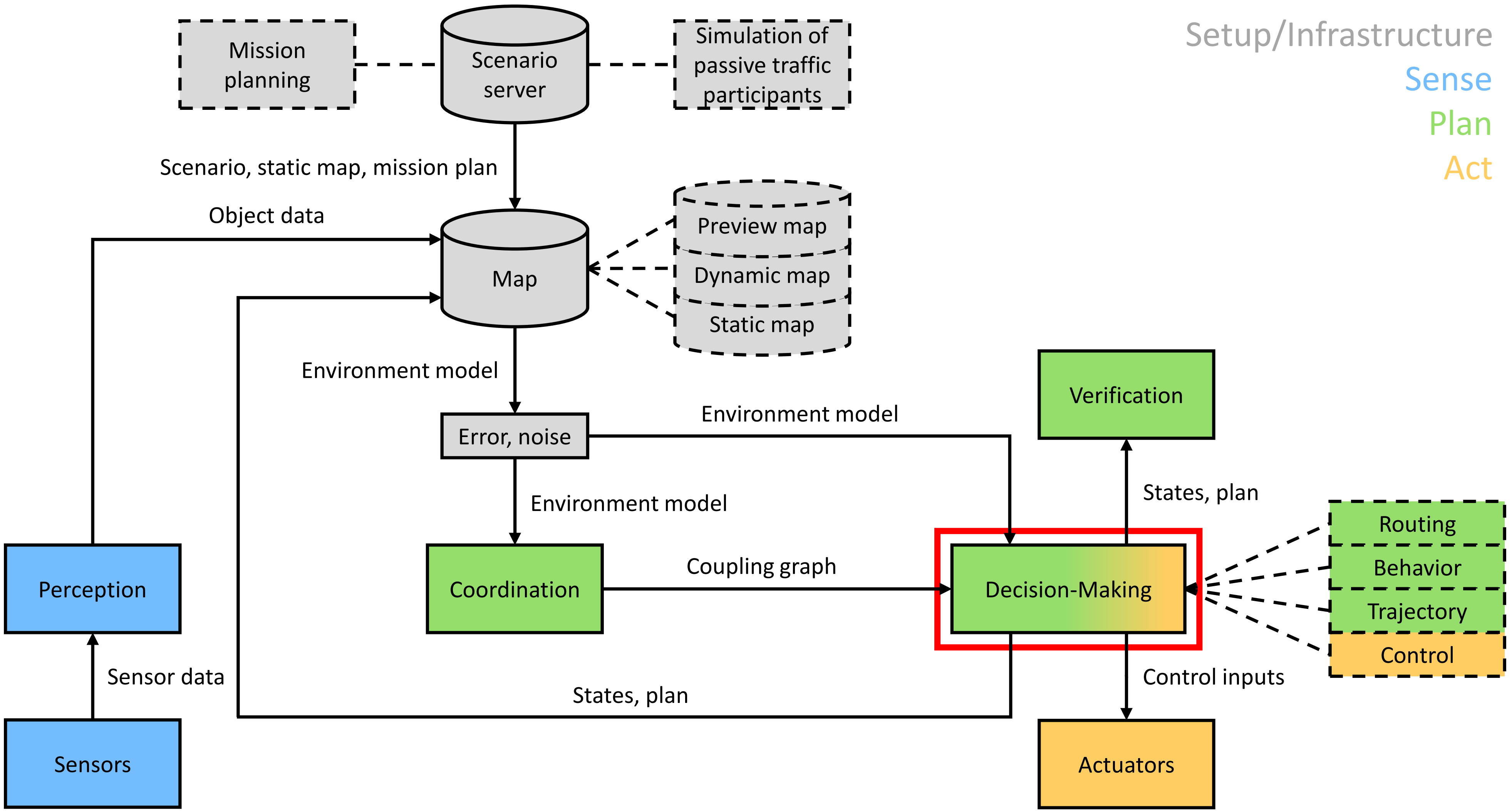
- Dynamic vehicle models
- Control and optimization
- Network and distribution
- Software architectures and testing concepts



CPM Lab architecture



CPM Lab architecture



Setup/Infrastructure
Sense
Plan
Act

Literature

- J. Maciejowski. Predictive Control with Constraints. Prentice Hall, 2002.
- S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
- R.C. Dorf and R.H. Bishop. Modern Control Systems. Prentice-Hall, 2008.

Further literature (1)

- F. Borrelli, A. Bemporad, and M. Morari. Predictive control for linear and hybrid systems. Cambridge University Press, 2017.
- Many

Further literature (2)

- B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
- B. Alrifaee, J. Maczijewski, and D. Abel. Sequential Convex Programming MPC for Dynamic Vehicle Collision Avoidance. In IEEE Conference on Control Technology and Applications (CCTA), 2017.
- B. Alrifaee, M. G. Mamaghani, and D. Abel. Centralized Non-Convex Model Predictive Control for Cooperative Collision Avoidance of Networked Vehicles. In IEEE International Symposium on Intelligent Control (ISIC), 2014.
- B. Alrifaee, K. Kostyszyn, and D. Abel. Model Predictive Control for Collision Avoidance of Networked Vehicles Using Lagrangian Relaxation. In IFAC Symposium on Control in Transportation Systems (CTS), 2016.
- B. Alrifaee and J. Maczijewski. Real-time Trajectory Optimization for Autonomous Vehicle Racing using Sequential Linearization. In IEEE Intelligent Vehicles Symposium (IV), 2018.
- B. Alrifaee. MATLAB Simulation of Networked Model Predictive Control for Vehicle Collision Avoidance, 2017. Available: https://doi.org/10.5281/zenodo.1252992

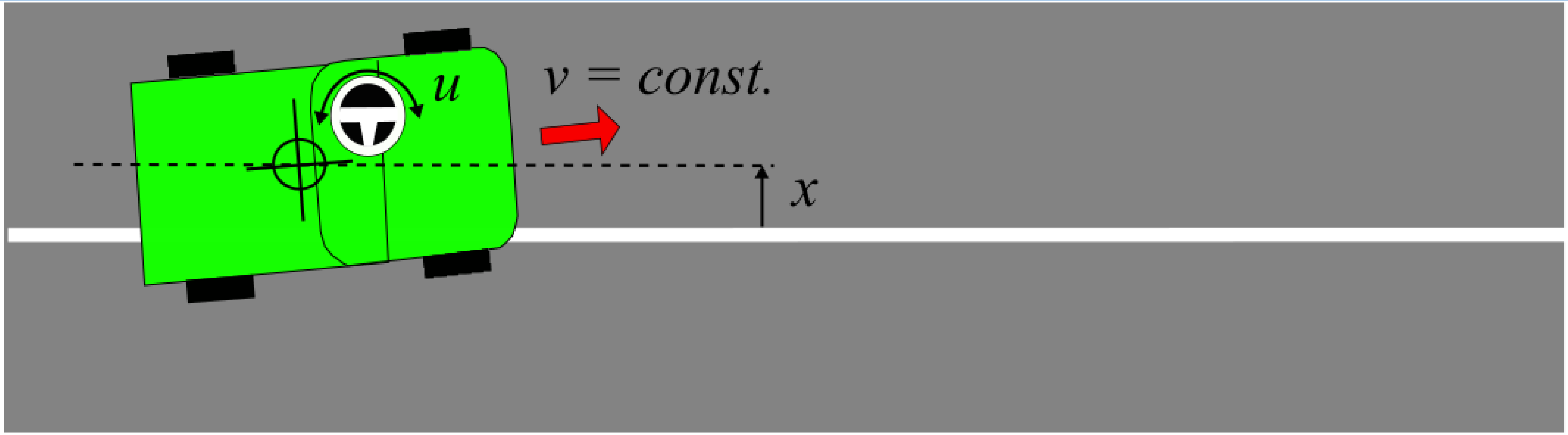
Model predictive control

- Reading
  - B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
    - Section 2.3, pages 8-14
    - Section 4.5.1.1, pages 100-101
  - J. Maciejowski. Predictive Control with Constraints. Prentice Hall, 2002. **[optional]**
    - Section 2.6.1, pages 53-56
    - Sections 2.6.2 and 2.6.3, pages 56-61 **[optional]**
    - Section 3.1.1, pages 74-77
    - Section 3.2.1, pages 81-84
    - Section 3.4, pages 97-99

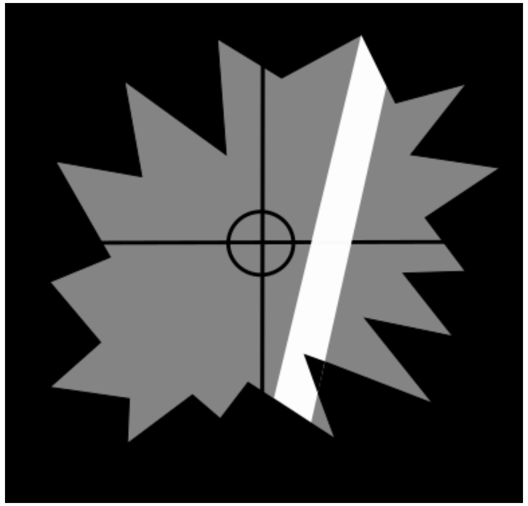
Model predictive control

- Pedestrian walking
  - Task
    - Follow the line
    - Act as little as possible
    - Consider physical and logical limits
  - Sense
    - Use sensing organs
    - Build an environment **model**
  - Plan
    - Use discrete model to anticipate your free movement for 5 steps
    - Consider your physical (body) limits
    - Consider physical and logical limits of your
      - Action
      - Position
      - Orientation
    - Plan your actions for 2 steps
  - Act
    - Move one step using some muscles of your body

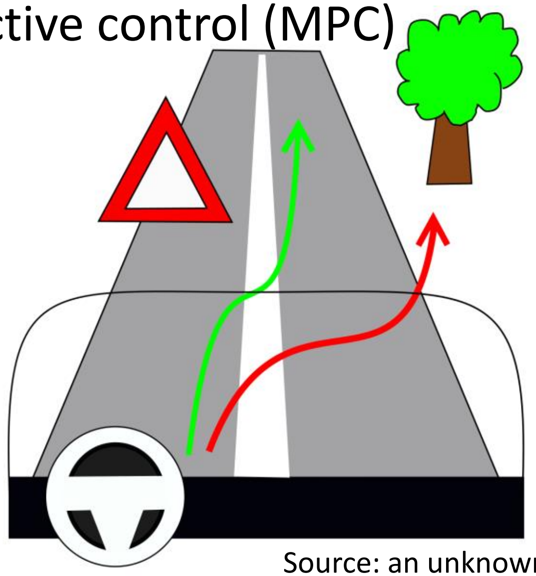
Model predictive control



classical control (PID)

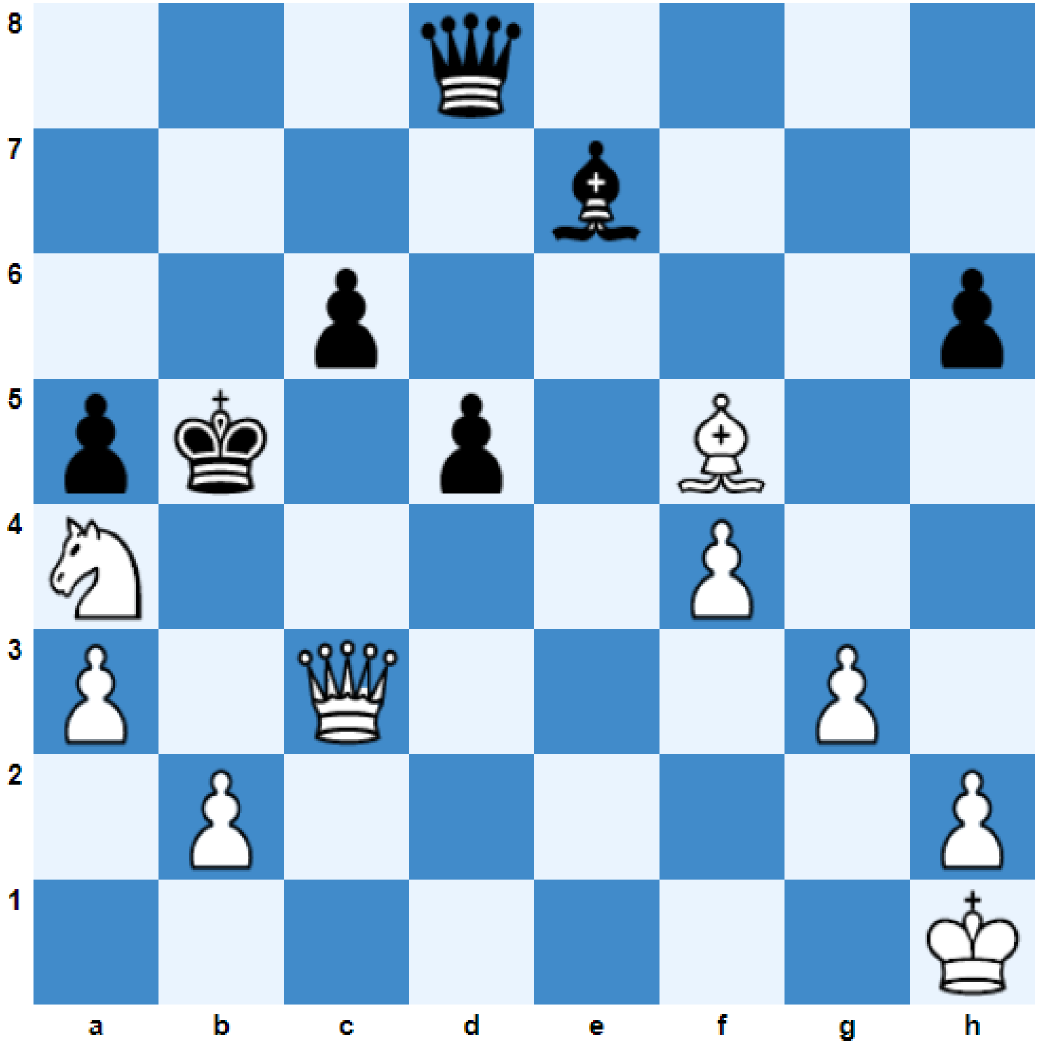


predictive control (MPC)



