Institute of Automotive Technology
Faculty for Mechanical Engineering
Technical University of Munich

Autonomous Driving Software Engineering

Prof. Dr.-Ing. Markus Lienkamp

Nico Uhlemann, Dipl.-Ing.

Simon Sagmeister, M. Sc.

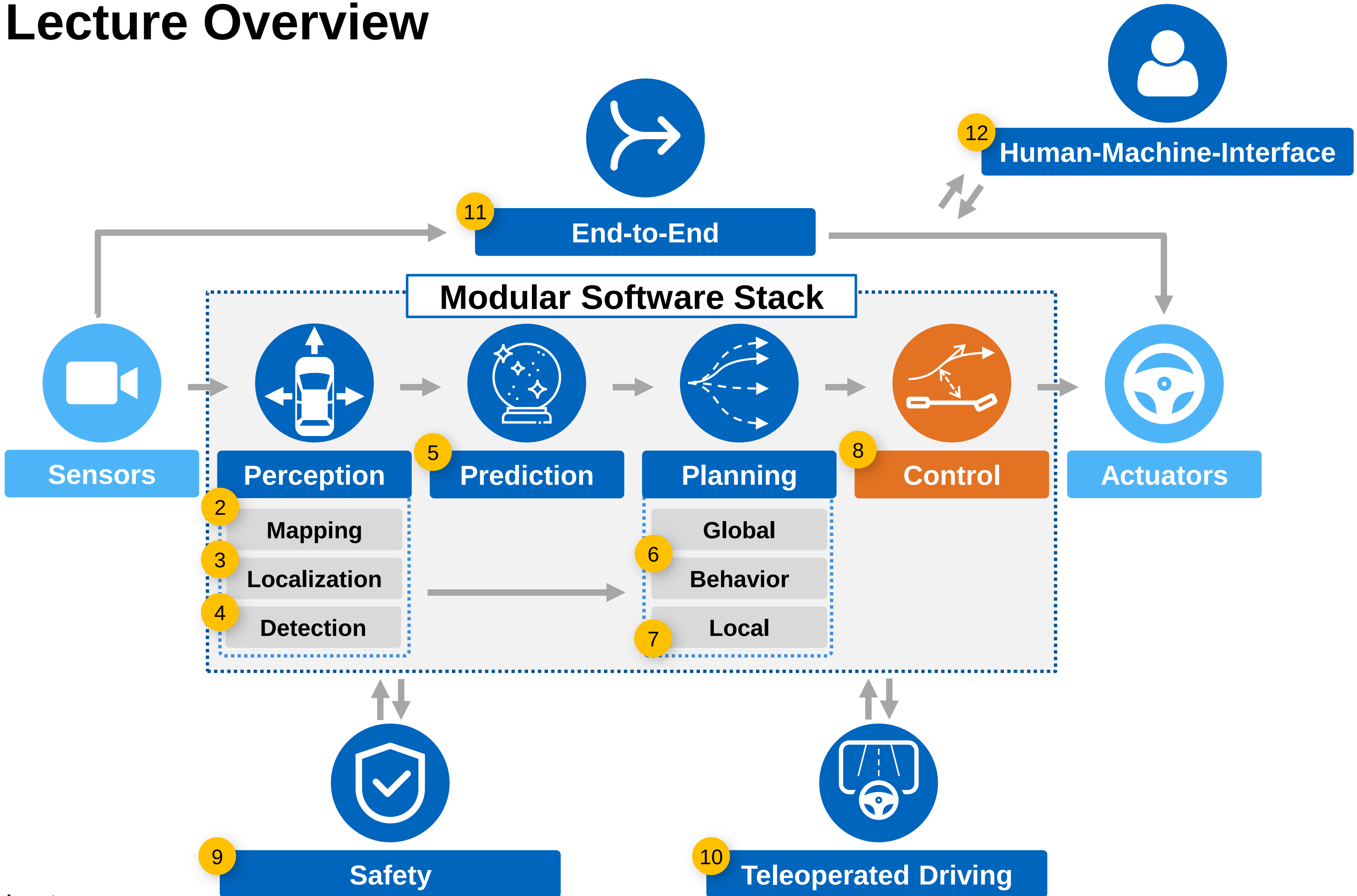
Autonomous Driving Software Engineering
Prof. Dr. Markus Lienkamp

Simon Sagmeister, M. Sc.

Agenda

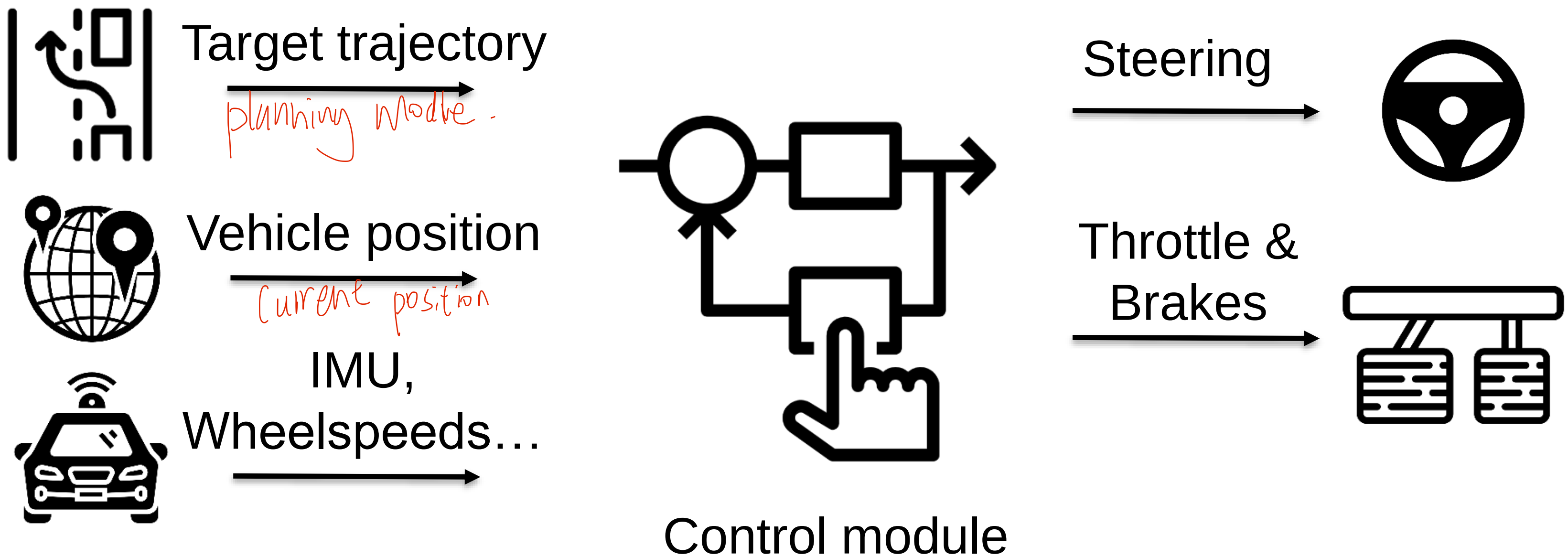
1. **Introduction**
2. Basic control (Geometric & PID)
3. Model-based control
4. Model predictive control
5. Summary

