Model Predictive Control for Autonomous Vehicles

TSFS12: Autonomous Vehicles – Planning, Control, and Learning Systems

Lecture 7: Björn Olofsson <bjorn.olofsson@liu.se>

LINKÖPING UNIVERSITY

Purpose of this Lecture

- Give a **background** on **receding-horizon control** and **model predictive control** (MPC).
- Discuss different types of **MPC problems** for **autonomous vehicles** and how they can be **solved**.
- Illustrate use of **MPC** for **trajectory-tracking** and **path-following applications**.

Expected Take-Aways from this Lecture

- Basic **knowledge** about **model predictive control**.
- Be familiar with **applications of MPC for autonomous vehicles**.
  - **Trajectory tracking**.
  - **Path following**.
- Familiarity with **solution of MPC problems**.

Literature Reading

The following book and article sections are the main reading material for this lecture. References to further reading are provided throughout the slides and at the end of the lecture slides.

- Chapter 1 in Rawlings, J. B., D. Q. Mayne, & M. Diehl: *Model Predictive Control: Theory, Computation, and Design*. 2nd Edition. Nob Hill Publishing, 2017.
- Section V-C in Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E: ”A survey of motion planning and control techniques for self-driving urban vehicles”. *IEEE Transactions on Intelligent Vehicles*, 1(1), 33-55, 2016.

Outline of the Lecture

- **Application example**: Look-ahead control for autonomous trucks.
- **Fundamentals** of **model predictive control** (MPC).
- **Trajectory tracking** for autonomous vehicles using **MPC**.
- **Path following** for autonomous vehicles using **MPC**.
- **Tools** and **libraries** for **optimization** and **MPC**.

Context in the Architecture for an Autonomous Vehicle

Model predictive control is possible to apply at several different layers in the architecture for different tasks.

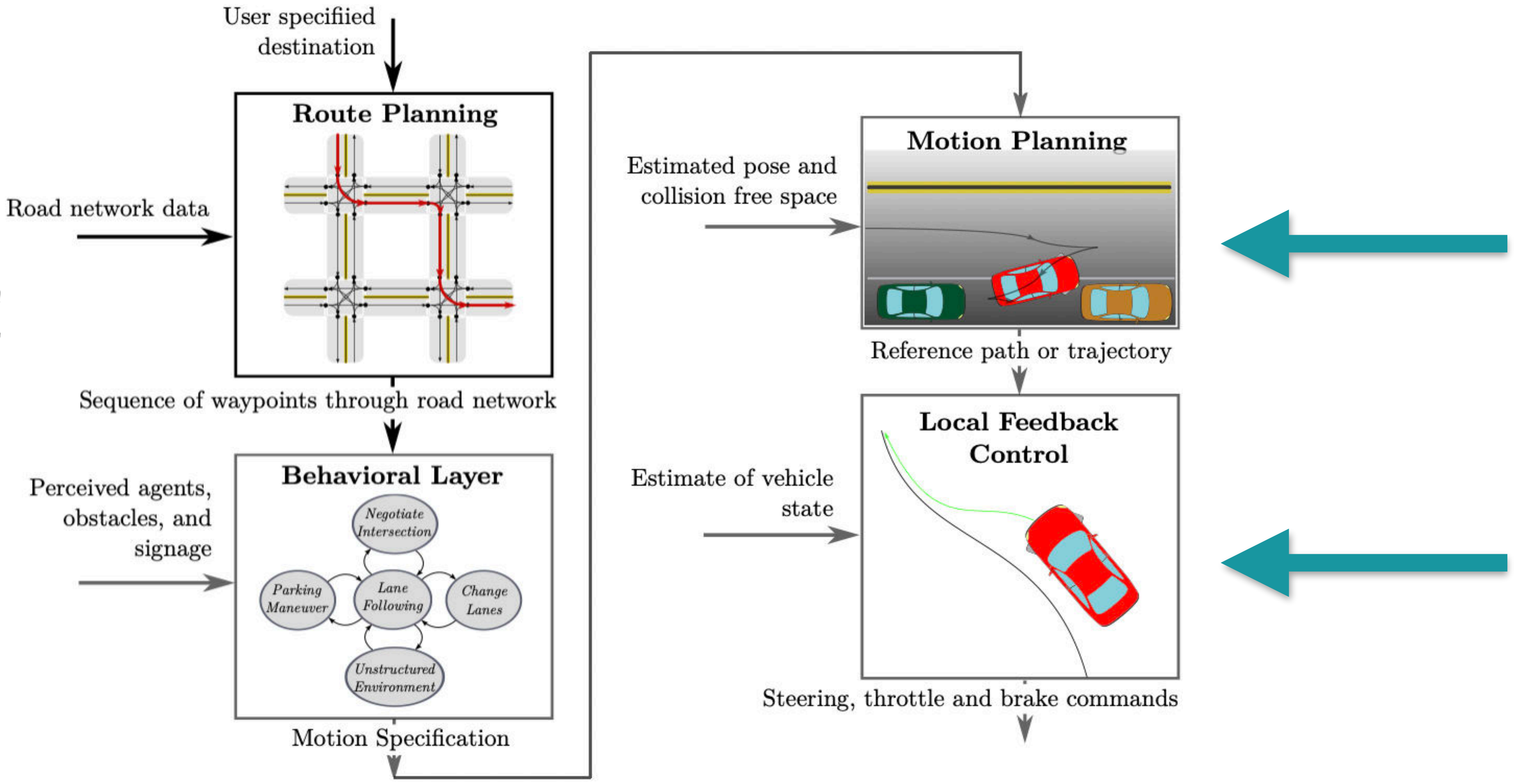
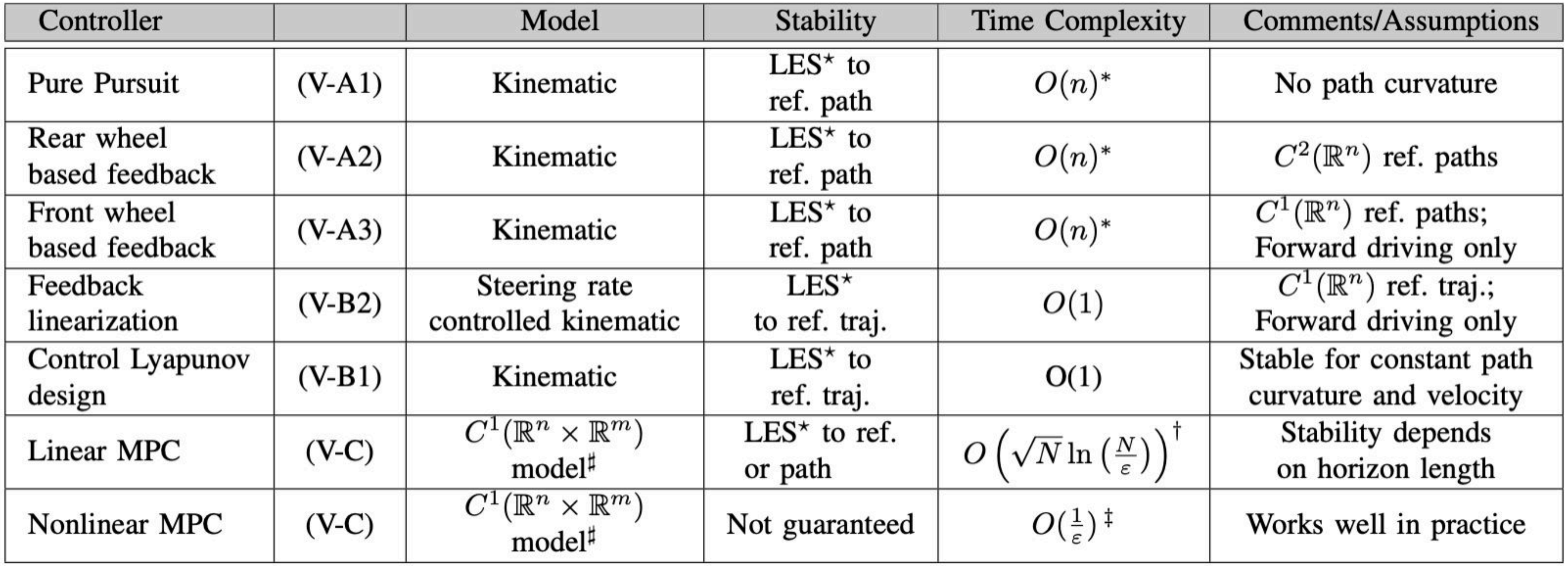

User specified destination
Route Planning
Road network data
Sequence of waypoints through road network
Behavioral Layer
Perceived agents, obstacles, and signage
Negotiate Intersection
Parking Maneuver
Lane Following
Change Lanes
Unstructured Environment
Motion Specification
Motion Planning
Estimated pose and collision free space
Reference path or trajectory
Local Feedback Control
Estimate of vehicle state
Steering, throttle and brake commands

Figure from: Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E: "A survey of motion planning and control techniques for self-driving urban vehicles". IEEE Transactions on Intelligent Vehicles, 1(1), 33-55, 2016.