Source: an unknown PhD candidate

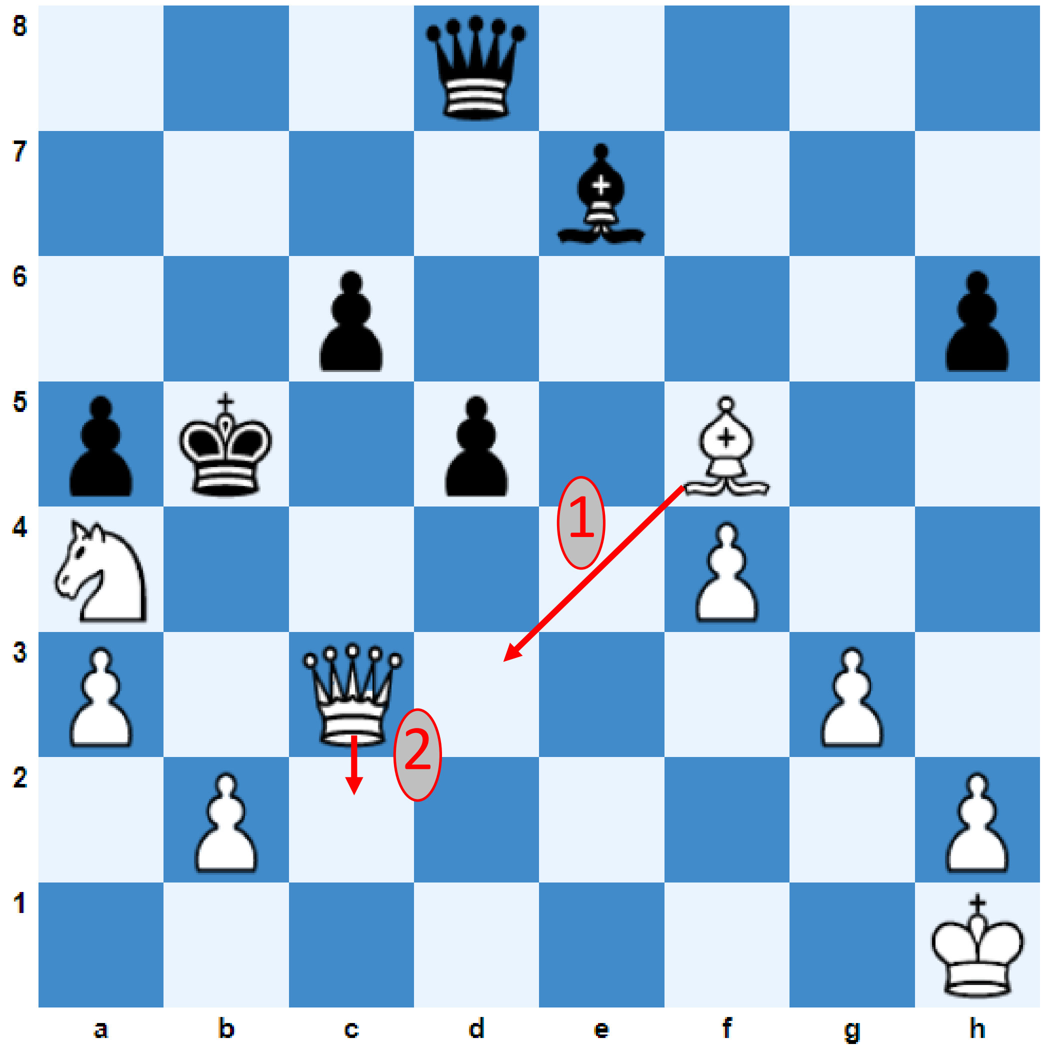
Model predictive control



schach-tipps.de, euroschach.de

Model predictive control

1. Bd3
2. Qc2

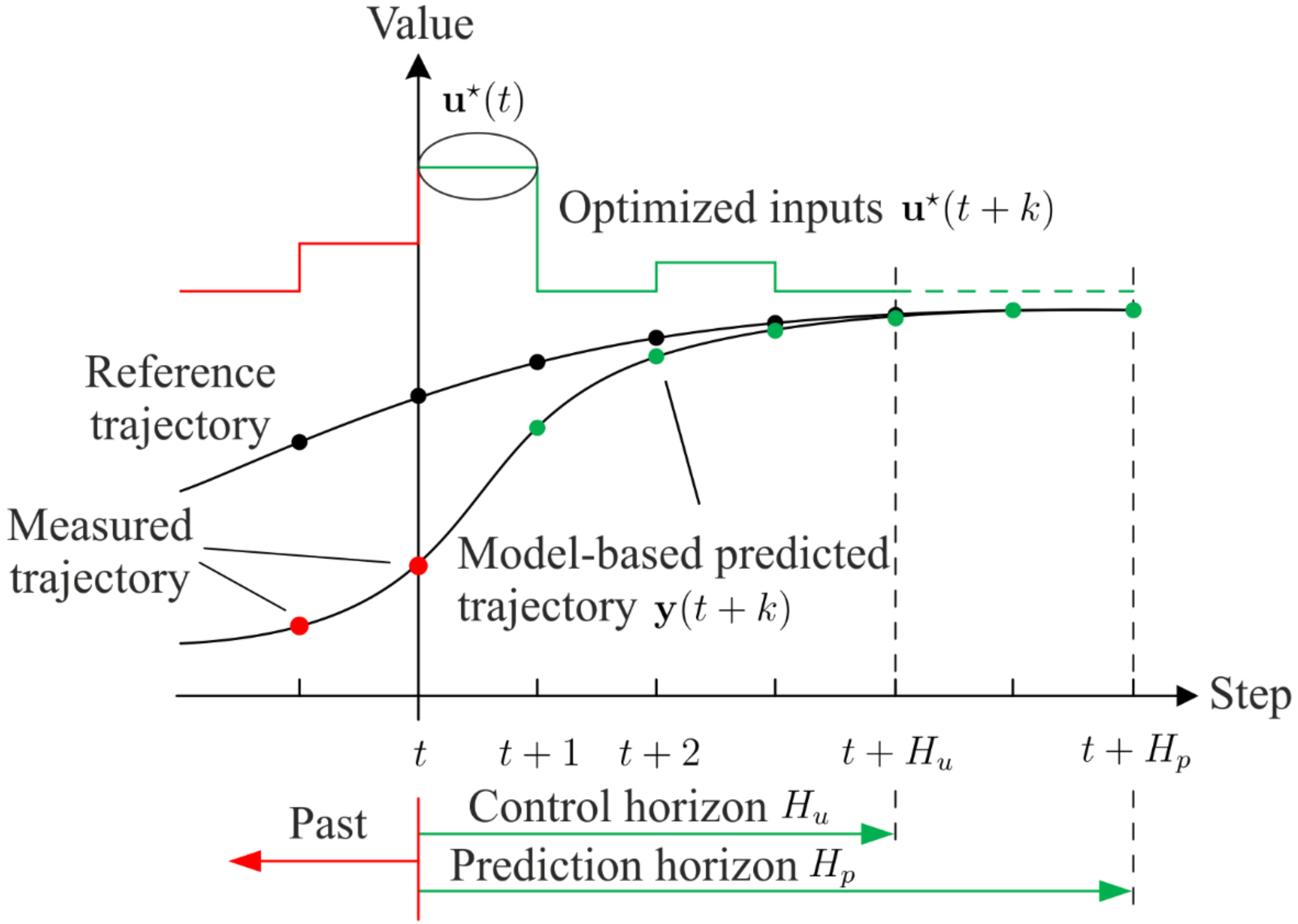


schach-tipps.de, euroschach.de

Model predictive control

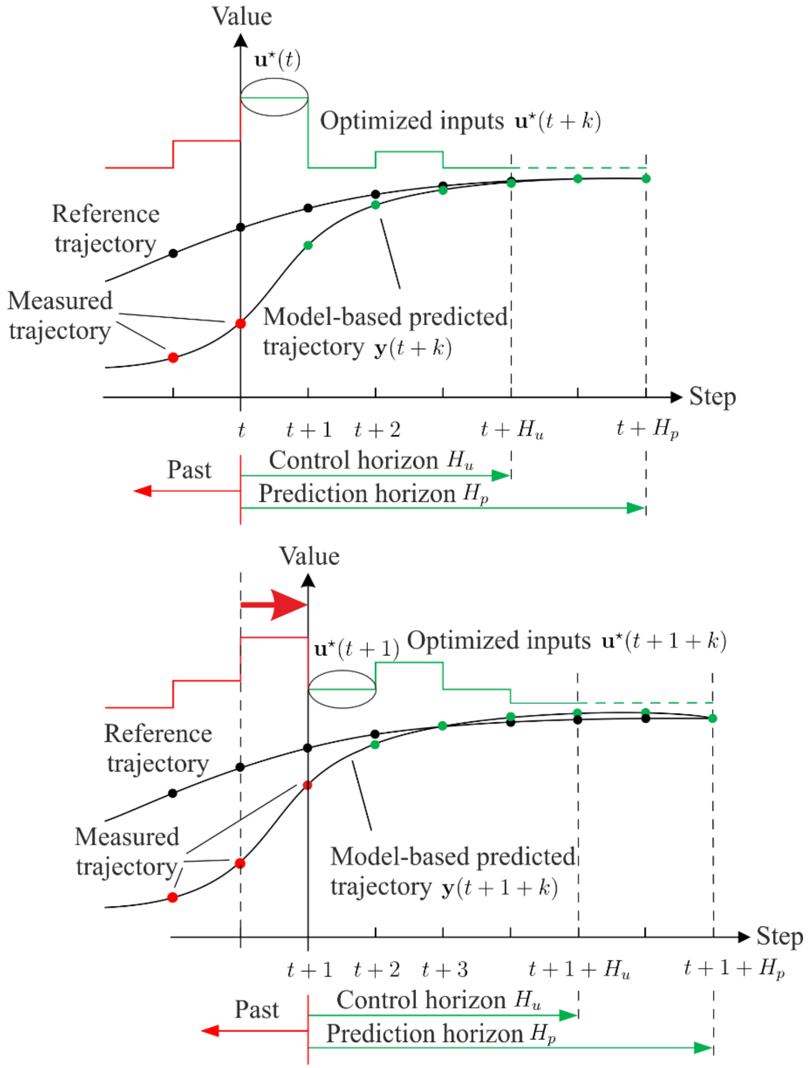
- Further examples
  - Automotive systems
  - Aeronautic industry
    - See SpaceX landing
  - Smart electricity grids
  - Drinking water networks
  - Financial engineering

Model predictive control



Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Model predictive control



Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Model predictive control

$$J^\star = \min_{\Delta \mathbf{u}(\cdot)} \sum_{k=1}^{H_p-1} l_x(\mathbf{x}(t+k), \mathbf{r}(t+k)) + l_{x_{H_p}}(\mathbf{x}(t+H_p), \mathbf{r}(t+H_p)) +$$
$$\sum_{k=0}^{H_u-1} l_u(\Delta \mathbf{u}(t+k))$$

subject to:

$$\mathbf{x}(t+1+k) = f(\mathbf{x}(t+k), \mathbf{u}(t+k)),\ k = 0, \dots, H_p - 1$$
$$\mathbf{x}(t+k) \in \mathcal{X},\ k = 1, \dots, H_p - 1$$
$$\mathbf{x}(t+H_p) \in \mathcal{X}_{H_p}$$
$$\mathbf{u}(t+k) \in \mathcal{U},\ k = 0, \dots, H_u - 1$$
$$\Delta \mathbf{u}(t+k) \in \Delta\mathcal{U},\ k = 0, \dots, H_u - 1$$

where

$$\Delta \mathbf{u}(t+k) = \mathbf{u}(t+k) - \mathbf{u}(t+k-1),\ k = 0, \dots, H_u - 1$$