Lecture Overview



X = Lectures

Role of control in the AV software stack



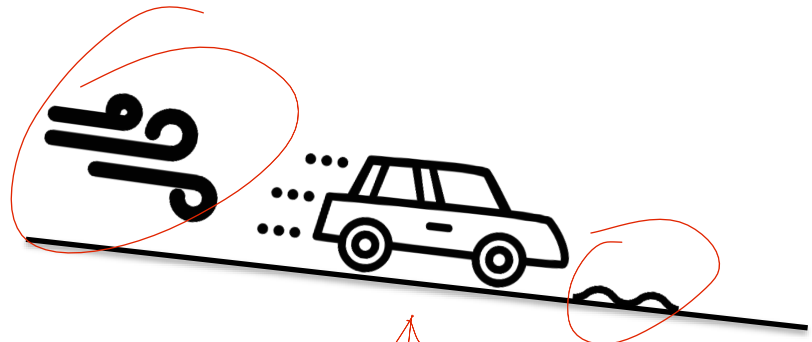
- Control is focussed on the short-term behavior (roughly 2 seconds) and updates frequently (at minimum with 20 Hertz)
- Real-time operation is crucial
- Control needs maximum knowledge about the vehicle

Why do we actually need feedback control?

How engineers like to think about the world:

model, ideal world

What it's actually like:



real world

- The real world is full of uncertainties and disturbances
- Control is about taking into account those uncertainties and manage them in a structured way to achieve the desired task

The evolution of autonomous vehicle control

Geometric controllers

Model-based feedforward and feedback controllers

Model predictive controllers

Stanley – Stanford (2005)

Talos – MIT (2007)

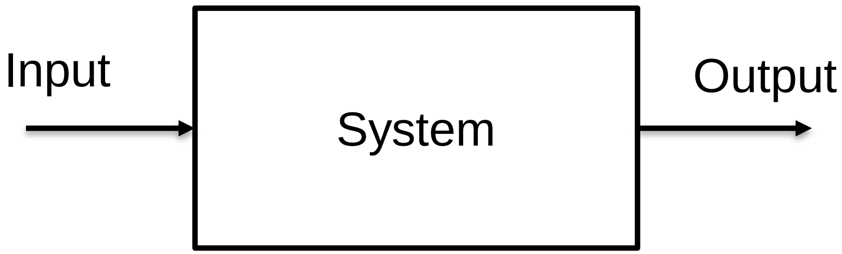
Shelley – Stanford (2015)

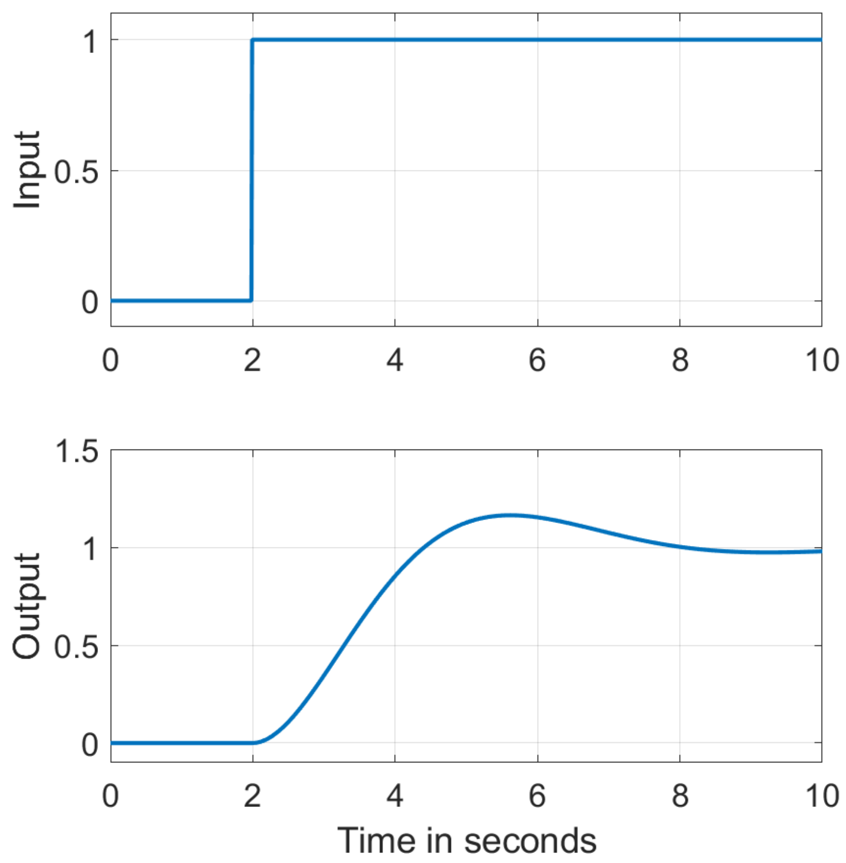
Roborace – TUM (2018)

1:43 Racing - ETH (2015)

Indy Challenge – TUM (2021)

Control theory basics – System dynamics I





Examples:

- Spring-mass-damper systems
- Inverse pendulum
- Autonomous vehicles

Control theory basics – System dynamics II

System matrix State Input matrix = out of system

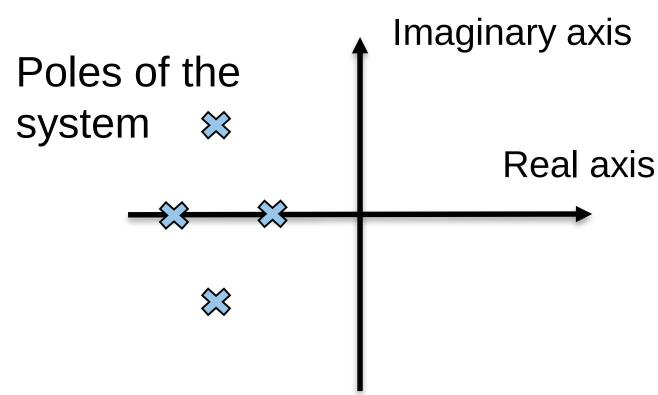
$$\dot{x} = Ax + Bu$$ ← Input

Output → $$y = Cx + Du$$

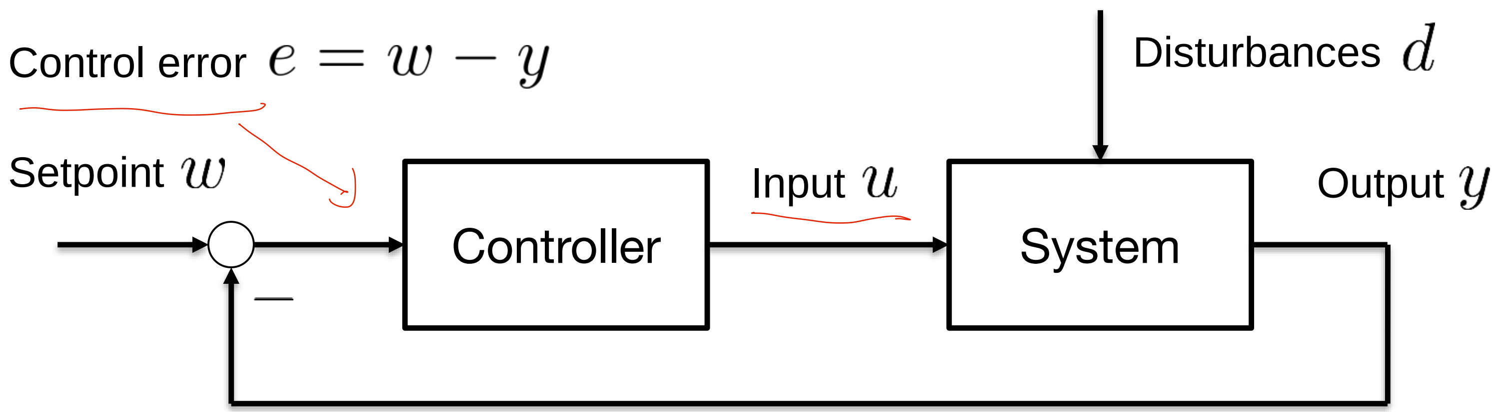
Output matrix Feedthrough matrix

System dynamics are given from the spectrum of A:

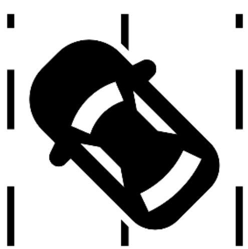
$$\det(sI - A) = 0$$



Control theory basics – Standard control loop I

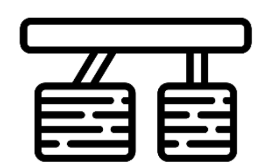


Examples for control loops in autonomous vehicles:



Vehicle motion

Steering angle



Brake pressure

Control theory basics – PID controller design I

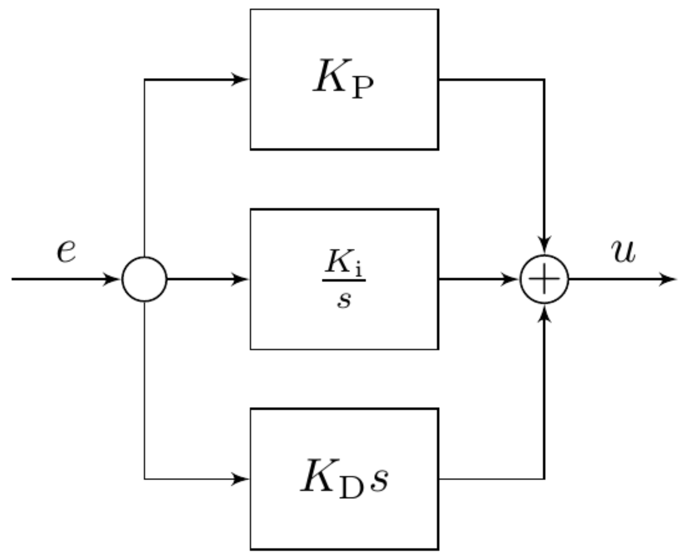
Control error

$$u = k_{\mathrm{P}} e + k_{\mathrm{I}} \int e \, dt + k_{\mathrm{D}} \dot{e}$$

Control value

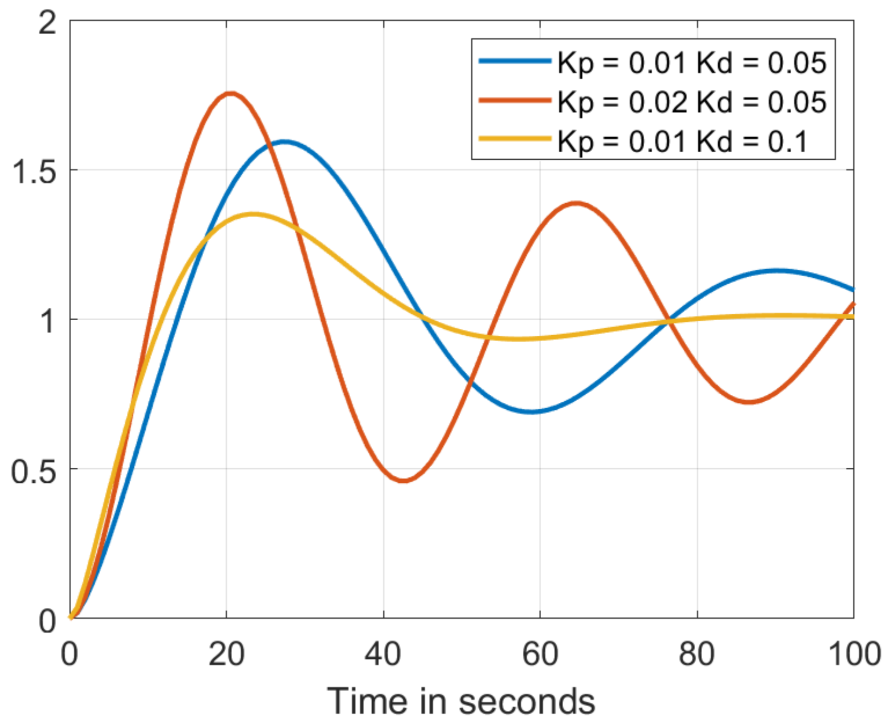
Control gains

Block diagram with transfer functions:



- Three main parts: **P**roportional / **I**ntegral / **D**erivative
- Standard approach for feedback control in industrial applications

Control theory basics – PID controller design II



- Tuning often done empirically during system operation
- Works well for systems with low complexity (e.g. mass-spring-damper system)

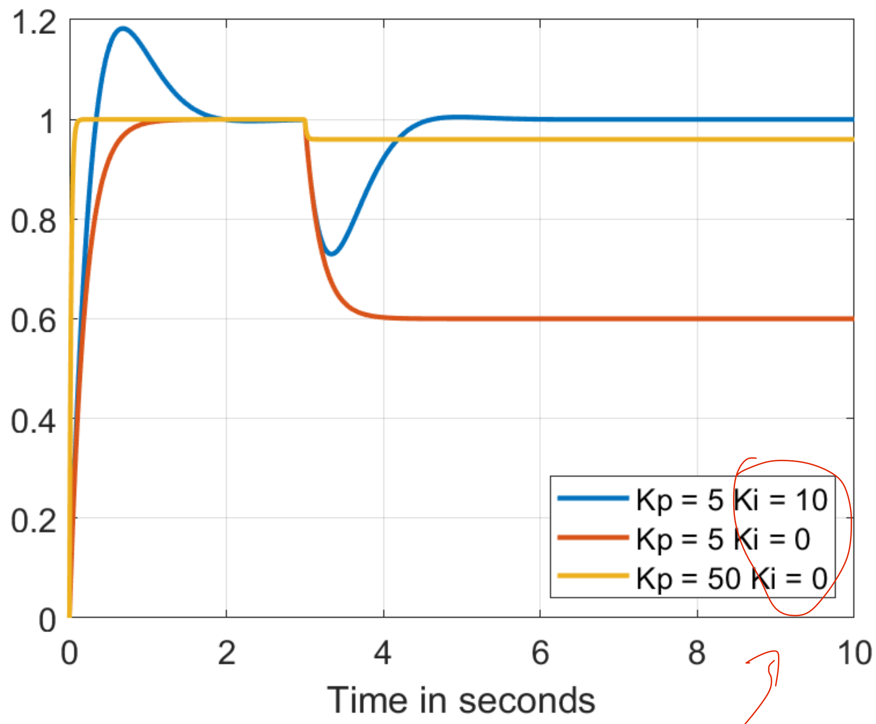
在系统运行过程中经常根据经验进行调整

对低复杂度的系统有很好的效果（如大规模弹簧破坏系统）

Control theory basics – PID controller design III

在存在不确定性或外部干扰的情况下，零稳定状态需要整体行动

替代方案：通过高反馈增益来抑制干扰，但这可能会导致现实世界中的振荡



- Zero steady-state in the presence of uncertainty or external disturbances requires integral action
- Alternative: high feedback gains to suppress disturbances but this might lead to oscillations in real world scenarios

Control theory basics – State-space design

- Algorithmic method based on designing a state-feedback matrix

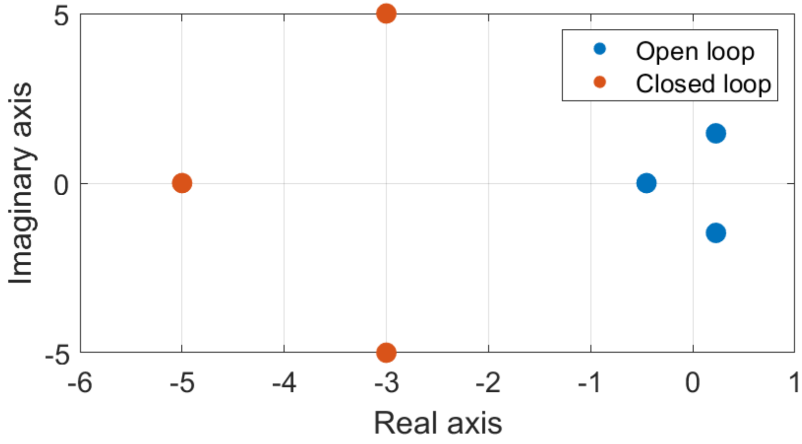
$$\dot{x} = Ax + Bu$$
$$u = -\boxed{K}x$$

state vektor

- Analyze characteristic equation of closed-loop dynamics

$$\dot{x} = (A - BK)\,x$$

- Possibilities to determine controller
  - Pole-placement via coefficient comparison
  - LQR controller design



Autonomous Driving Software Engineering
Prof. Dr. Markus Lienkamp

Simon Sagmeister, M. Sc.