Overview of Control Methods for Autonomous Vehicles



	Controller		Model	Stability	Time Complexity	Comments/Assumptions
Lecture 6	Pure Pursuit	(V-A1)	Kinematic	LES* to ref. path	$O(n)^*$	No path curvature
	Rear wheel based feedback	(V-A2)	Kinematic	LES* to ref. path	$O(n)^*$	$C^2(\mathbb{R}^n)$ ref. paths
	Front wheel based feedback	(V-A3)	Kinematic	LES* to ref. path	$O(n)^*$	$C^1(\mathbb{R}^n)$ ref. paths; Forward driving only
	Feedback linearization	(V-B2)	Steering rate controlled kinematic	LES* to ref. traj.	$O(1)$	$C^1(\mathbb{R}^n)$ ref. traj.; Forward driving only
Lecture 6	Control Lyapunov design	(V-B1)	Kinematic	LES* to ref. traj.	O(1)	Stable for constant path curvature and velocity
This lecture	Linear MPC	(V-C)	$C^1(\mathbb{R}^n \times \mathbb{R}^m)$ model♯	LES* to ref. or path	$O\left(\sqrt{N}\ln\left(\frac{N}{\varepsilon}\right)\right)^\dagger$	Stability depends on horizon length
This lecture	Nonlinear MPC	(V-C)	$C^1(\mathbb{R}^n \times \mathbb{R}^m)$ model♯	Not guaranteed	$O(\frac{1}{\varepsilon})^\ddagger$	Works well in practice

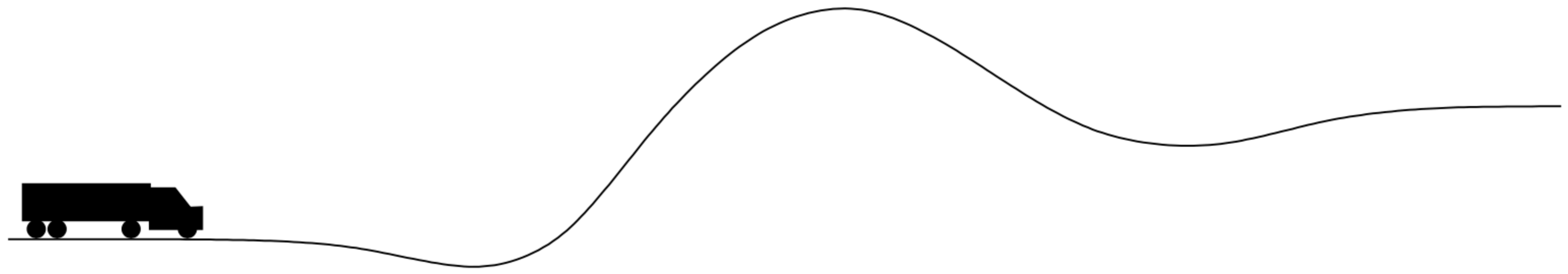
Table from: Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E: "A survey of motion planning and control techniques for self-driving urban vehicles". IEEE Transactions on intelligent vehicles, 1(1), 33-55, 2016.

LINKÖPING UNIVERSITY

Application Example:
Look-Ahead Control for Autonomous Trucks

Scenario for Look-Ahead Control (1/2)

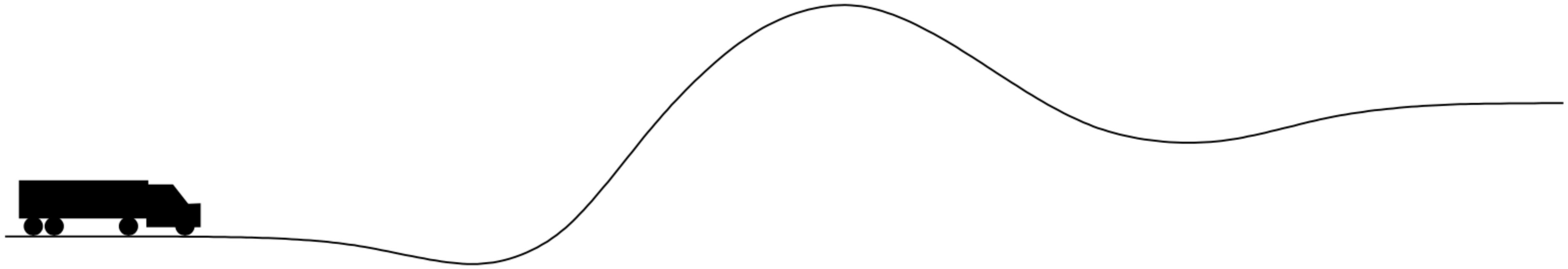
- An autonomous **long-haulage heavy truck** on open road.
- An **on-board road topography map** and **on-board GPS**.
- **Control inputs**: engine torque, brake torque, and gear.



Hellström, E., Ivarsson, M., Åslund, J., & Nielsen, L: "Look-ahead control for heavy trucks to minimize trip time and fuel consumption". Control Engineering Practice, 17(2), 245-254, 2009.

Scenario for Look-Ahead Control (2/2)

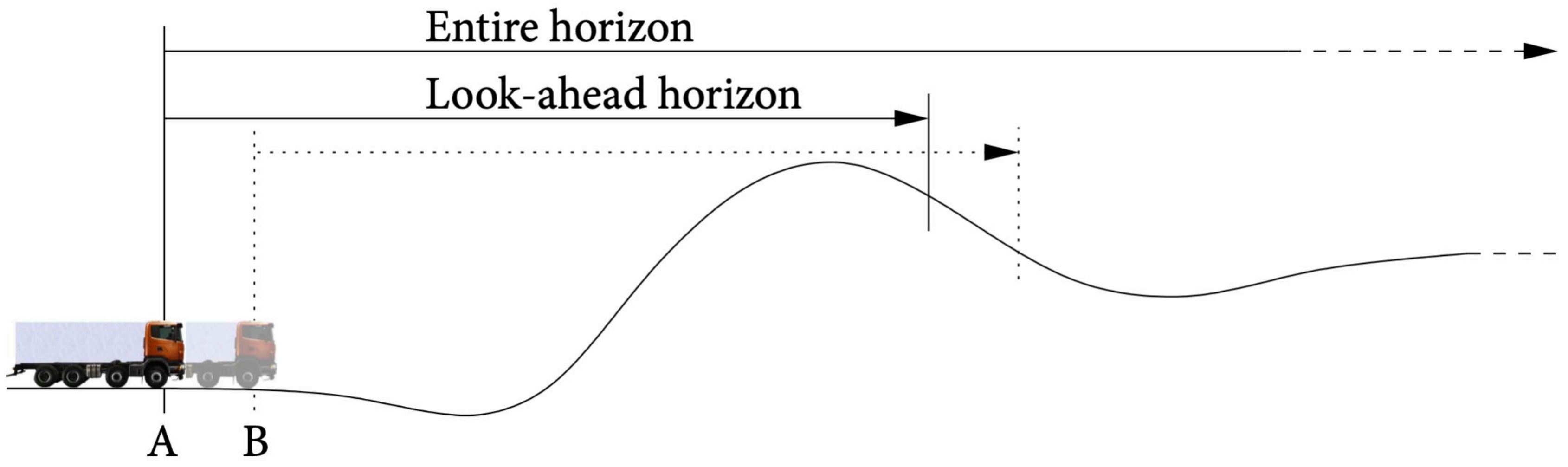
- Objective: **Minimize consumption of fuel**, with a maximum trip time.
  - An **autonomous cruise controller** with **optimal control actions**, considering road topography.
- Can be formalized as a **mathematical optimization problem** (recall Lecture 5).