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Model predictive control

- Result

$$\Delta\mathbf{u}^\star(\cdot) = \begin{pmatrix}\Delta\mathbf{u}^\star(t) & \dots & \Delta\mathbf{u}^\star(t+H_p-1)\end{pmatrix}^T$$

- Apply

$$\mathbf{u}^\star(t) = \mathbf{u}(t-1) + \Delta\mathbf{u}^\star(t), \quad \text{during} \quad [t, t+1)$$

$$J^\star = \min_{\Delta\mathbf{u}(\cdot)} \sum_{k=1}^{H_p-1} l_x(\mathbf{x}(t+k), \mathbf{r}(t+k)) + l_{x_{H_p}}(\mathbf{x}(t+H_p), \mathbf{r}(t+H_p)) + \sum_{k=0}^{H_u-1} l_u(\Delta\mathbf{u}(t+k))$$

subject to:

$$\mathbf{x}(t+1+k) = f(\mathbf{x}(t+k), \mathbf{u}(t+k)),\ k = 0, \dots, H_p - 1$$
$$\mathbf{x}(t+k) \in \mathcal{X},\ k = 1, \dots, H_p - 1$$
$$\mathbf{x}(t+H_p) \in \mathcal{X}_{H_p}$$
$$\mathbf{u}(t+k) \in \mathcal{U},\ k = 0, \dots, H_u - 1$$
$$\Delta\mathbf{u}(t+k) \in \Delta\mathcal{U},\ k = 0, \dots, H_u - 1$$

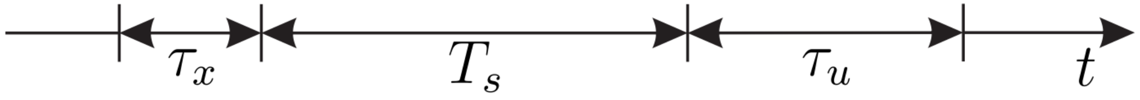
where

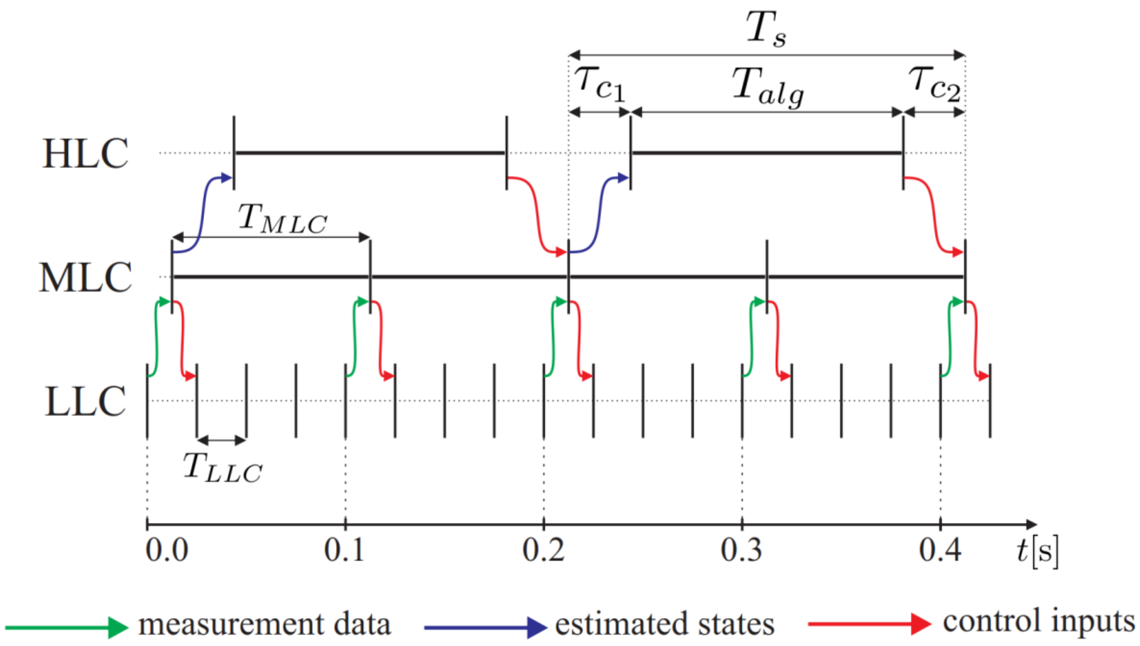
$$\Delta\mathbf{u}(t+k) = \mathbf{u}(t+k) - \mathbf{u}(t+k-1),\ k = 0, \dots, H_u - 1$$

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Model predictive control

Embedded implementation





$$J^\star = \min_{\Delta\mathbf{u}(\cdot)} \sum_{k=1}^{H_p-1} l_x(\mathbf{x}(t+k), \mathbf{r}(t+k)) + l_{x_{H_p}}(\mathbf{x}(t+H_p), \mathbf{r}(t+H_p)) + \sum_{k=0}^{H_u-1} l_u(\Delta\mathbf{u}(t+k))$$

subject to:

$$\mathbf{x}(t+1+k) = f(\mathbf{x}(t+k), \mathbf{u}(t+k)),\ k = 0, \dots, H_p - 1$$
$$\mathbf{x}(t+k) \in \mathcal{X},\ k = 1, \dots, H_p - 1$$
$$\mathbf{x}(t+H_p) \in \mathcal{X}_{H_p}$$
$$\mathbf{u}(t+k) \in \mathcal{U},\ k = 0, \dots, H_u - 1$$
$$\Delta\mathbf{u}(t+k) \in \Delta\mathcal{U},\ k = 0, \dots, H_u - 1$$

where

$$\Delta\mathbf{u}(t+k) = \mathbf{u}(t+k) - \mathbf{u}(t+k-1),\ k = 0, \dots, H_u - 1$$

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Model predictive control

- Pedestrian walking as MPC

Pedestrian walking		**MPC**	
Task	Follow the line	Reference	Objective function
	Act as little as possible	Control input	
	Consider physical and logical limits	Constraints	
Plan	Use discrete model to anticipate your free movement for 5 steps	Predict, prediction horizon	
	Consider your physical (body) limits	Model as a constraint	
	Consider physical and logical limits of your • Action • Position • Orientation	Inputs' and states' constraints	
	Plan your actions for 2 steps	Control horizon	

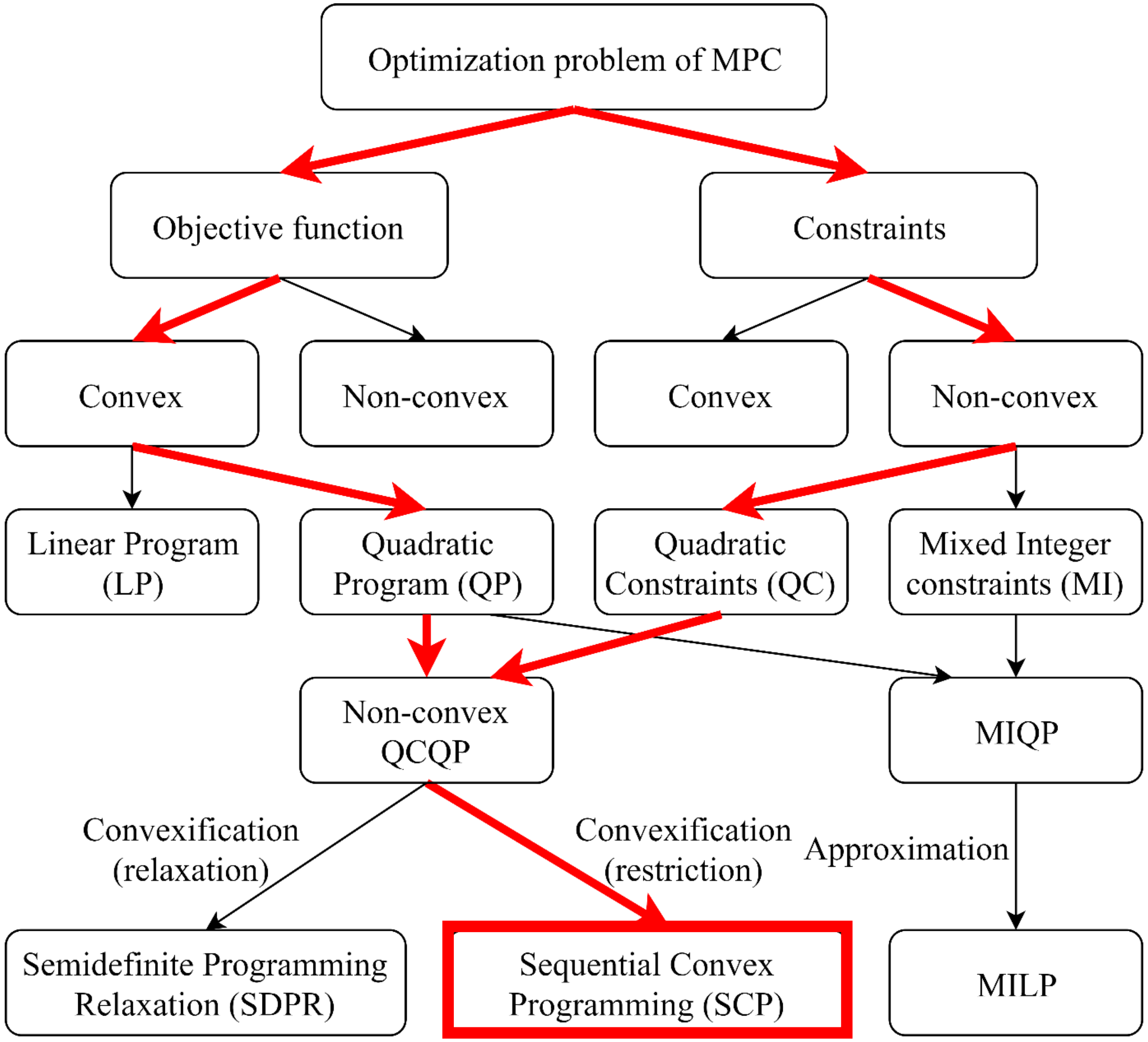
Sequential convex programming

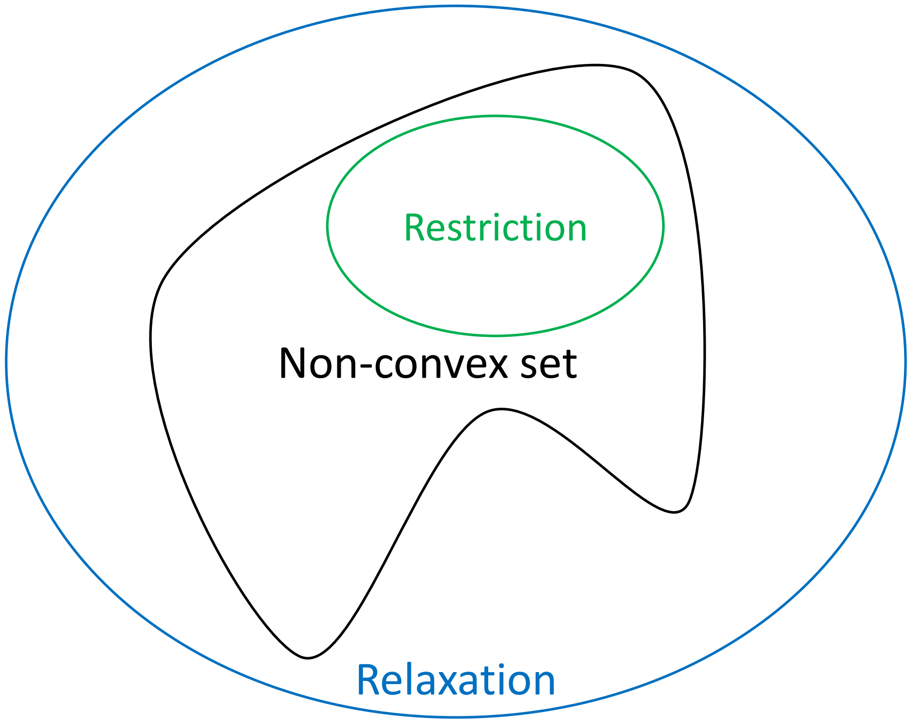
- Reading
  - B. Alrifaee. Networked Model Predictive Control for Vehicle Collision Avoidance. PhD thesis, RWTH Aachen University, 2017.
    - Section 4.5, the introduction part, pages 99-100
    - Section 4.5.3, pages 110-119