Agenda

Decoupling of longitudinal and lateral dynamics

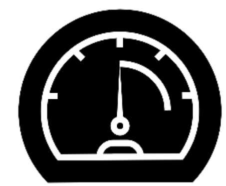
Joint trajectory tracking control

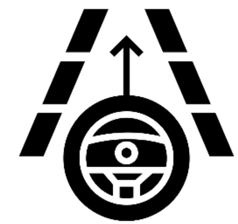
✓ Real vehicles have coupled dynamics

✗ Difficult to handle due to nonlinearities

✗ Advanced control algorithms necessary

Decoupled longitudinal and lateral control





✓ Controllers are easy to design and tune

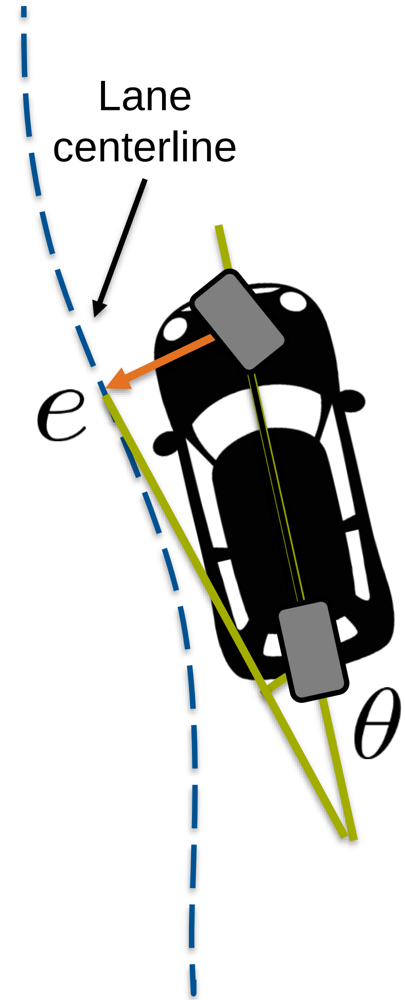
✗ Less accurate for high acceleration and dynamic maneuvers

Stanley controller

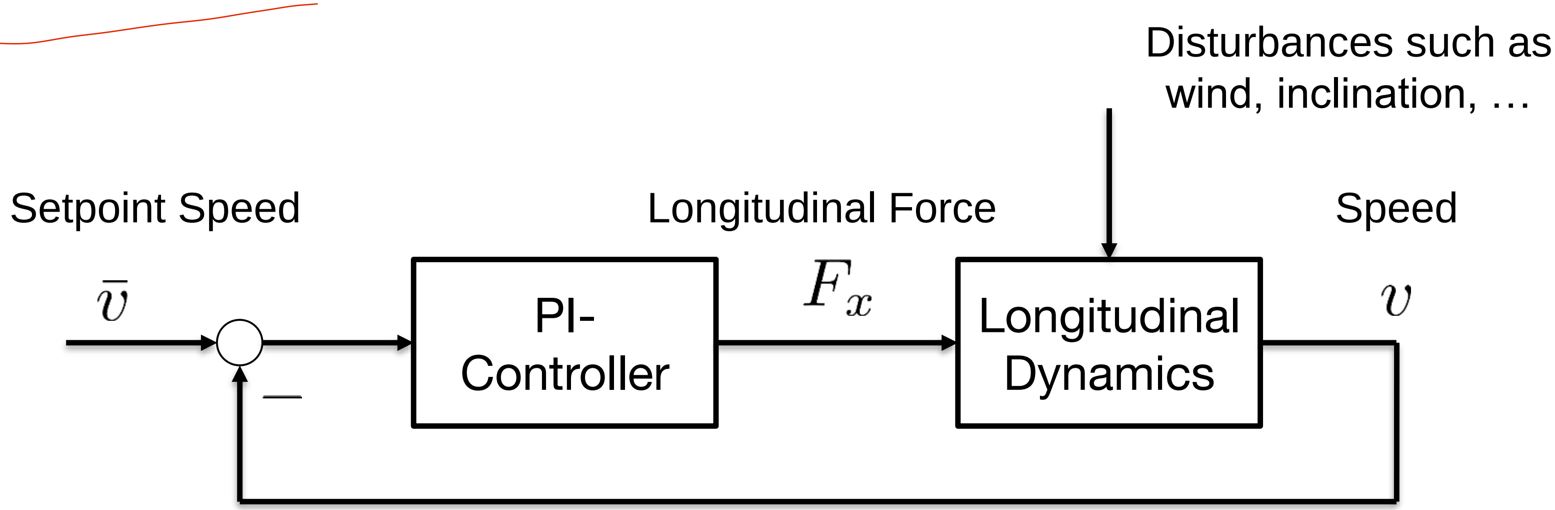
- Determine cross-track error e and heading error θ at the front axle
- Align front wheels with path and add a correction term

$$\delta = \theta + \arctan\left(\frac{ke}{v}\right)$$

- The control gain k adjusts the convergence speed



Velocity PI controller



$$F_x = -k_\mathrm{p} e - k_\mathrm{i} \int e \, dt \qquad e = \bar{v} - v$$

- Longitudinal dynamics are a single integrator dynamic, therefore no derivative part is needed
- Integrator required to achieve zero steady-state error due to external disturbance