Hellström, E., Ivarsson, M., Åslund, J., & Nielsen, L: "Look-ahead control for heavy trucks to minimize trip time and fuel consumption". Control Engineering Practice, 17(2), 245-254, 2009.

Receding Horizon

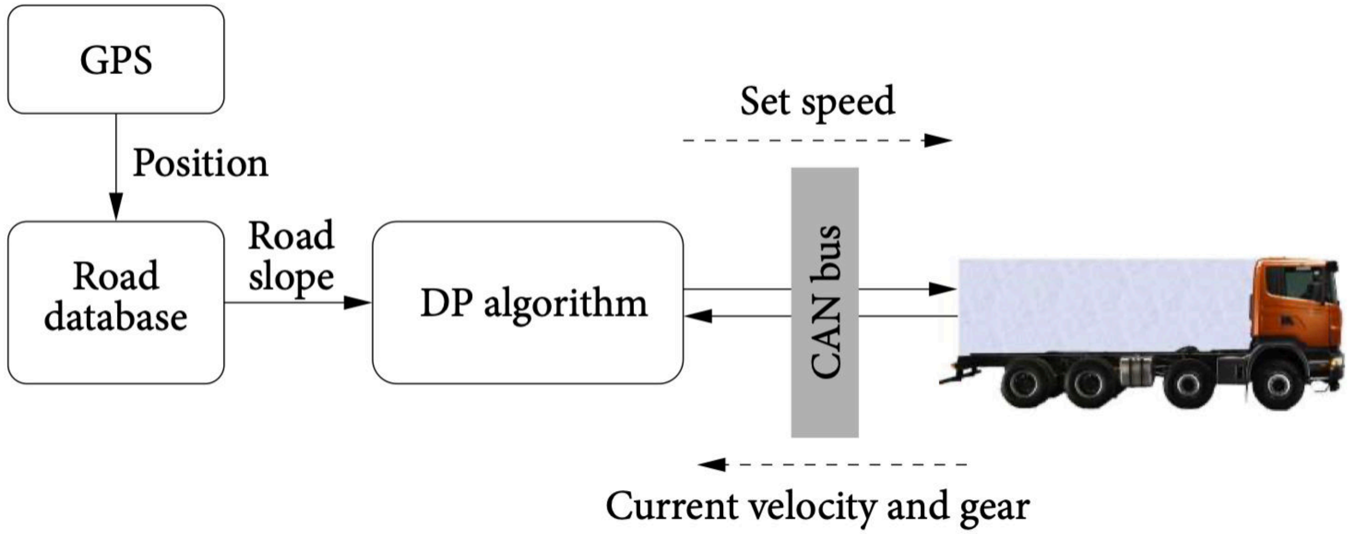
- Objective is to **minimize a performance criterion** over the complete driving mission – however, **high computational cost** and **uncertainty**.
- Consider a **finite horizon** (look-ahead horizon) at point A, **solve problem** and **apply control inputs** in first part of the interval until point B, then **solve problem again** at point B over a shifted finite horizon – i.e., a **receding horizon**.



Hellström, E., Ivarsson, M., Åslund, J., & Nielsen, L: "Look-ahead control for heavy trucks to minimize trip time and fuel consumption". Control Engineering Practice, 17(2), 245-254, 2009.

Application of Look-Ahead Control

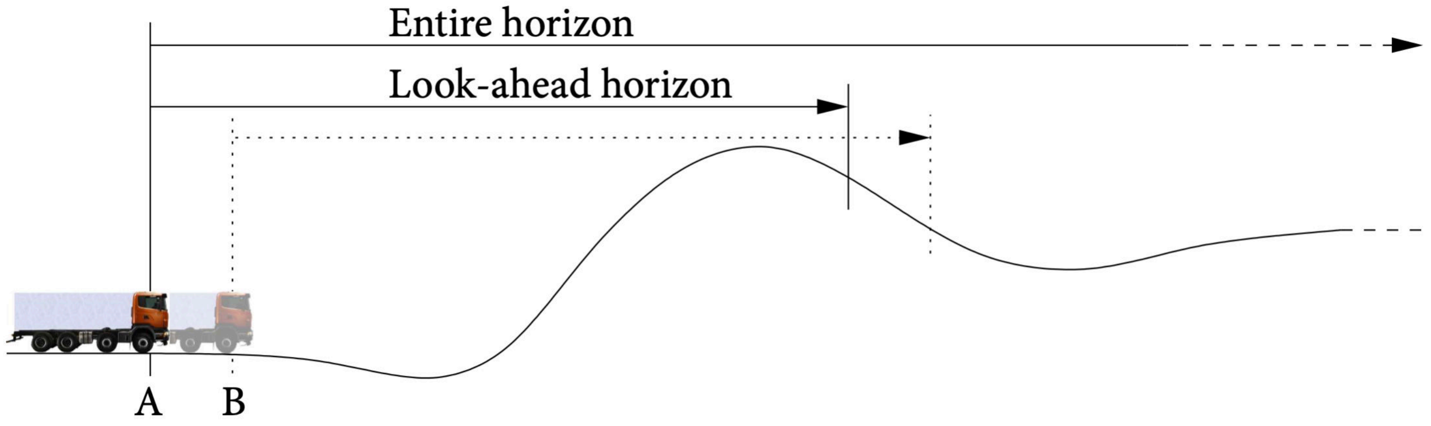
- A **GPS** gives the position of the autonomous truck.
- A **road data base** gives the road slope ahead of the vehicle from the current position over a **limited distance ahead of the vehicle** (typically about 2 km).
- A dynamic programming (DP) algorithm, together with a **model of the vehicle**, is used to **determine the control inputs** that minimize the fuel consumption over this intervall.



Hellström, E., Ivarsson, M., Åslund, J., & Nielsen, L: "Look-ahead control for heavy trucks to minimize trip time and fuel consumption". Control Engineering Practice, 17(2), 245-254, 2009.
Bertsekas, D. P: Reinforcement learning and optimal control. Belmont, MA: Athena Scientific, 2019.

Key Characteristics of Look-Ahead Control

- The look-ahead approach can take **limitations on control signals** and **internal vehicle states** into account.
- In this driving segment, the truck will **accelerate before an uphill** if it is necessary to avoid that the **speed decreases below a lower limit** on speed because of limits on the propelling force.