Sequential convex programming – concept

- "Solves" non-convex optimization problems
- Idea
  - Convexify non-convex parts of the objective function and constraints using convex approximations and preserve their convex parts
  - Solve convex problem
    - Global solution
    - Computationally efficient
  - New approximation around optimal solution
  - Repeat until convergence

Sequential convex programming





Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming

- Application to "solve" non-convex QCQPs
- QCQP (Quadratic Constrained Quadratic Program)

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T \mathbf{P}_0 \mathbf{x} + \mathbf{q}_0^T \mathbf{e} + r_0 \\ \text{subject to:} \quad & \mathbf{x}^T \mathbf{P}_i \mathbf{x} + \mathbf{q}_i^T \mathbf{e} + r_i \leq 0, \; i = 1, \dots, m \end{aligned}$$

- $\mathbf{P}_i$ is symmetric but not positive semi-definite
  - In the vehicle example, the collision avoidance constraints are concave functions

Sequential convex programming – concept

- “Solves” non-convex optimization problems
- Idea
  - Convexify non-convex parts of the objective function and constraints using convex approximations and preserve their convex parts
  - Solve convex problem
    - Global solution
    - Computationally efficient
  - New approximation around optimal solution
  - Repeat until convergence

Sequential convex programming – algorithm

Algorithm 1 SCP algorithm to "solve" the non-convex QCQP

Input: The non-convex QCQP
Output: Solution vector $\mathbf{x} \in \mathbb{R}^n$

1: $c := 1$ {Iteration counter}
2: Determine starting point $\mathbf{x}_c$
3: Compute the objective value of the non-convex program J_c using $\mathbf{x}_c$
4: Form a convex approximation of the non-convex parts of the inequality constraints using $\mathbf{x}_c$
5: Compute the optimal solution $\mathbf{x}_{c+1}$ of the resulting convex program
6: Compute the objective value of the non-convex program J_{c+1} using $\mathbf{x}_{c+1}$
7: **if** $J_c - J_{c+1} \leq \epsilon$ **then**
8:     **return** $\mathbf{x}_{c+1}$
9: **end if**
10: $c := c + 1$
11: **go to** $\boxed{4}$

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – algorithm

Starting point

- Use time-shifted solution from previous time step of MPC
- No feasible solution
  - May be bad local minimum
  - Try other starting points

Algorithm 1 SCP algorithm to "solve" the non-convex QCQP

Input: The non-convex QCQP
Output: Solution vector $\mathbf{x} \in \mathbb{R}^n$

1: $c := 1$ {Iteration counter}
2: Determine starting point $\mathbf{x}_c$
3: Compute the objective value of the non-convex program J_c using $\mathbf{x}_c$
4: Form a convex approximation of the non-convex parts of the inequality constraints using $\mathbf{x}_c$
5: Compute the optimal solution $\mathbf{x}_{c+1}$ of the resulting convex program
6: Compute the objective value of the non-convex program J_{c+1} using $\mathbf{x}_{c+1}$
7: **if** $J_c - J_{c+1} \leq \epsilon$ **then**
8:     **return** $\mathbf{x}_{c+1}$
9: **end if**
10: $c := c + 1$
11: **go to** 4

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – algorithm

Convex approximation (local)

- Linearization of quadratic constraints around $\mathbf{x}_c$
  - Best convex approximation of a concave function is its affine approximation*
  - Affine function lies above the concave function at all points, i.e., the affine function is globally upper bound on the concave function
  - Resulting program is a restriction of the original one

Convex approximation (local)
Linearization of quadratic constraints around $\mathbf{x}_c$
Best convex approximation of a concave function is its affine approximation*
Affine function lies above the concave function at all points, i.e., the affine function is globally upper bound on the concave function
Resulting program is a restriction of the original one

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

* D. Zwick. Best Approximation by Convex Functions. The American Mathematical Monthly, 94(6):528–534, 1987. Link

Sequential convex programming – algorithm

Solve convex sub-problem

- Fast
- Iterating around the new solution produces solutions with lower objective values, because convex; and therefore, of original program

Algorithm 1 SCP algorithm to "solve" the non-convex QCQP

Input: The non-convex QCQP
Output: Solution vector $\mathbf{x} \in \mathbb{R}^n$

1: $c := 1$ {Iteration counter}
2: Determine starting point $\mathbf{x}_c$
3: Compute the objective value of the non-convex program J_c using $\mathbf{x}_c$
4: Form a convex approximation of the non-convex parts of the inequality constraints using $\mathbf{x}_c$
5: Compute the optimal solution $\mathbf{x}_{c+1}$ of the resulting convex program
6: Compute the objective value of the non-convex program J_{c+1} using $\mathbf{x}_{c+1}$
7: **if** $J_c - J_{c+1} \leq \epsilon$ **then**
8:     **return** $\mathbf{x}_{c+1}$
9: **end if**
10: $c := c + 1$
11: **go to** 4

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – algorithm

Progress evaluation

- Decreases in objective value of original problem indicates progress

Algorithm 1 SCP algorithm to "solve" the non-convex QCQP

Input: The non-convex QCQP
Output: Solution vector $\mathbf{x} \in \mathbb{R}^n$

1: $c := 1$ {Iteration counter}
2: Determine starting point $\mathbf{x}_c$
3: Compute the objective value of the non-convex program J_c using $\mathbf{x}_c$
4: Form a convex approximation of the non-convex parts of the inequality constraints using $\mathbf{x}_c$
5: Compute the optimal solution $\mathbf{x}_{c+1}$ of the resulting convex program
6: Compute the objective value of the non-convex program J_{c+1} using $\mathbf{x}_{c+1}$
7: **if** $J_c - J_{c+1} \leq \epsilon$ **then**
8: **return** $\mathbf{x}_{c+1}$
9: **end if**
10: $c := c + 1$
11: **go to** $\boxed{4}$

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – discussion

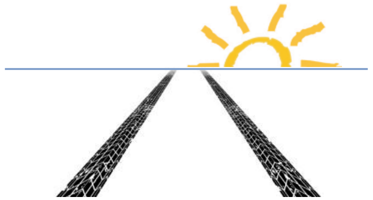
- Fast (in our applications <100ms)
- Locally optimal solutions of QCQP problem
- SCP sub-problem is restriction of QCQP
  - SCP feasible implies QCQP feasible
  - Feasibility issues, SCP too restrictive
  - Solution: slack variable

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Model predictive control – vehicle example

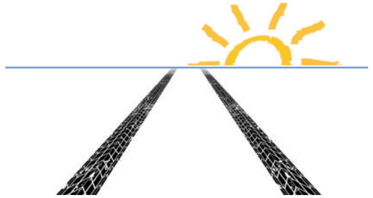
Objective Function

- Follow a predefined reference trajectory
  - Minimize the distances between the vehicle position and the reference trajectory



Prediction Horizon

- Only accept small input changes
  - Keep the steering angle changes as small as possible



Control Horizon

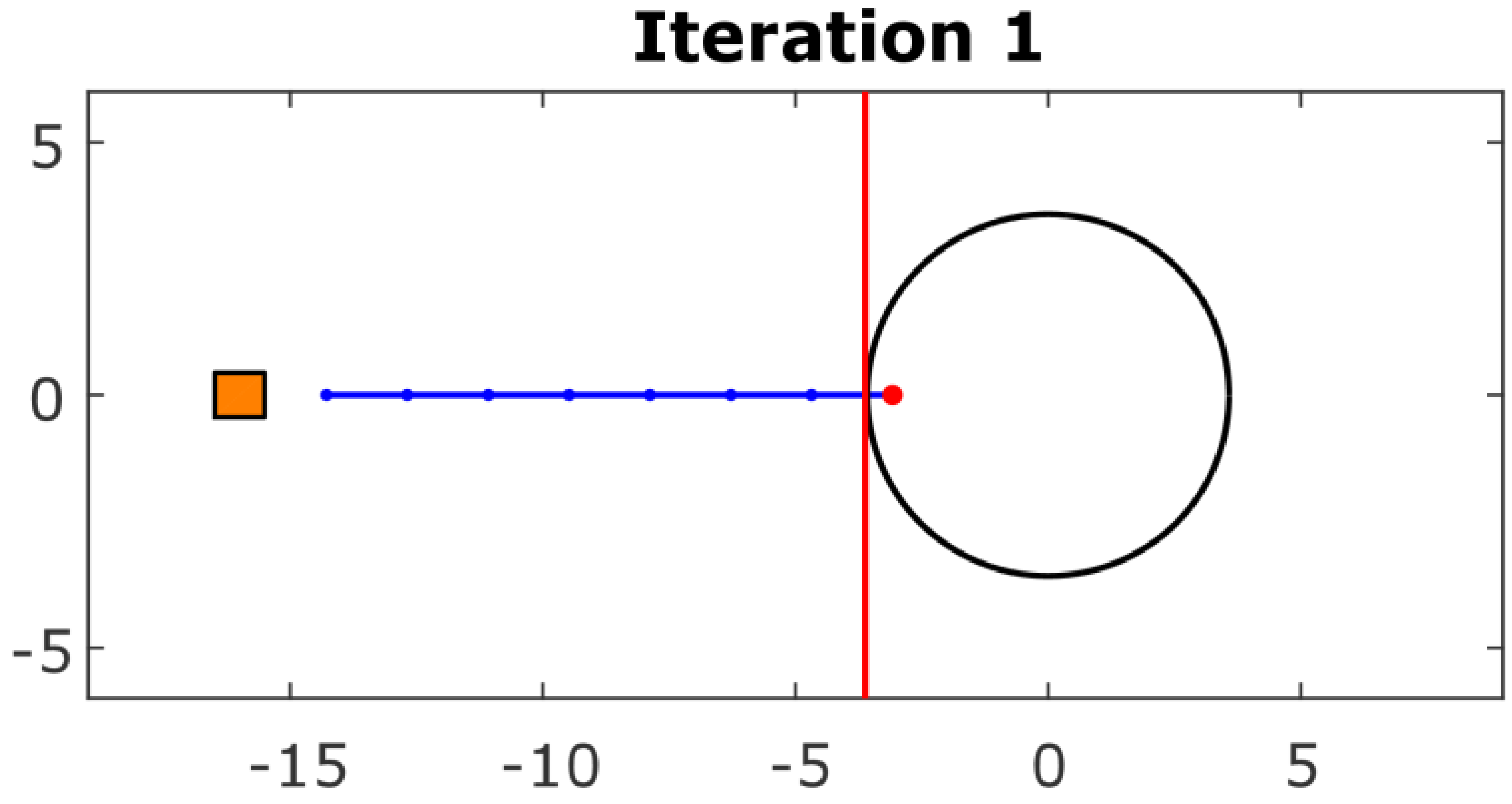
Model predictive control – vehicle example

Constraints

- Input constraints, do not exceed
  - A maximum steering angle
  - A maximum change of the steering angle per time step
  - A maximum lateral acceleration

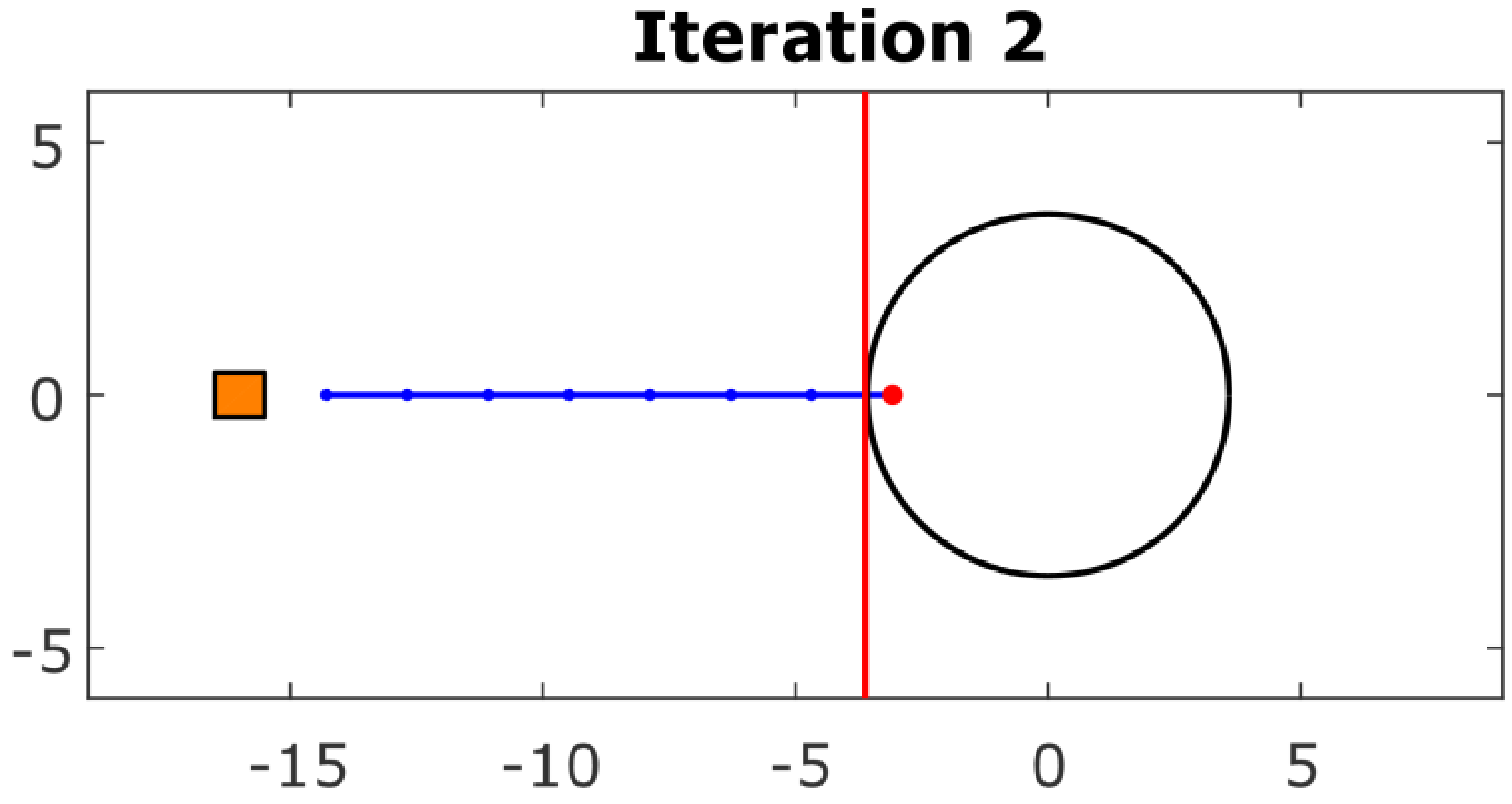
- Collision avoidance constraints
  - Do not collide with other vehicles or obstacles

Sequential convex programming – vehicle example



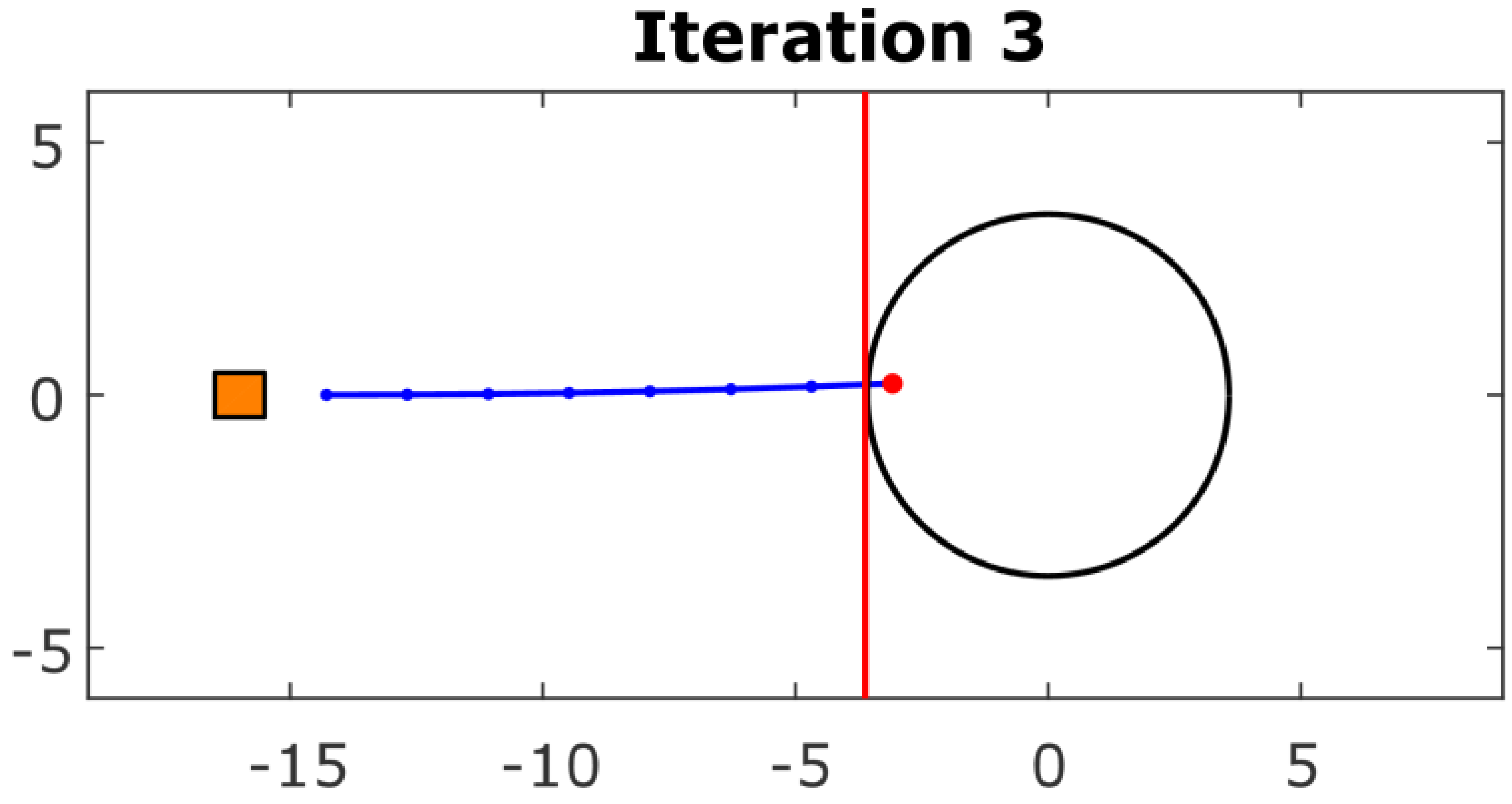
Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example



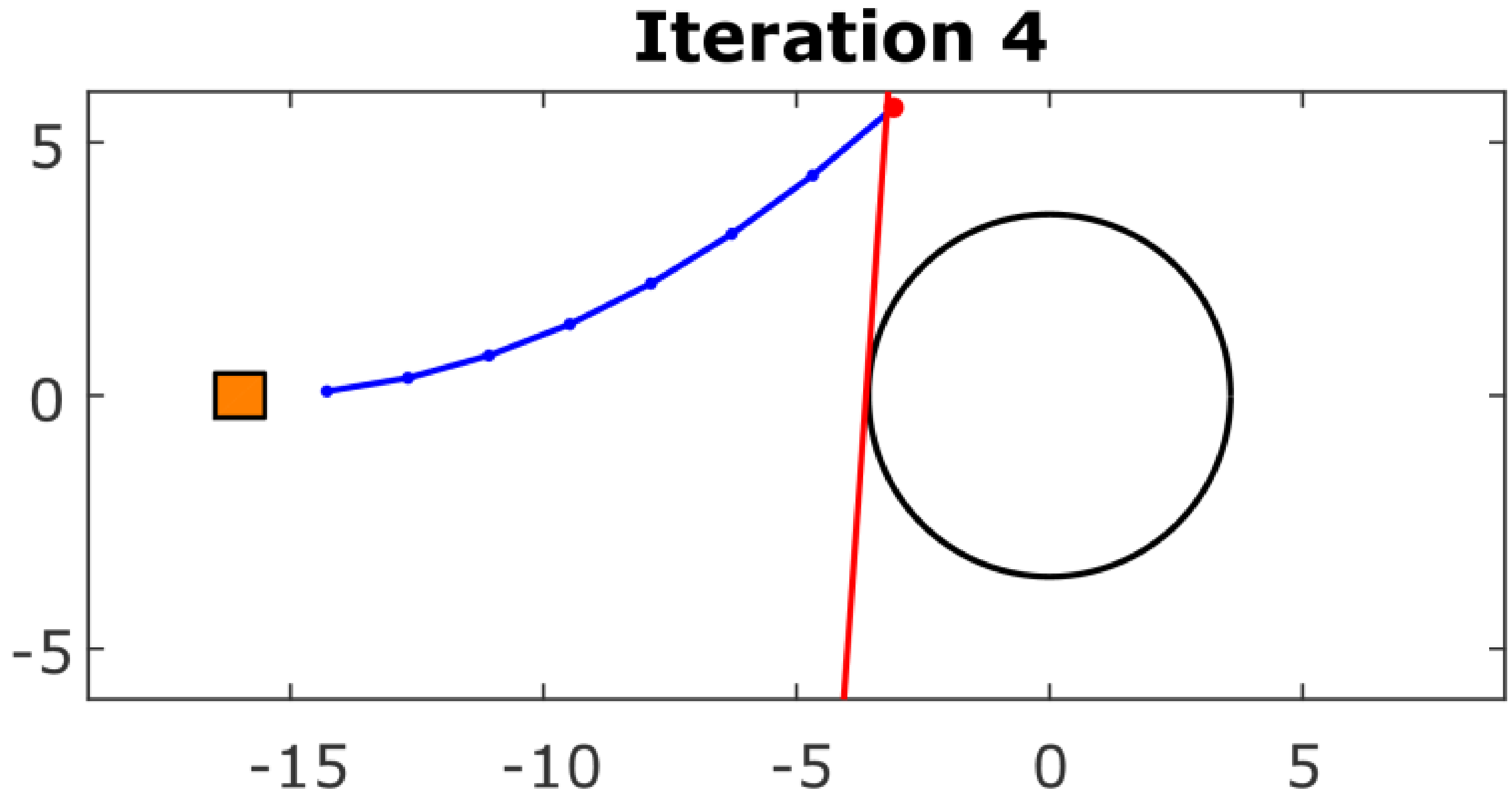
Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example



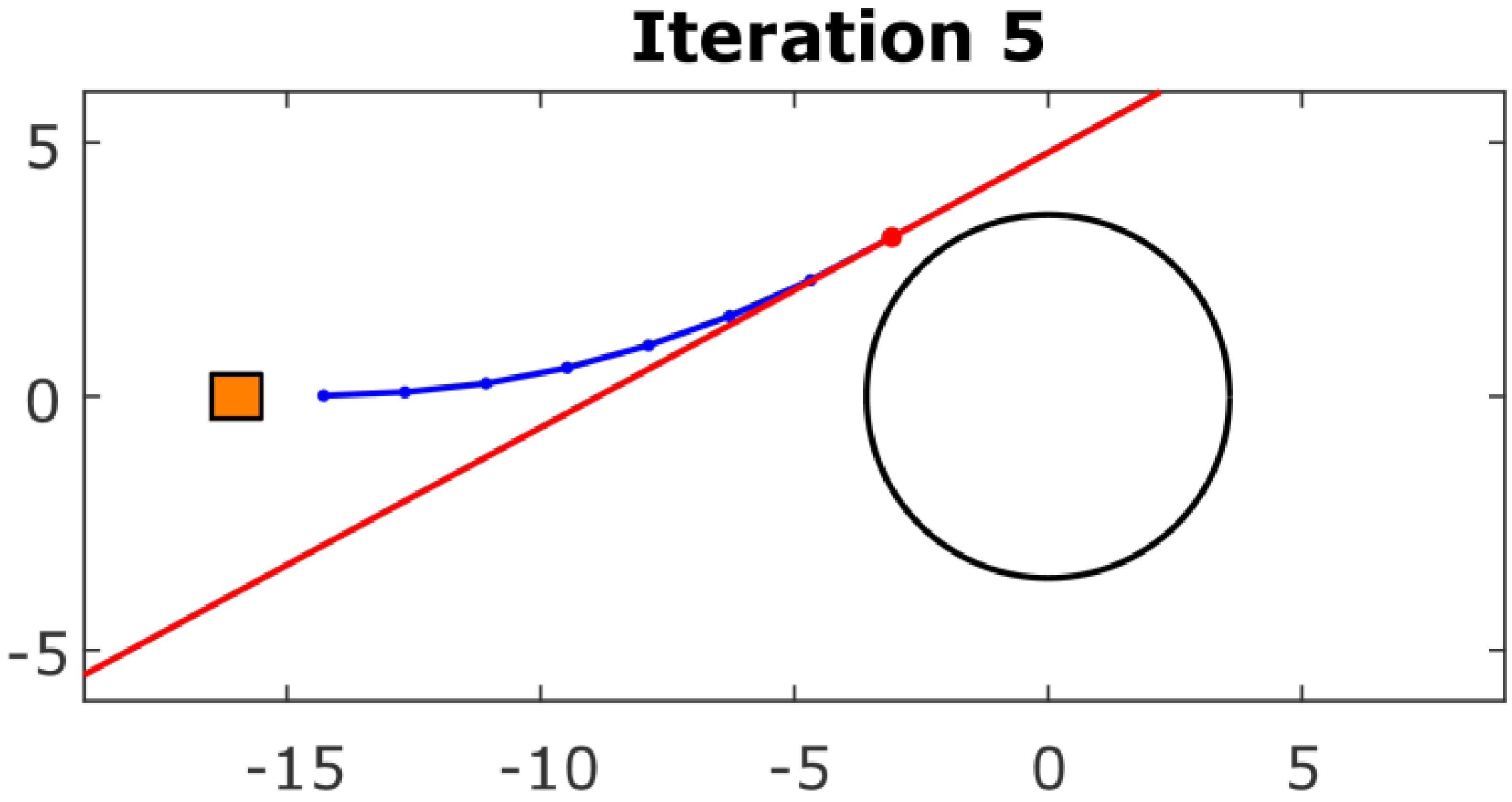
Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example