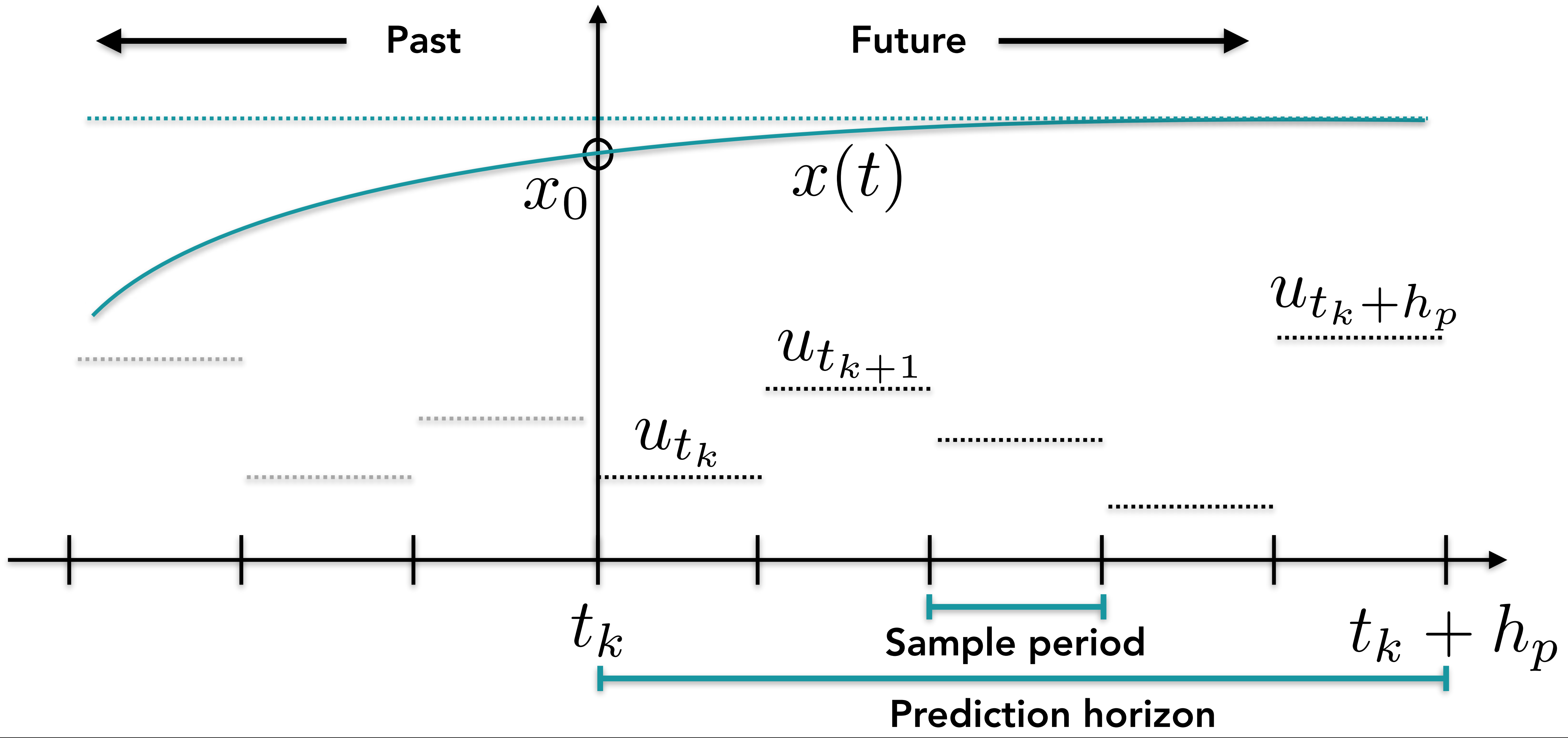


- Can handle **changing driving conditions**, thanks to re-optimization.

Hellström, E., Ivarsson, M., Åslund, J., & Nielsen, L: "Look-ahead control for heavy trucks to minimize trip time and fuel consumption". Control Engineering Practice, 17(2), 245-254, 2009.

Fundamentals of Model Predictive Control

Model Predictive Control (MPC)



Key Concepts in MPC (1/2)

- A **model** of the system is used for **prediction**.
- The length of the finite horizon is called the **prediction horizon**.
- The **control horizon** is the **subset of the prediction horizon** over which the control inputs are allowed to vary.
  - Could also be the **full set**, i.e., the horizons coincide.

Key Concepts in MPC (2/2)

- **Cost function** describing desired **objective** and **constraints** on control inputs and states.
- User-specified **weights** often used to trade **different components** to each other in the **cost function** of the MPC formulation.

Recall Optimization Problem from Lecture 5

Lagrange integrand

Mayer term

- **Optimization problem** over time horizon $[0, T]$, where T possibly is a free optimization variable:

$$
\begin{aligned}
\text{minimize} \quad & \int_0^T L(x(t), u(t))\,\mathrm{d}t + \Gamma(x(T)) \\
\text{subject to} \quad & x(0) = x_0,\ \ \dot{x}(t) = f(t, x(t), u(t)), \\
& x(t) \in \mathbb{X},\ \ u(t) \in \mathbb{U},\ \ x(T) \in \mathbb{X}_T,\ \ t \in [0, T]
\end{aligned}
$$

Initial conditions

System dynamics

State and control constraints

Terminal constraints

A General MPC Formulation

- Solve **optimization problem** over the **time horizon** $[t_k, t_k + h_p]$ at selected time instants $t_k,\ k = 0, 1, 2, \dots$, with h_p the prediction horizon:

$$\begin{aligned} \text{minimize} \quad & \int_{t_k}^{t_k+h_p} L(x(t), u(t))\,\mathrm{d}t + \Gamma(x(t_k + h_p)) \\ \text{subject to} \quad & x(t_k) = x_0,\ \dot{x}(t) = f(t, x(t), u(t)), \\ & x(t) \in \mathbb{X},\ u(t) \in \mathbb{U},\ x(t_k + h_p) \in \mathbb{X}_{t_k+h_p},\ t \in [t_k, t_k + h_p] \end{aligned}$$

- The **computed control inputs** are applied until next time step t_{k+1}.
- The **current states** x_0 are updated at each new iteration.

A QP MPC Formulation (1/2)

- Recall the **LQR control design** from Lecture 6. Assume a **linear model**:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

- Choose the **cost function** as the following:

$$\begin{aligned} L(x(t), u(t)) &= x^{\mathrm{T}}(t)Qx(t) + u(t)^{\mathrm{T}}Ru(t) \\ \Gamma(x(t_k + h_p)) &= x(t_k + h_p)^{\mathrm{T}} P_f x(t_k + h_p) \end{aligned}$$

A QP MPC Formulation (2/2)

- The possibly time-varying **weight matrices** Q, R, P_f are positive (semi-)definite and are specified by the user.
  - Trade **different objectives** to each other.
- With all remaining constraints linear or affine, the problem is a **quadratic program (QP)**, i.e., a convex optimization problem.

Model Considerations in MPC

- The model equations of the system appear as a **constraint**, therefore imply properties of the resulting **optimization problem**.
- Recall the discussion on **convex vs. non-convex optimization** from Lecture 5.
- **Non-linear models** possible to include in the **MPC formulation**.
- **Linearizing a non-linear model** along a **nominal/reference trajectory** leads to a linear time-varying (LTV) system:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

Hard vs. Soft Constraints

- Explicit constraint that needs to be fulfilled exactly – **hard constraint**.
- **Penalty in cost function** with certain weight – **soft constraint** (fulfil it as good as possible, given other objectives).
  - Could be beneficial in real-time applications.
- **Slack variables for constraints** can be introduced in the MPC formulation to avoid infeasible problems because of **constraints**.

Hard vs. Soft Constraints – Example

- Assume that the speed v of an autonomous car should be kept close to a **desired reference velocity** v_{ref}.
- A **soft constraint** can be formulated as a component in the **cost function** as:

$$\int_{t_k}^{t_k+h_p} (v - v_{\mathrm{ref}})^2 \,\mathrm{d}t$$

- A **hard constraint** can be formulated as an **explicit constraint** as:

$$v_{\mathrm{ref}} - \varepsilon \leq v \leq v_{\mathrm{ref}} + \varepsilon$$

Discretization of MPC Problem

- **Continuous-time MPC problems** need to be transformed to a **finite-dimensional optimization problem** (a non-linear program, **NLP**) for solution in a computer.
- Recall the methods from Lecture 5:
    - Direct **multiple shooting** or **collocation**.
    - **Discretization** of **system dynamics** and **cost function**, with polynomial approximation of control inputs and possibly states.

Computational Considerations

- A **longer prediction horizon** h_p implies a larger optimization problem, and thus higher computational cost.
- Also the **length of the sampling period** $\Delta t = t_{k+1} - t_k$, i.e., how often the MPC problem is solved, affects the computational cost.
- A **control horizon shorter than the prediction horizon** can be used to save computational resources.