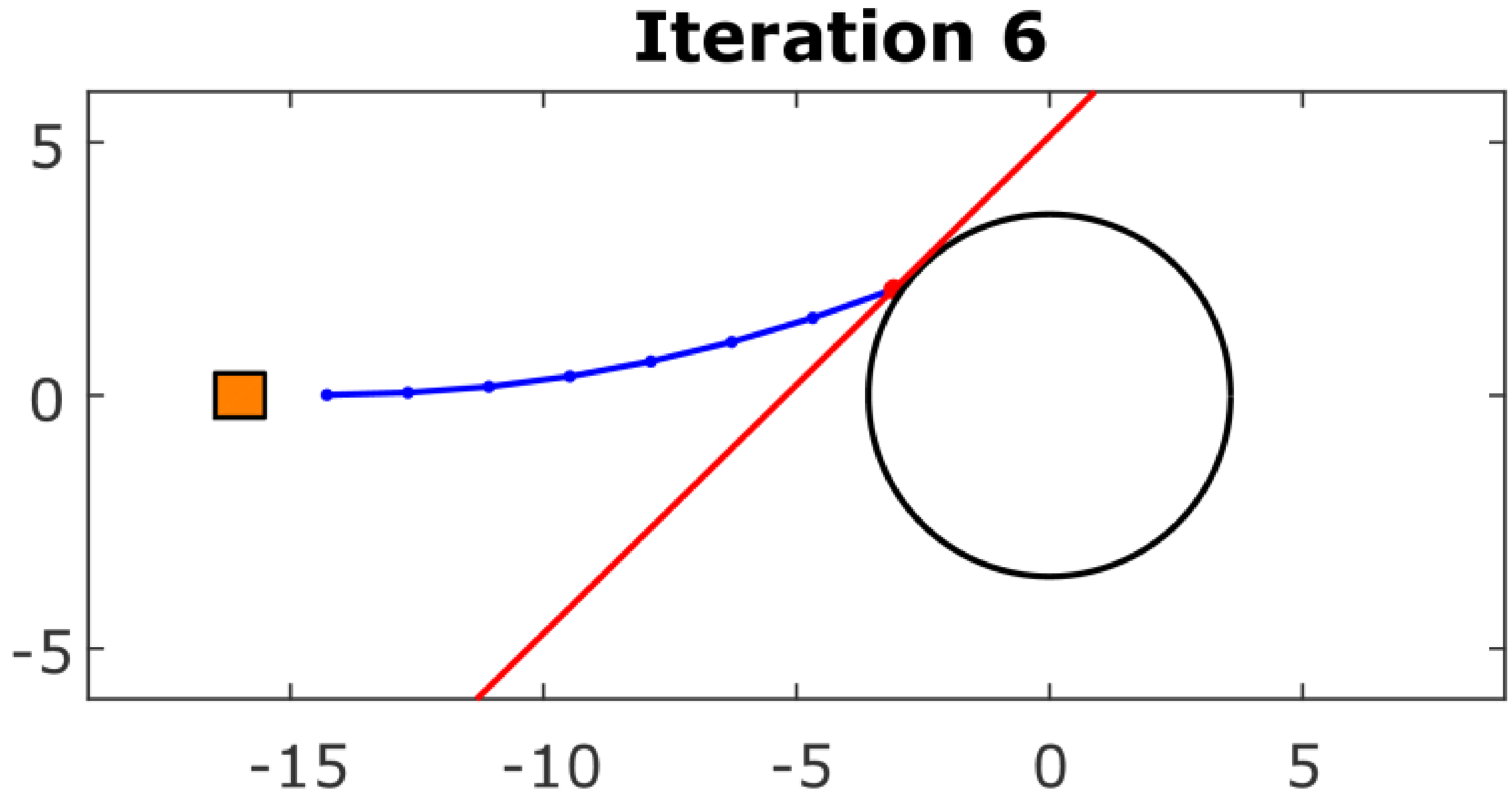
Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example



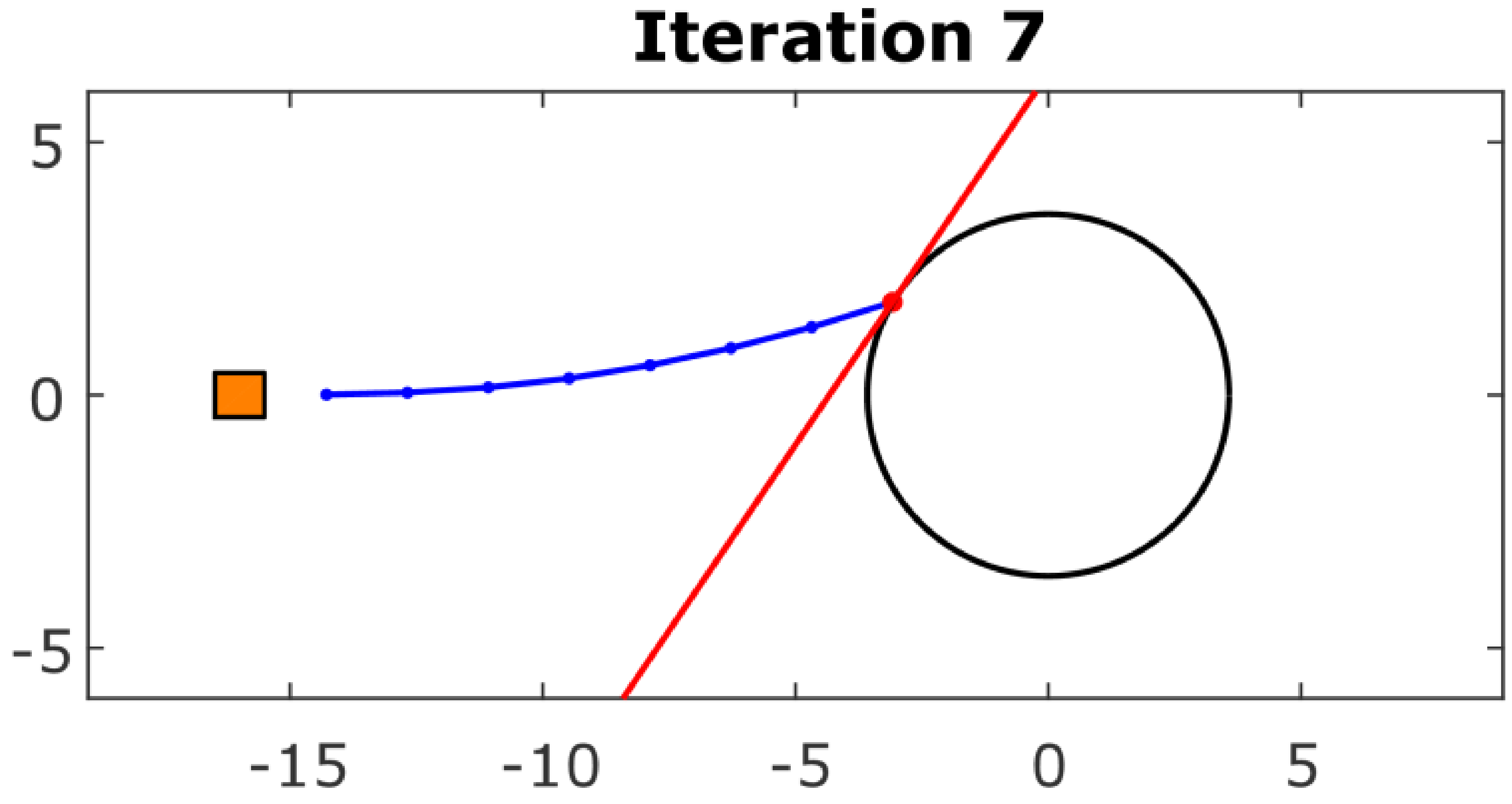
Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example



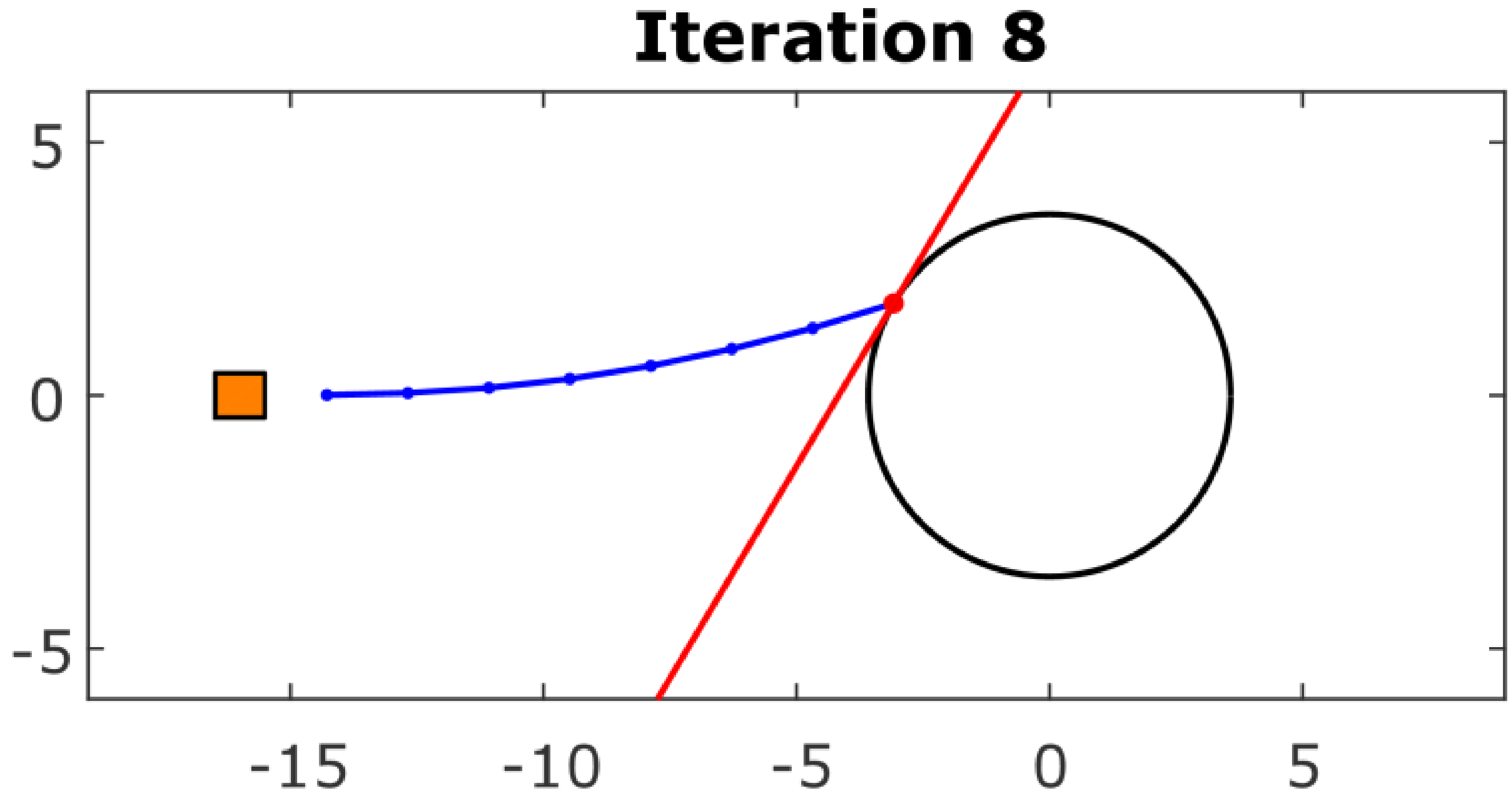
Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example



Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

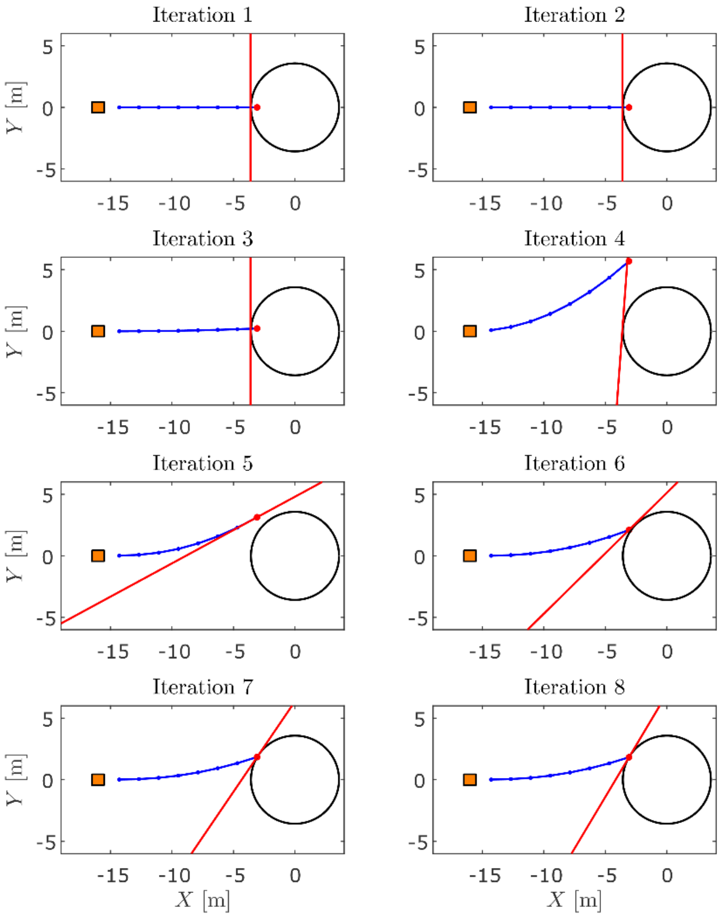
Sequential convex programming – vehicle example

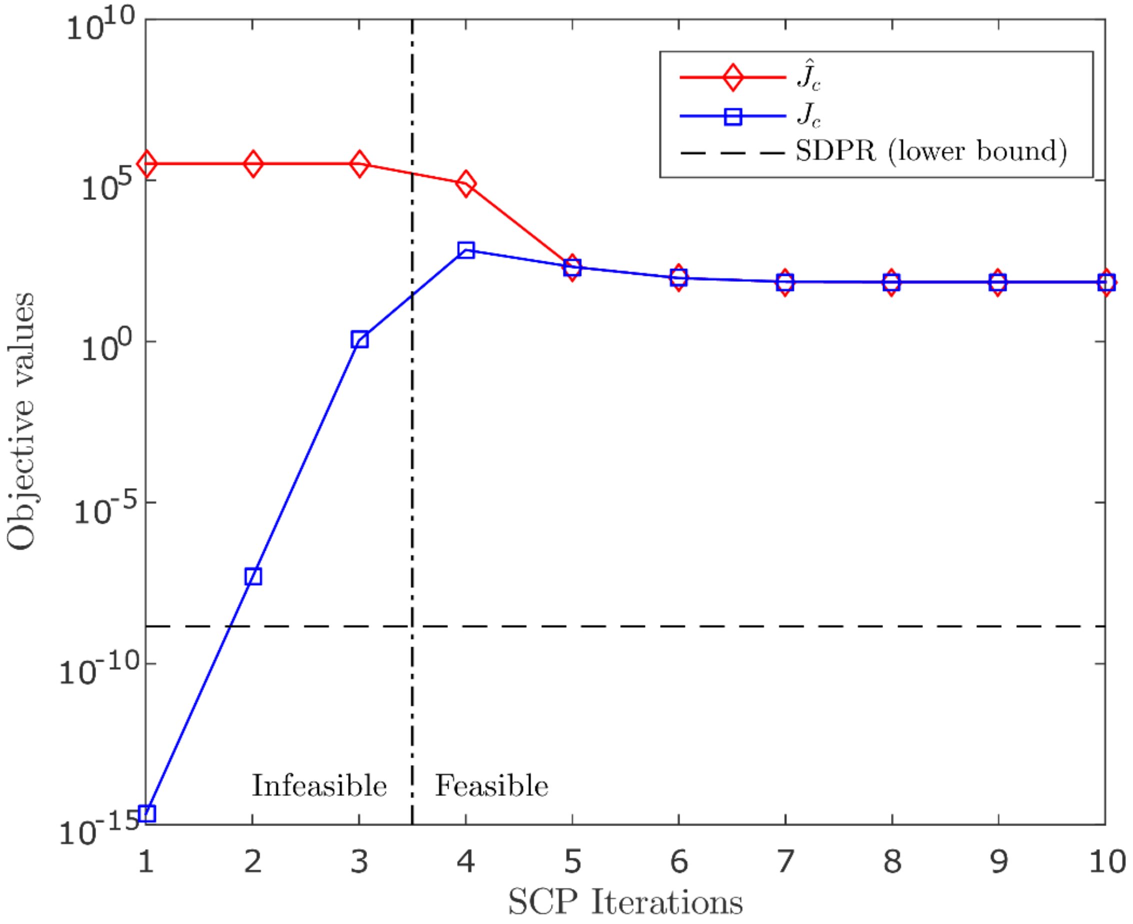


Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example

- Convergence of SCP: $\widehat{J_c}$, Progress of SCP: J_c

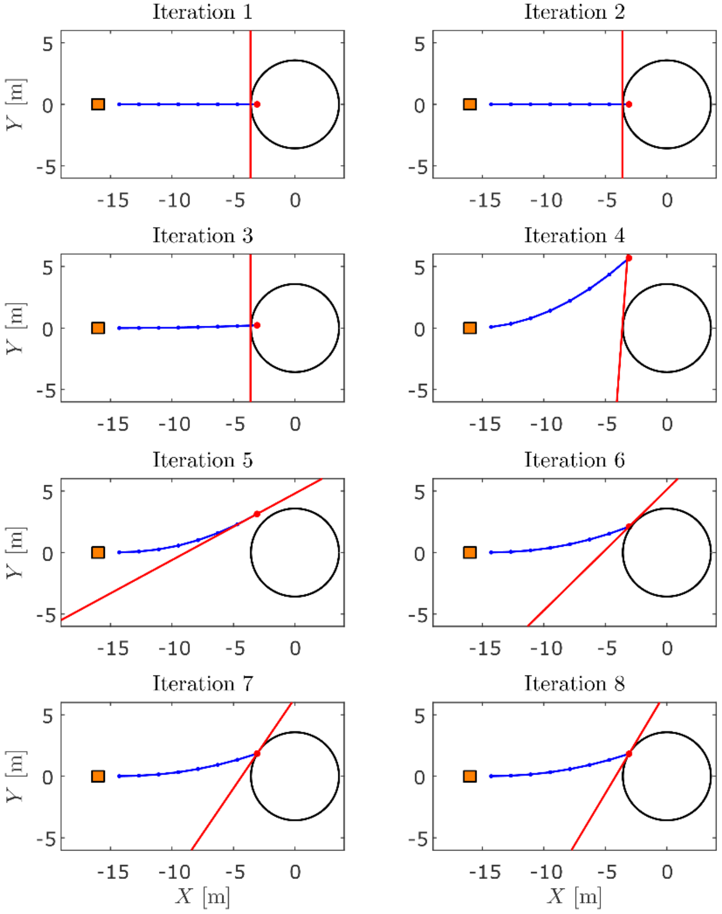


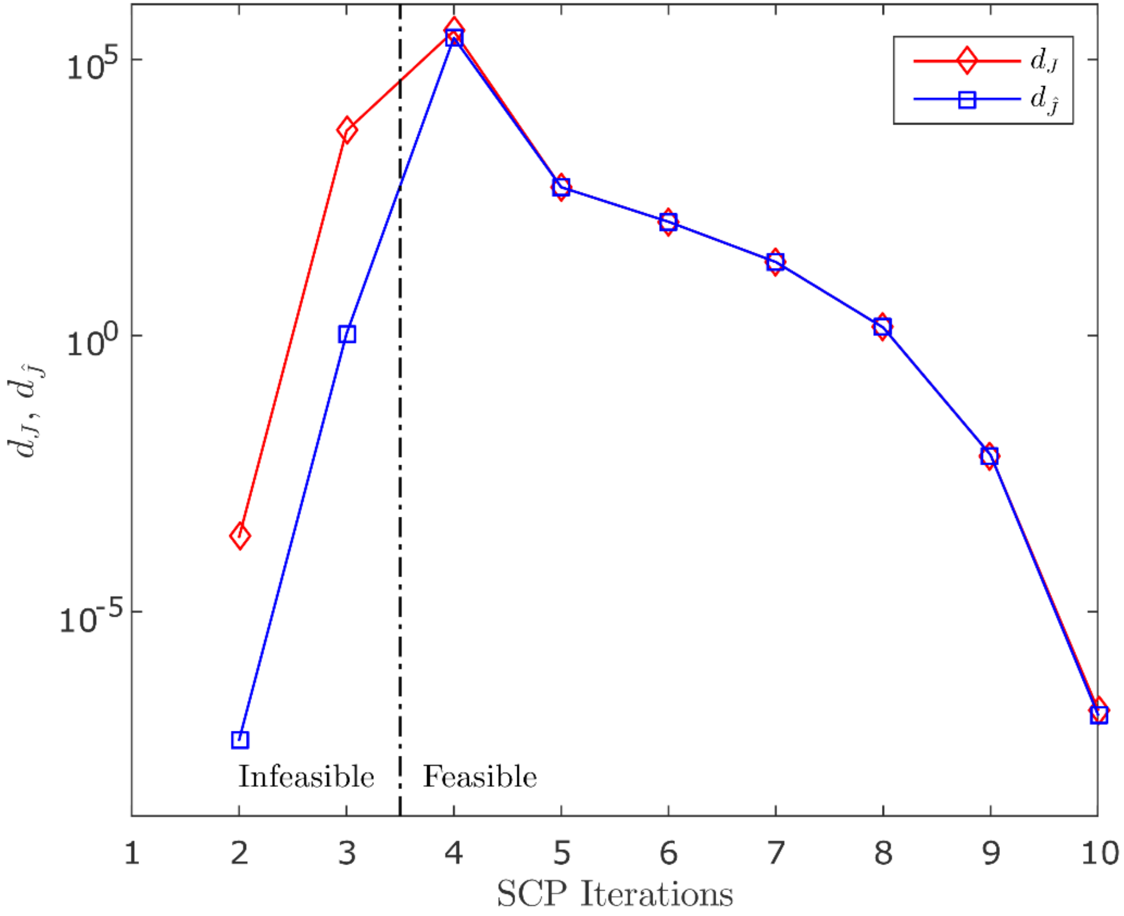


Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example

- Predicted and real decrease of the objective value
  - $d_{\hat{J}} \leq d_J$ because approximate program is a restriction

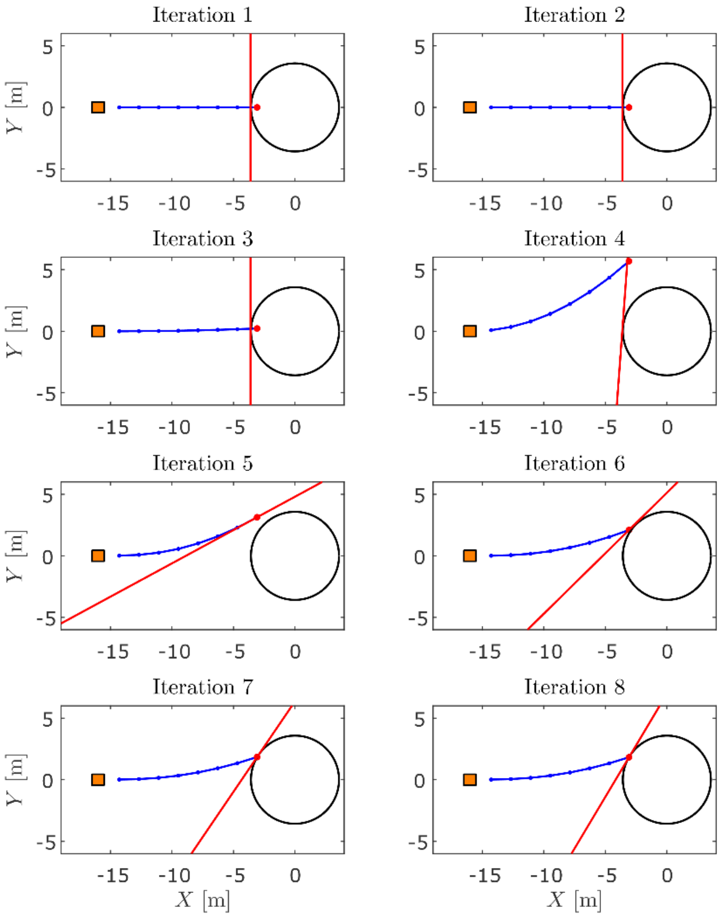


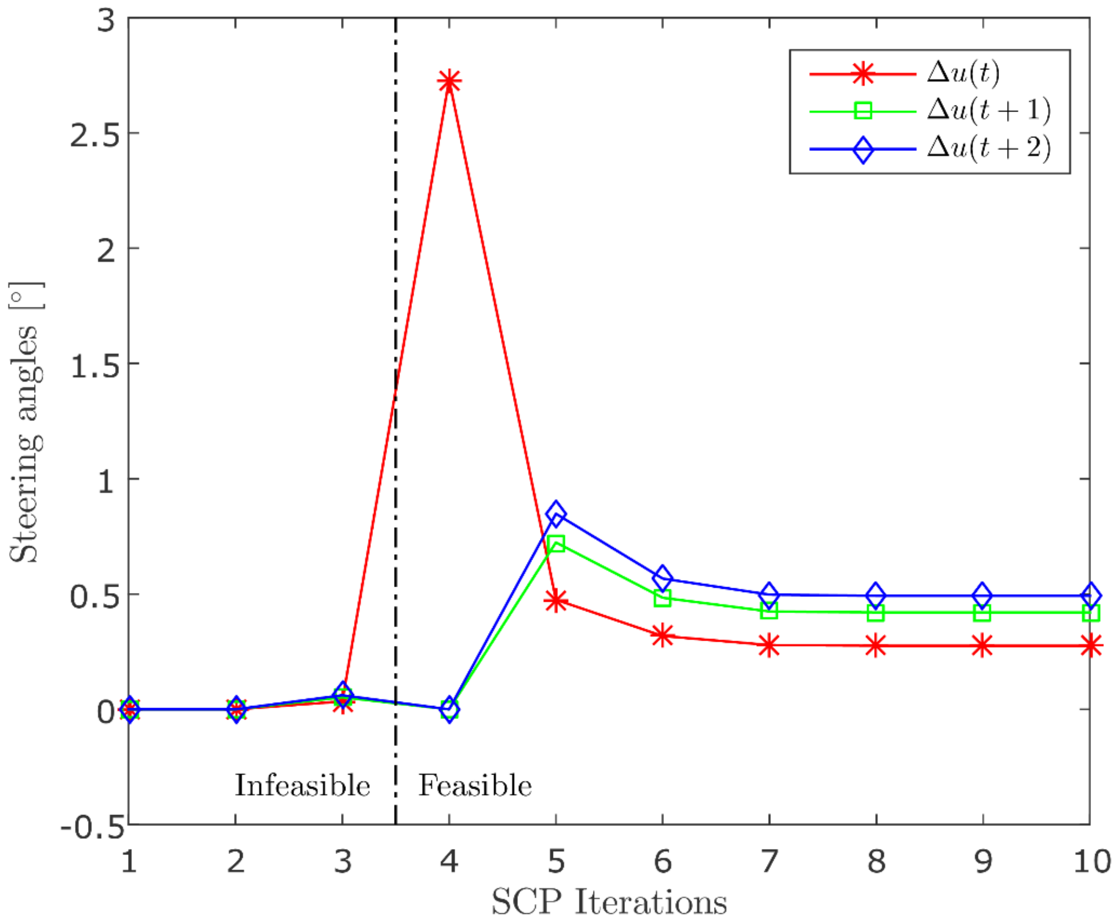


Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example

- Convergence of the control input (control horizon = 3)

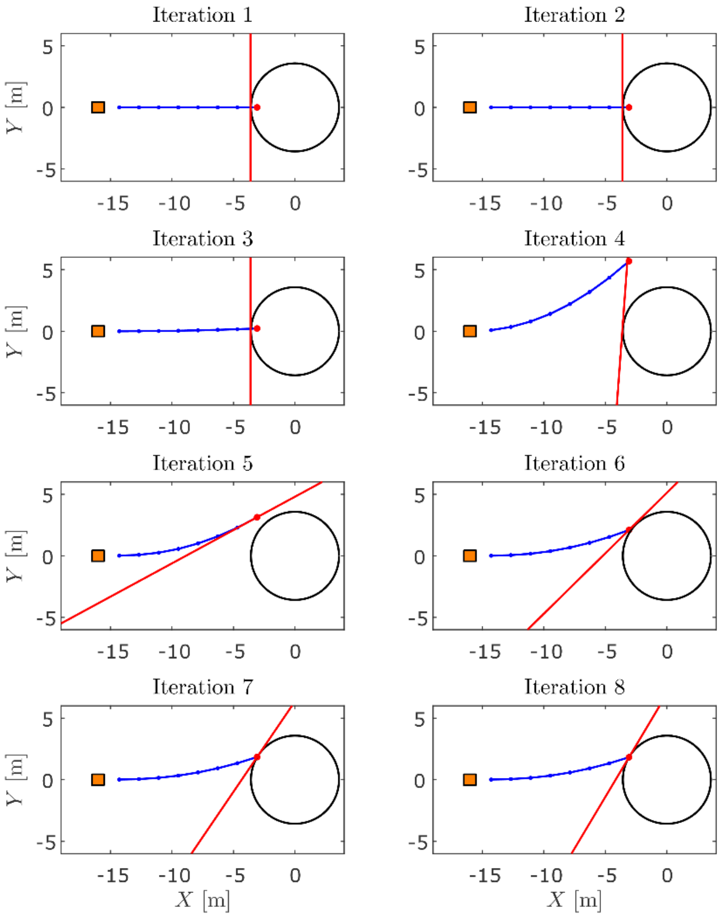


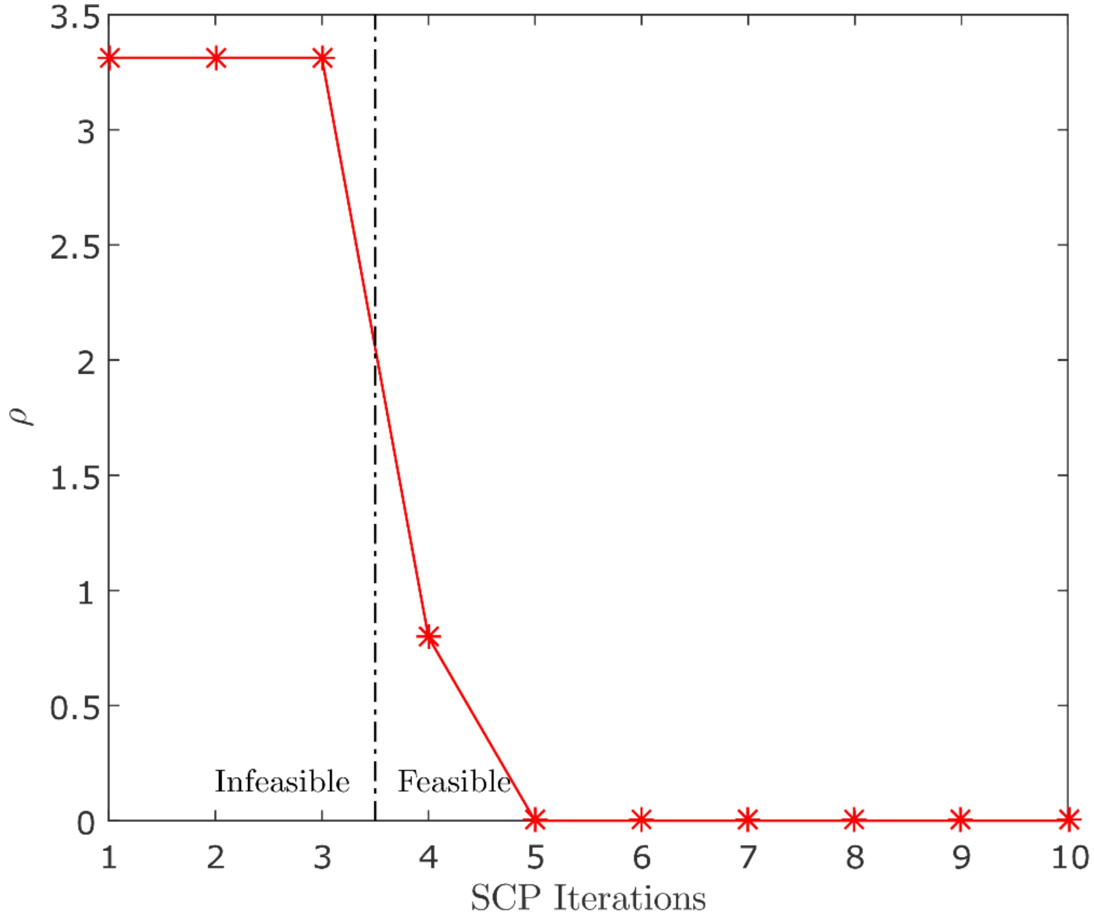


Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example

- Slack variable, progress of SCP

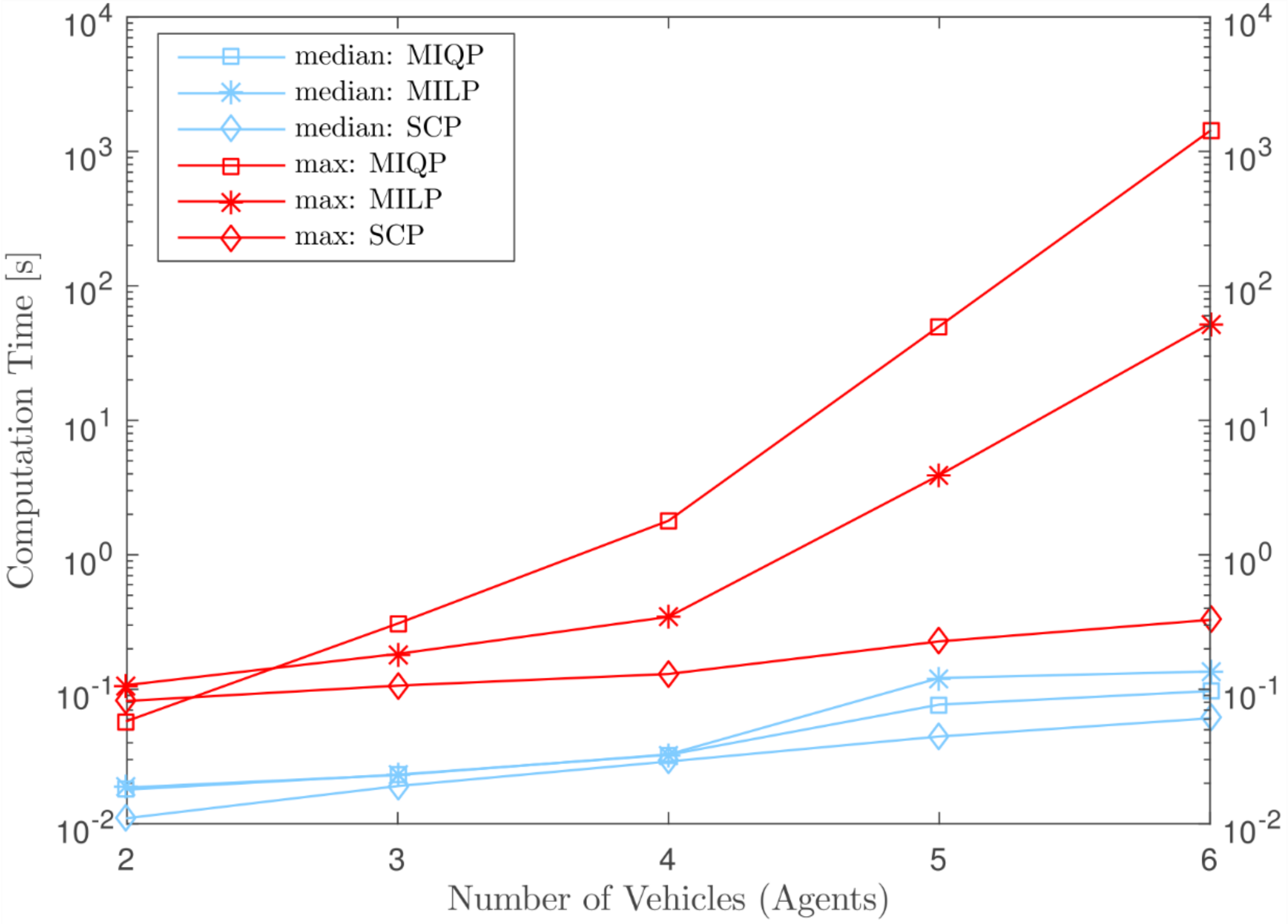




Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example

- Computation time



Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming – vehicle example

- Video of “Frogger” simulation
- Video of experimental results
- Video of trust
- Vehicle racing, videos of simulation results
  - https://youtu.be/t4tkZA8yZkg
  - https://youtu.be/_BxhYhIbORk

Sequential convex programming – conclusion

- SCP finds a (good) upper bound on the non-convex optimization problem starting from the time-shifted last solution of MPC

Alrifaee, Networked Model Predictive Control for Vehicle Collision Avoidance, 2017

Sequential convex programming

- MATLAB exercise
  - B. Alrifaee. MATLAB Simulation of Networked Model Predictive Control for Vehicle Collision Avoidance, 2017. Available: https://doi.org/10.5281/zenodo.1252992
    - Function \controller\SCP_optimizer.m