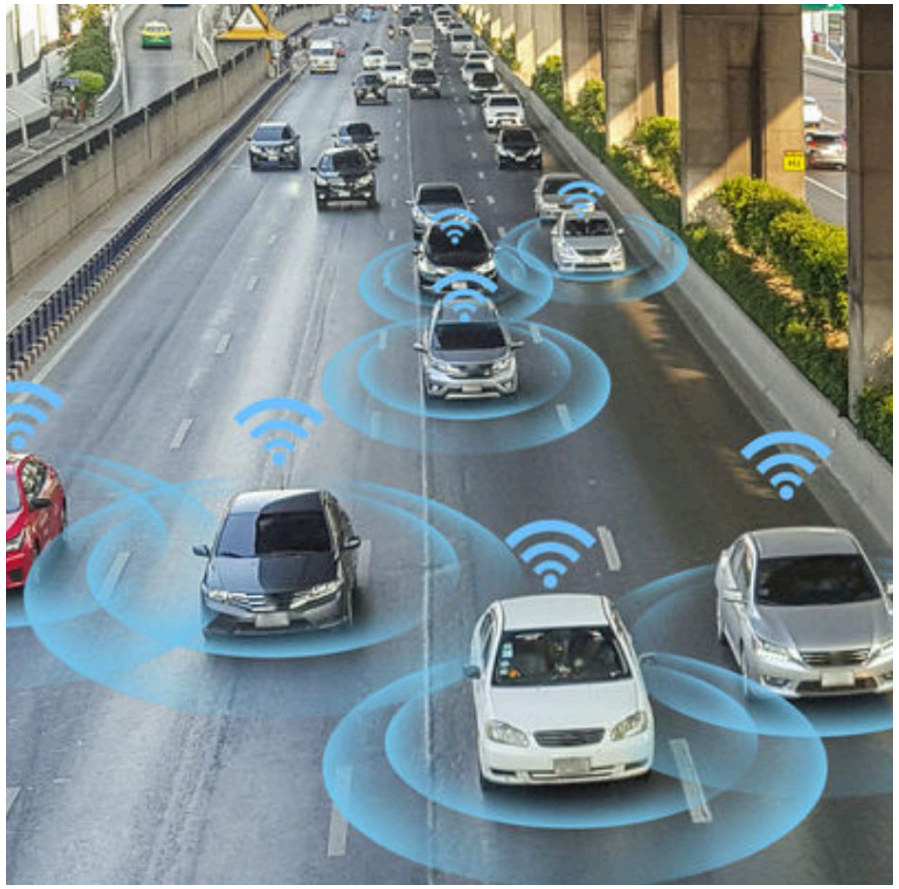
Stability of MPC

- **Length of prediction horizon** important in practice.
    - Sufficiently long horizon avoids short-term optimization.
- Choice of **cost** or **constraint** at **end of prediction horizon**.
    - Compare with having a sufficiently good **cost-to-go estimate**.
- **Extensive theory** exists for different formulations, see references for further details.

Rawlings, J. B., D. Q. Mayne, & M. Diehl: Model Predictive Control: Theory, Computation, and Design. Nob Hill Publishing, 2017.
Di Cairano, S., Kalabić, U. V., & Berntorp, K: "Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds". In IEEE Conference on Decision and Control (CDC), 709-714, 2016.

Why MPC for Autonomous Vehicles?

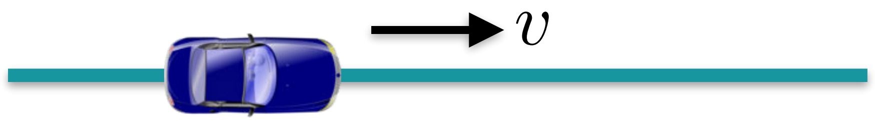
- Constantly **changing environment** with **inherent uncertainty** typical for autonomous vehicles.
- Different **sensor systems** provide updated information periodically (**situation awareness**).
- MPC offers **re-planning capabilities** combined with **optimal feedback control**.



MPC for Autonomous Trajectory Tracking

Trajectory Tracking vs. Path Following

- Recall the difference between a **path** (curve in space) and a **trajectory** (time-parameterized, velocity profile included).
- **MPC** can be formulated both for **trajectory tracking** and **path-following** applications.
- Consider the **example** of **driving along a straight line**.
    - Driving exactly at the straight line with a lower speed than according to the reference trajectory, gives a large trajectory error but no path error.



Problem Formulation for Trajectory Tracking

- Assume that we have **reference values** x_{ref} for the **states** that should be tracked.
- An **MPC cost function** for trajectory tracking could then be:

$$\int_{t_k}^{t_k+h_p} \left\{ (x(t) - x_{\mathrm{ref}}(t))^{\mathrm{T}} Q (x(t) - x_{\mathrm{ref}}(t)) + u(t)^{\mathrm{T}} R u(t) \right\} \mathrm{d}t$$

Example Constraints in Trajectory Following

- A key feature of MPC is that **constraints** can be included in the formulation and thus considered in the controller.
- **Linear constraints** on **inputs** and **states** formulated as:

$$u_{\min} \leq u(t) \leq u_{\max}$$

$$x_{\min} \leq x(t) \leq x_{\max}$$

- Input can be **acceleration** and states could be **position** and **velocity** for an autonomous vehicle, thus specifying, e.g., **maximum forward acceleration** or **not allowed areas in geometric space**.

MPC for Kinematic Path Following

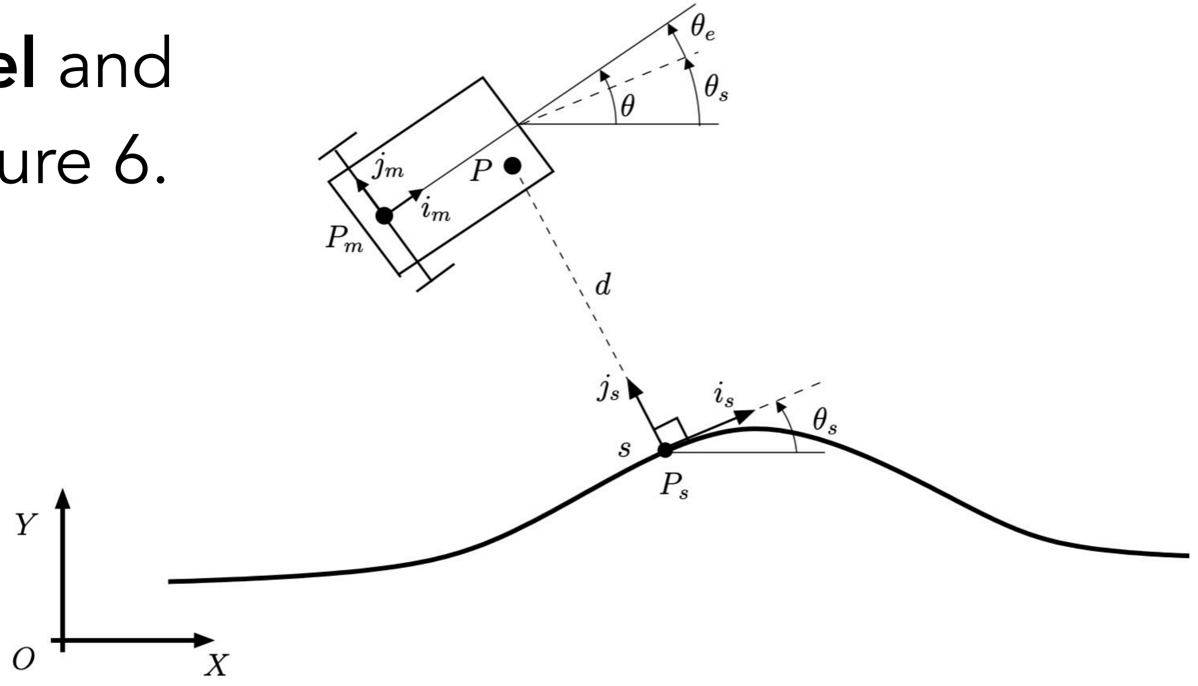
Path Following for Kinematic Model

- Recall the **kinematic motion model** and **path-following** problem from Lecture 6.

$$\dot{s} = \frac{v}{1 - dc(s)} \cos(\theta_e),$$

$$\dot{d} = v \sin(\theta_e),$$

$$\dot{\theta}_e = vu - \dot{s}c(s)$$



- States $x = \begin{pmatrix} d & \theta_e \end{pmatrix}$ and system dynamics $f(x, u) = \begin{pmatrix} v \sin(\theta_e) \\ vu - \dot{s}c(s) \end{pmatrix}$

MPC Formulation for Kinematic Path Following (1/2)

- An **MPC problem** for **path following**, where the objective is to achieve that $x \to 0$ (i.e., no path deviations).
- With the **weights** $Q = \mathrm{diag}(q_1, q_2),\ R = r$, the MPC problem for **path following** can be formulated as:

$$\begin{aligned} \text{minimize} \quad & \int_{t_k}^{t_k+h_p} q_1 d^2 + q_2 \theta_e^2 + r u^2 \, \mathrm{d}t \\ \text{subject to} \quad & x(t_k) = x_0,\ \dot{x}(t) = f(x(t), u(t)), \\ & u(t) \in \mathbb{U},\ t \in [t_k, t_k + h_p] \end{aligned}$$

MPC Formulation for Kinematic Path Following (2/2)

- The **constraint** on the control input can be determined based on **steering-angle limitations** as:

$$\mathbb{U} = \{u \mid 1/L \tan(\delta_{\min}) \leq u \leq 1/L \tan(\delta_{\max})\}$$

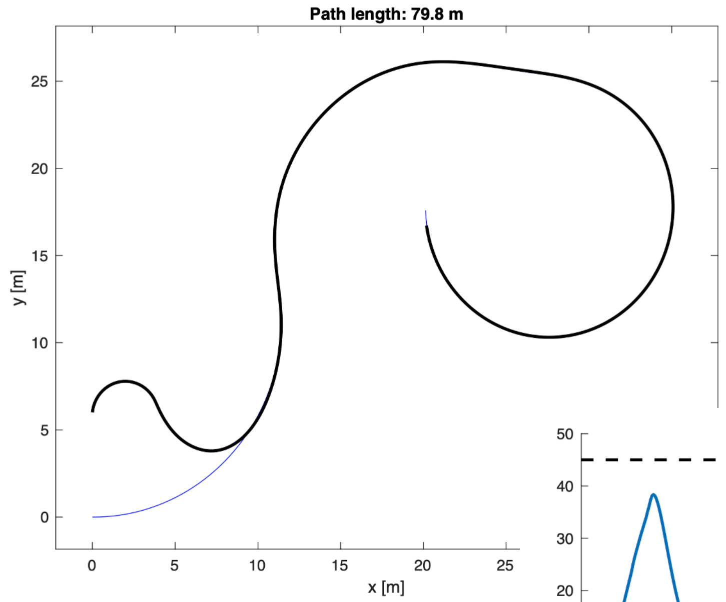
- Compared to the LQR approach, with **MPC** a **finite-horizon optimization problem** is instead solved at each sample instant, considering both the **non-linear vehicle model** and **constraints** on the input.
  - Results in a **computationally more expensive** controller.

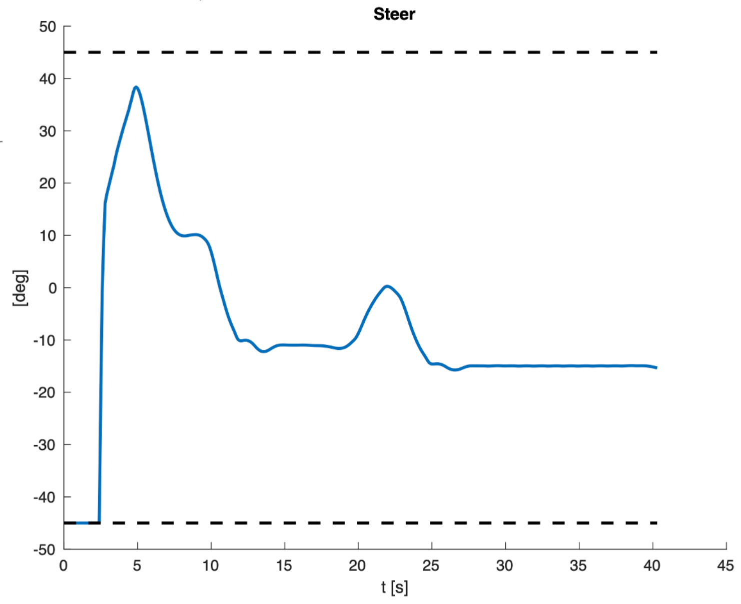
Example: Extra Assignment for Hand-in 3 (1/2)

- Part of extra assignment for **Hand-in Exercise 3** to implement this **path-following MPC**.
- **Linearization** of the non-linear kinematic single-track model at each MPC update step.
  - **Quadratic cost function** and **linear constraints**, leads to **convex optimization problem** (efficient solvers exist).

Example: Extra Assignment for Hand-in 3 (2/2)

- Application on an **example path** from Hand-in Exercise 3.
- **Constraints on control signal** (steering angle).
- Choices of **prediction horizon** and **sample period**.
- **Multiple shooting** with **explicit Runge-Kutta** method of fourth order for **discretization** of vehicle motion equations (see Lecture 5).





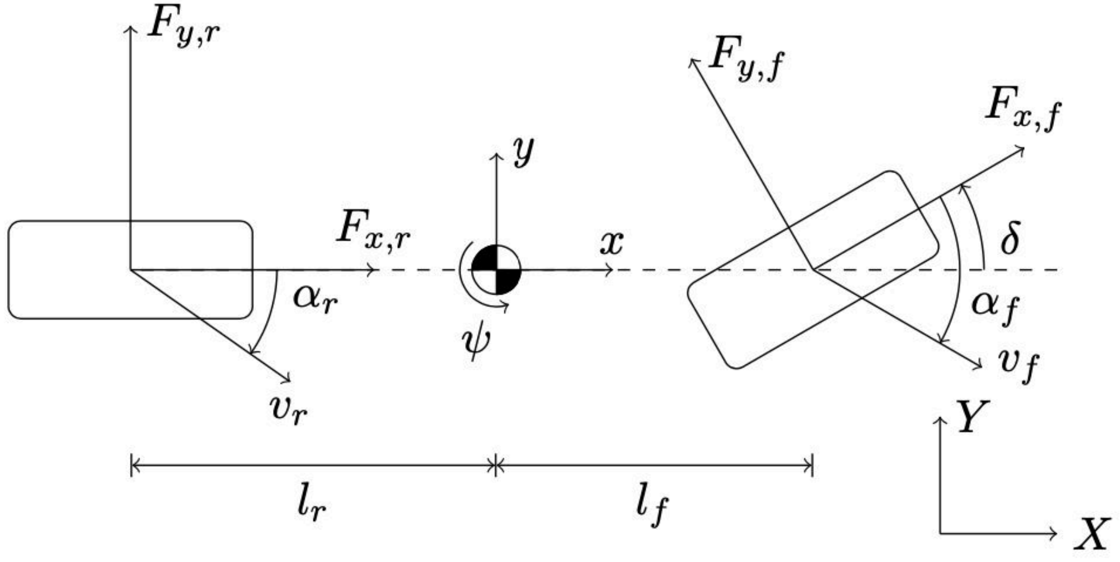
MPC for Autonomous Path Following

Problem Formulation

- The task is to **follow a desired path** (position and orientation) and a nominal **velocity profile**.
- To account for **dynamic effects** of a ground vehicle, a **single-track model** with a **linear tire–road interaction model** is used.
  - More **complex model** than the kinematic model in the previous example, beneficial for **higher velocities** and in **more advanced maneuvers**.
- Design an **MPC** (objective and constraints) for this task.

Dynamic Vehicle Model (1/4)

- Consider the **single-track vehicle model** (lumped wheels on each axle).
- **Front steering** and **front-wheel driven**.



$$m(\dot{v}_x - v_y\dot{\psi}) = F_{x,f}\cos(\delta) + F_{x,r} - F_{y,f}\sin(\delta),$$
$$m(\dot{v}_y + v_x\dot{\psi}) = F_{y,f}\cos(\delta) + F_{y,r} + F_{x,f}\sin(\delta),$$
$$I_Z\ddot{\psi} = l_f F_{y,f}\cos(\delta) - l_r F_{y,r} + l_f F_{x,f}\sin(\delta)$$

Dynamic Vehicle Model (2/4)

- The **slip angles** (angle between the velocity vector and the direction of the wheel) are given by:

$$\alpha_f = -\arctan\left(\frac{v_{f,y}}{v_{f,x}}\right), \qquad \alpha_r = -\arctan\left(\frac{v_{r,y}}{v_{r,x}}\right)$$

- A low-complexity **tire–road interaction model** for normal driving with linear tire stiffnesses is adopted:

$$F_{y,f} = C_{\alpha,f}\alpha_f, \qquad F_{y,r} = C_{\alpha,r}\alpha_r$$

Dynamic Vehicle Model (3/4)

- A **path parameter** is introduced to describe the traversal along the reference path (*traversal computed in the MPC*):

$$\dot{s} = \frac{\mathrm{d}s}{\mathrm{d}t}$$

- The **vehicle position** in the **global coordinate frame** is obtained by integration of the quantities:

$$\begin{pmatrix} \dot{p}_X \\ \dot{p}_Y \end{pmatrix} = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix}$$

Faulwasser, T., & Findeisen, R: "Nonlinear model predictive control for constrained output path following". IEEE Transactions on Automatic Control, 61(4), 1026-1039, 2015.

Dynamic Vehicle Model (4/4)

- Collect the **states** and the **control inputs** in the vectors:

$$x(t) = \begin{pmatrix} p_X & p_Y & \psi & v_x & v_y & \dot{\psi} & s \end{pmatrix}^{\mathrm{T}}$$

$$u(t) = \begin{pmatrix} \delta & F_{x,f} & F_{x,r} & \dot{s} \end{pmatrix}^{\mathrm{T}}$$

- With these variables, the **vehicle dynamics** can be written as an **explicit ordinary differential equation system** as:

$$\dot{x}(t) = f_{\mathrm{car}}(x(t), u(t))$$

MPC Problem (1/4)

- **Reference values** for the vehicle consist of **position** in global coordinate frame, **longitudinal velocity**, and **yaw orientation**.
- Introduce the **vector of reference values** as:

$$\begin{pmatrix} p_{X,\mathrm{ref}} & p_{Y,\mathrm{ref}} & v_{x,\mathrm{ref}} & \psi_{\mathrm{ref}} \end{pmatrix}^{\mathrm{T}}$$

MPC Problem (2/4)

- Introduce a function that describes the **error quantities** that the controller should **drive to zero** (*reference values as function of the path parameter* s):

$$\Phi(x(t)) = \begin{pmatrix} p_{X,\mathrm{ref}}(s) - p_X(t) \\ p_{Y,\mathrm{ref}}(s) - p_Y(t) \\ e_\perp(t,s) \\ \psi_{\mathrm{ref}}(s) - \psi(t) \\ v_{x,\mathrm{ref}}(s) - v_x(t) \end{pmatrix}$$

- The **orthogonal projection** of the path error is:

$$e_\perp(t,s) = \sin(\psi_{\mathrm{ref}})(p_{X,\mathrm{ref}}(s) - p_X(t)) - \cos(\psi_{\mathrm{ref}})(p_{Y,\mathrm{ref}}(s) - p_Y(t))$$

MPC Problem (3/4)

- The **finite-horizon MPC problem** to be solved at each sample instant can then be stated as:

$$\begin{aligned} \text{minimize} \quad & \int_{t_k}^{t_k+h_p} \Phi^{\mathrm{T}} Q \Phi + u^{\mathrm{T}} R u \,\mathrm{d}t \\ \text{subject to} \quad & x(t_k) = x_0,\ \dot{x}(t) = f_{\mathrm{car}}(x(t), u(t)), \\ & x(t) \in \mathbb{X},\ u(t) \in \mathbb{U},\ t \in [t_k, t_k + h_p] \end{aligned}$$

- The weights $Q,\ R$ are used to **trade different objectives** in the cost function to each other.

Berntorp, K., Quirynen, R., Uno, T., & Di Cairano, S: "Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control". Vehicle System Dynamics, 58(5), 705-725, 2020.

MPC Problem (4/4)

- There is **large flexibility** in terms of the choice of **constraints**.
- There are typically constraints on the **steering angle** and the **driving** and **braking forces**.
- Update of **path parameter** also constrained.
- **Geometric constraints** also possible.

$$\begin{aligned} \delta_{\min} &\leq \delta \leq \delta_{\max}, \\ F_{x,f,\min} &\leq F_{x,f} \leq F_{x,f,\max}, \\ F_{x,r,\min} &\leq F_{x,r} \leq F_{x,r,\max}, \\ 1 - s_{\Delta} &\leq \dot{s} \leq 1 + s_{\Delta} \end{aligned}$$

Solution of the MPC Problem

- At **each sample instant**, the finite-horizon **MPC problem is solved** with updated state measurements (or often, in practice *state estimates*).
- The model is non-linear, and thus the problem is a **non-linear MPC** (NMPC).
- **Numerical solution** of the **optimization problem** required.

Solving MPC Optimization Problems

- **A large number of optimization problems** are solved in sequence online in MPC.
- Often there are **real-time computational constraints** for an MPC controller executing **online**.
- How can the associated **optimization problem** be **solved efficiently** and the **special structure** of it be utilized?

Solving MPC Optimization Problems Using RTI

- One proposed method is the **Real-Time Iteration (RTI)** scheme:
  - Multiple shooting for discretization of continuous dynamics.
  - Sequential quadratic programming (**SQP**), using one single Newton step, and an effective warm starting by using the state and control input trajectories from the previous MPC iteration.
  - Separation of computations in ***preparation phase*** and ***feedback response phase*** (highly important in real-time implementations).

Diehl, M., Bock, H. G., & Schlöder, J. P: "A real-time iteration scheme for nonlinear optimization in optimal feedback control". SIAM Journal on Control and Optimization, 43(5), 1714-1736, 2005.
Gros, S., Zanon, M., Quirynen, R., Bemporad, A., & Diehl, M: "From linear to nonlinear MPC: Bridging the gap via the real-time iteration". International Journal of Control, 93(1), 62-80, 2020.

Summary and Outlook of Model Predictive Control for Autonomous Vehicles

LINKÖPING UNIVERSITY

Characteristics of Model Predictive Control

- **Integrated planning** and **control** possible.
- Possible to apply at **different layers** of the **decision-making architecture** of an autonomous vehicle, **MPC offers a framework**.
- **Automatic inherent re-planning** obtained at each sample instant, ***implicit feedback mechanism***.
- **High flexibility** in terms of **modeling** and **constraints**.
  - Also one of the main challenges in the MPC design.

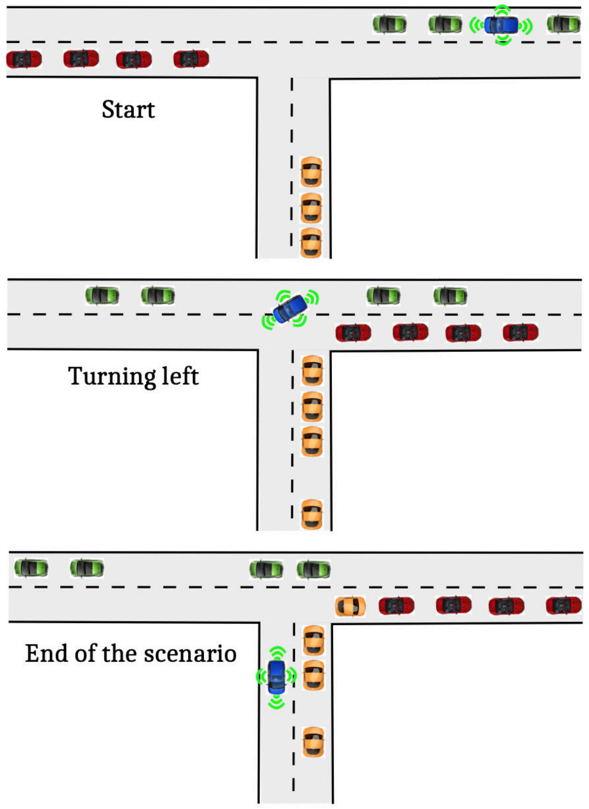
MPC Outlook (1/2)

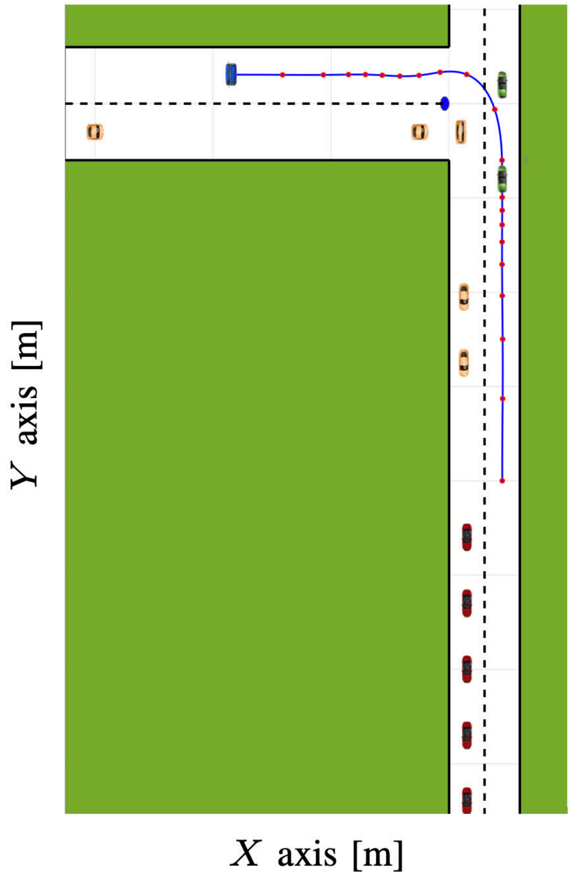
- **Decentralized formulations** and **optimization** for **multiple vehicles** – reduce computational cost at the local node.
  - Trade-off with **communication load**.
- Dual to MPC for **state estimation** – **Moving horizon estimation (MHE)**.
  - Allows explicit constraints in the estimation problem.

Rao, C. V., Rawlings, J. B., & Mayne, D. Q: "Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations". IEEE Transactions on Automatic Control, 48(2), 246-258, 2003.

MPC Outlook (2/2)

- Integrated **collision avoidance** and **local planning** in **complex traffic scenarios** with MPC.
- **Obstacle representation, obstacle motion prediction,** and **collision-risk estimates.**





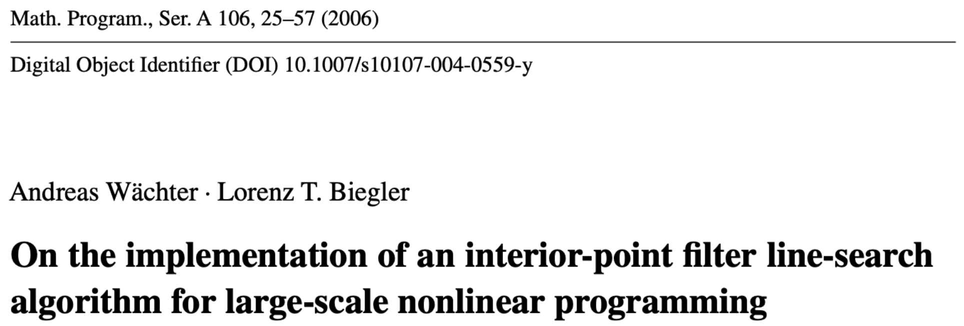
F. Mohseni, B. Olofsson, & E. Frisk: "Multi-Vehicle Autonomous Driving Using a Gaussian Risk and Model Predictive Control", 2020.

Tools and Libraries for Numerical Optimal Control and Model Predictive Control

CasADi

- Open-source tool for **numerical optimization and optimal control**.
- Includes efficient algorithms for **automatic differentiation** and computing Jacobians and Hessians in forward and backward mode.
- **Interfaces** to various NLP solvers and numerical integrators for ODEs/DAEs.
- Homepage: https://web.casadi.org

IPOPT


Math. Program., Ser. A 106, 25–57 (2006)

Digital Object Identifier (DOI) 10.1007/s10107-004-0559-y

Andreas Wächter · Lorenz T. Biegler

On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming

- Software package for **solving large-scale NLPs**.
- Implements a **primal-dual interior-point method** for finding local minima to the NLP.
- **Scaling** of model equations for numerical stability.
- Relies on external software packages for solving the **inherent linear equation systems** (e.g., MA27 and MA57 from HSL Mathematical Software Library).
- Homepage: https://github.com/coin-or/Ipopt

YOP

- YOP – Yet Another Optimal Control Problem Parser.
- **Matlab toolbox** for **numerical optimal control**, with an interface to CasADi for access to **integrators** and **solvers of nonlinear programs**.
- Developed at Div. Vehicular Systems at Linköping University.
- Several **examples** available on toolbox homepage: https://www.yoptimalcontrol.se/index.html

References and Further Reading

LINKÖPING
UNIVERSITY

References and Further Reading (1/2)

All the following books and articles are not part of the reading assignments for the course, but cover the topics studied during this lecture in more detail.

- Andersson, J., J. Gillis, G. Horn, and J. B. Rawlings, & M. Diehl: ”CasADi–A software framework for nonlinear optimization and optimal control”. *Mathematical Programming Computation*, 2018.
- Berntorp, K., Quirynen, R., Uno, T., & Di Cairano, S: ”Trajectory tracking for autonomous vehicles on varying road surfaces by friction-adaptive nonlinear model predictive control”. *Vehicle System Dynamics*, 58(5), 705-725, 2020.
- Bertsekas, D. P: *Reinforcement learning and optimal control*. Belmont, MA: Athena Scientific, 2019.
- Di Cairano, S., Kalabić, U. V., & Berntorp, K: ”Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds”. In IEEE Conference on Decision and Control (CDC), 709-714, 2016.
- Diehl, M., Bock, H. G., & Schlöder, J. P: ”A real-time iteration scheme for nonlinear optimization in optimal feedback control”. *SIAM Journal on Control and Optimization*, 43(5), 1714-1736, 2005.

References and Further Reading (1/2)

All the following books and articles are not part of the reading assignments for the course, but cover the topics studied during this lecture in more detail.

- Gros, S., Zanon, M., Quirynen, R., Bemporad, A., & Diehl, M: ”From linear to nonlinear MPC: Bridging the gap via the real-time iteration”. *International Journal of Control*, 93(1), 62-80, 2020.
- Faulwasser, T., & Findeisen, R: ”Nonlinear model predictive control for constrained output path following”. *IEEE Transactions on Automatic Control*, 61(4), 1026-1039, 2015.
- Hellström, E., Ivarsson, M., Åslund, J., & Nielsen, L: ”Look-ahead control for heavy trucks to minimize trip time and fuel consumption”. *Control Engineering Practice*, 17(2), 245-254, 2009.
- Paden, B., Čáp, M., Yong, S. Z., Yershov, D., & Frazzoli, E: ”A survey of motion planning and control techniques for self-driving urban vehicles”. *IEEE Transactions on Intelligent Vehicles*, 1(1), 33-55, 2016.
- Rao, C. V., Rawlings, J. B., & Mayne, D. Q: ”Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations”. *IEEE Transactions on Automatic Control*, 48(2), 246-258, 2003.
- Rawlings, J. B., D. Q. Mayne, & M. Diehl: *Model Predictive Control: Theory, Computation, and Design*. 2nd Edition. Nob Hill Publishing, 2017.

www.liu.se

LINKÖPING UNIVERSITY