Autonomous Driving Software Engineering
Prof. Dr. Markus Lienkamp

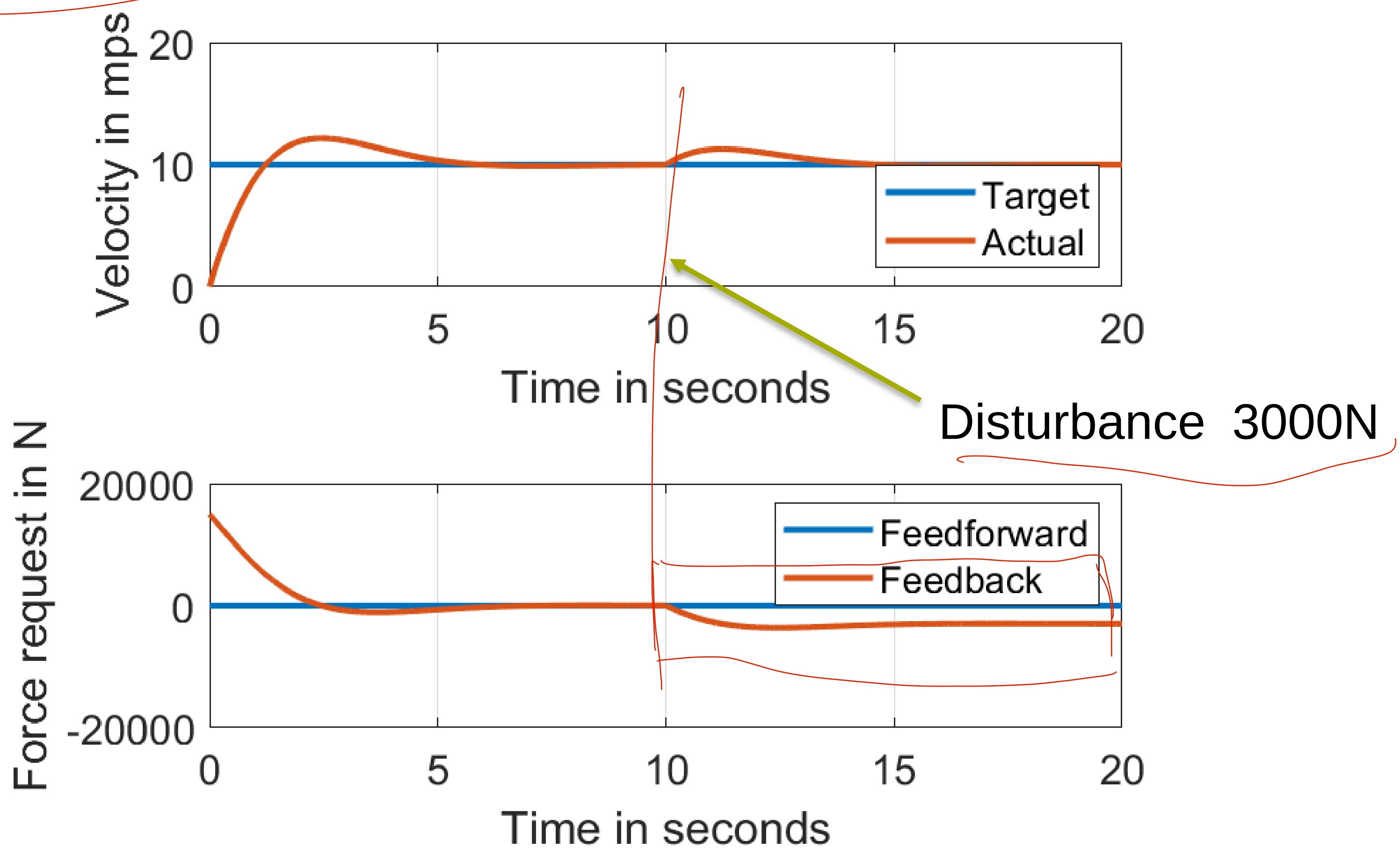
Simon Sagmeister, M. Sc.

Agenda

1. Introduction
2. Basic control (Geometric & PID)
3. **Model-based control**
4. Model predictive control
5. Summary

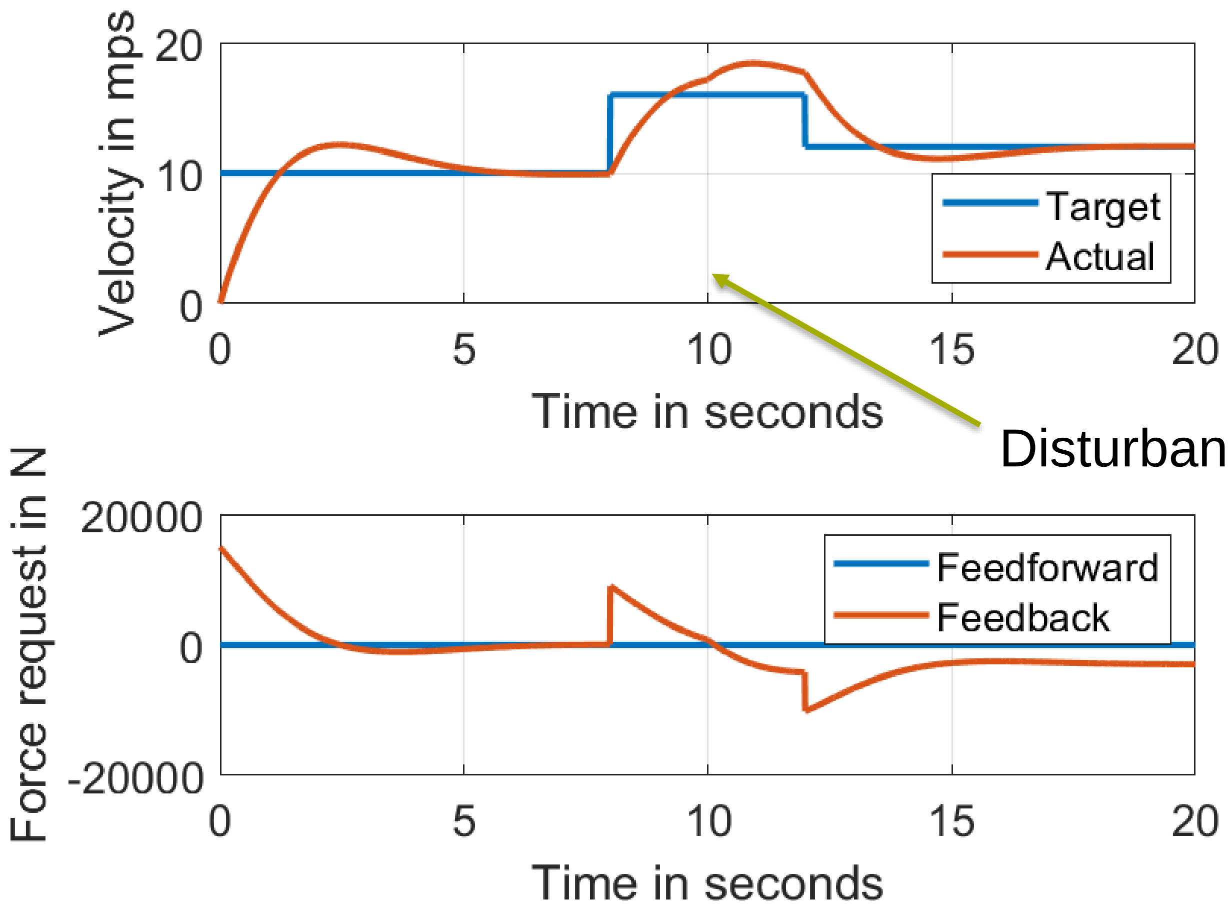

Improved velocity control with model knowledge

Setpoint **stabilization** with PI-controller:



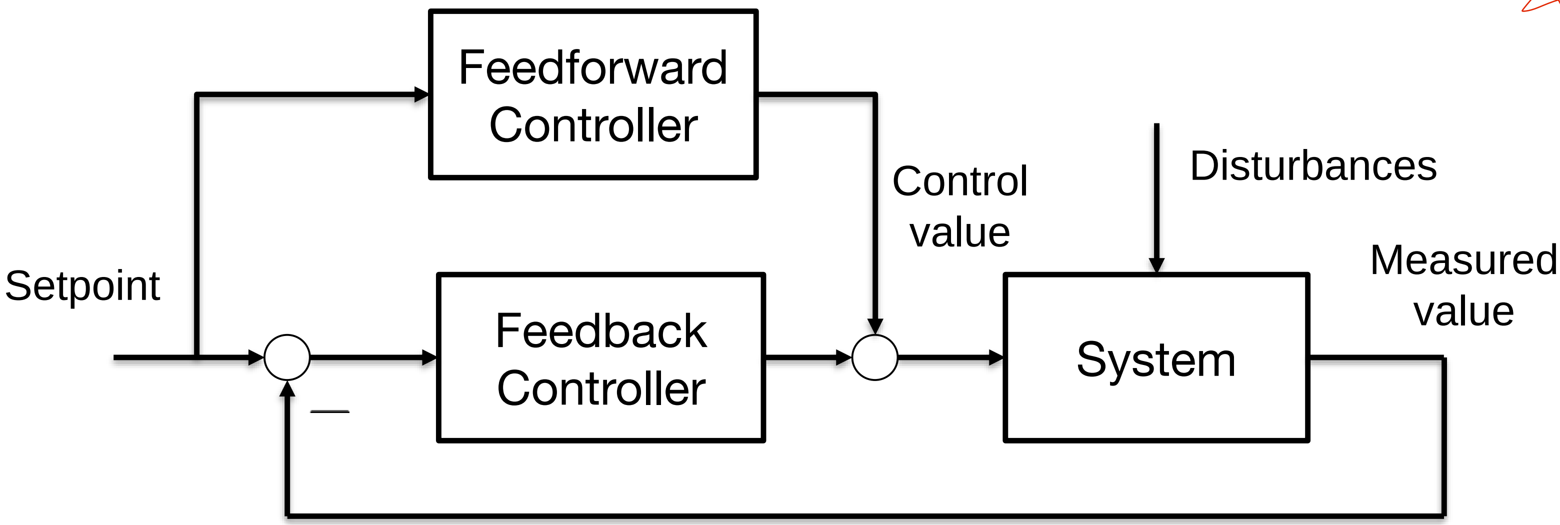
Improved velocity control with model knowledge

Setpoint **tracking** (e.g. changing speed limit) with PI-controller:



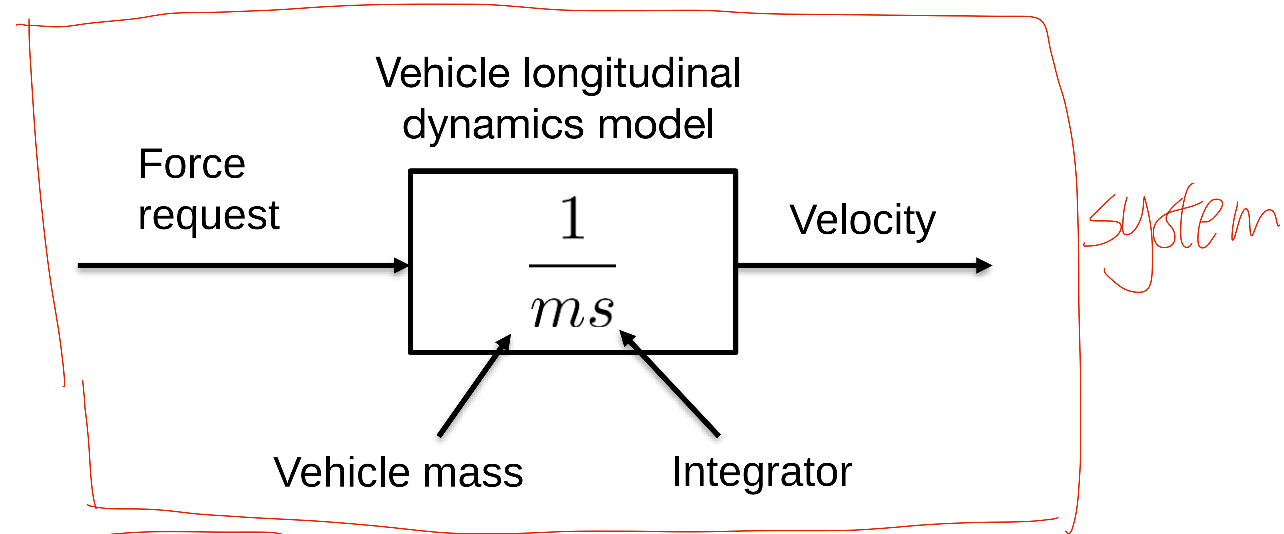
Disturbance 3000N

Improved velocity control with model knowledge

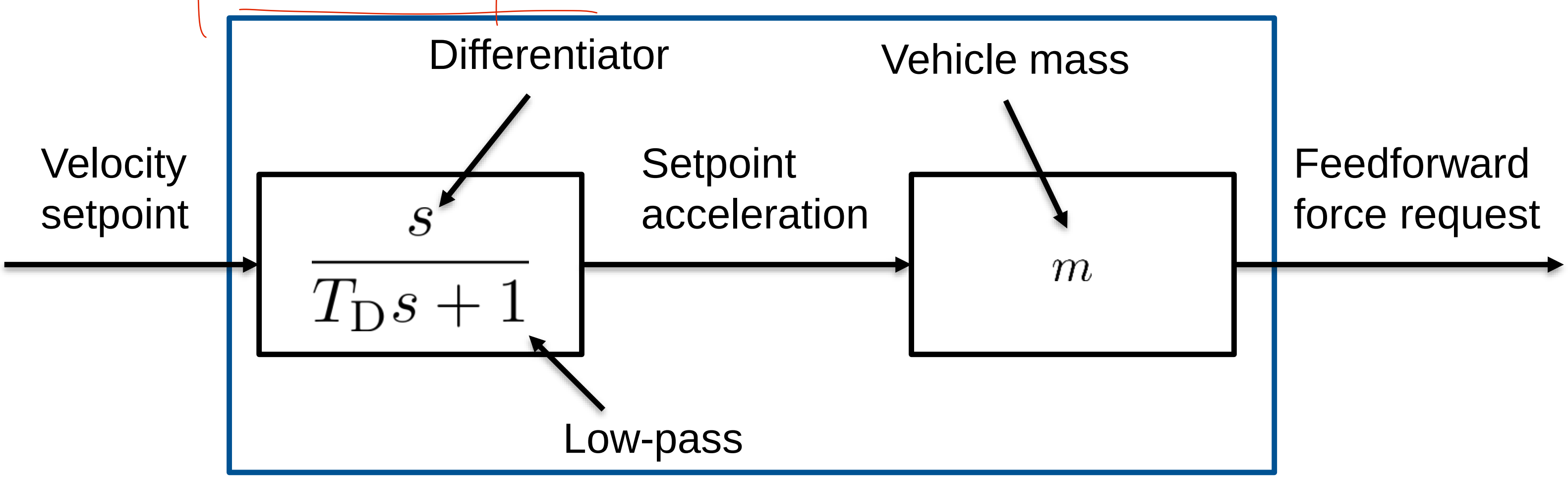


- Two degree of freedom structure allows to design separate dynamics for setpoint changes and disturbance influences
- One of the most common structures in industrial applications

Improved velocity control with model knowledge

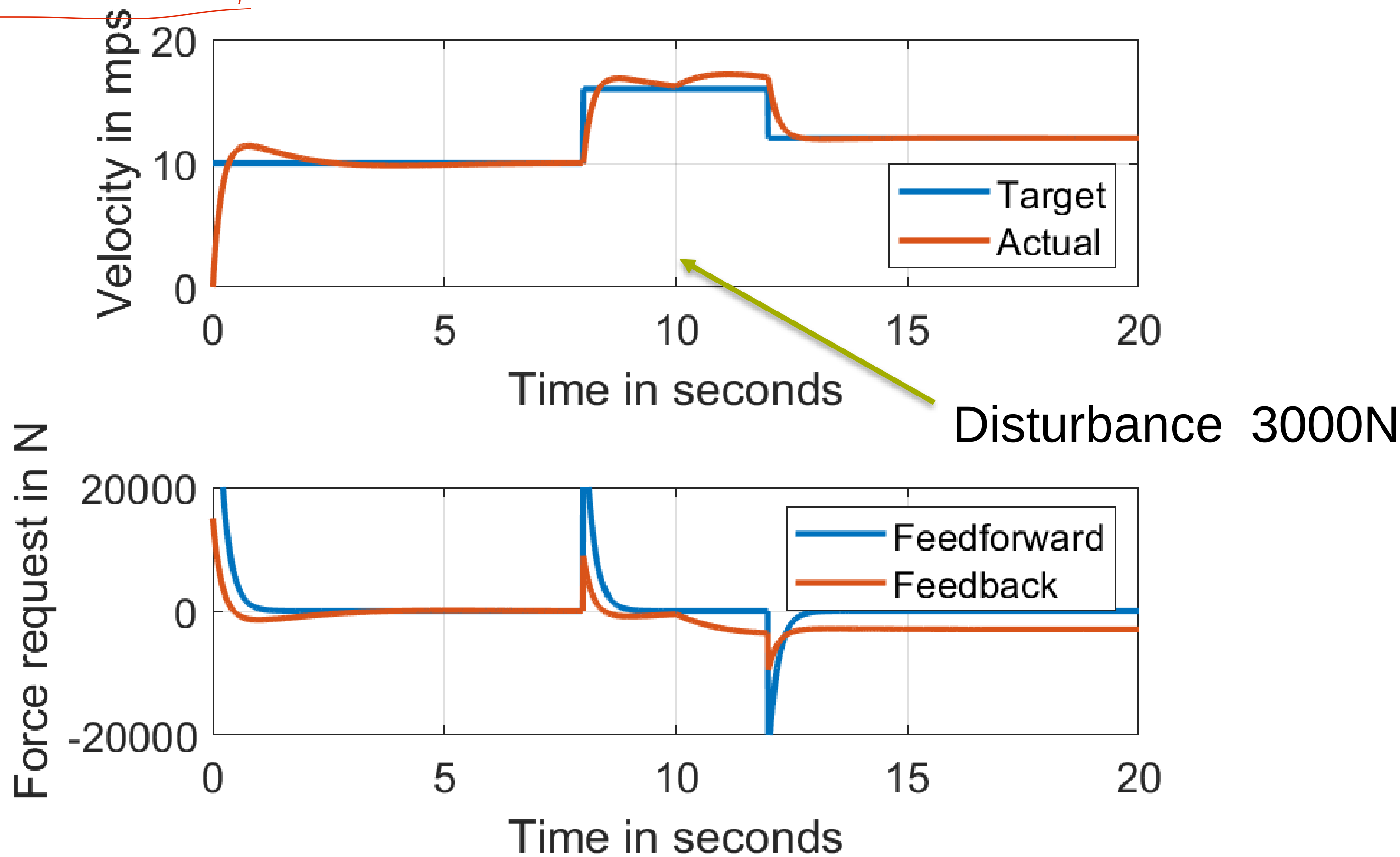


System inverse for feedforward control



Improved velocity control with model knowledge

Setpoint **tracking** (e.g. changing speed limit) with PI-controller and feedforward control:



Kinematic single track model

- Dynamic equations derived from *kinematic* constraints only → no tire slip and forces

$$\dot{x} = f(x, u)$$

$$x = \begin{bmatrix} p_1 & p_2 & \Psi \end{bmatrix}^{\mathrm{T}}$$

position heading

$$u = \begin{bmatrix} \delta & v \end{bmatrix}^{\mathrm{T}}$$

Steering angle

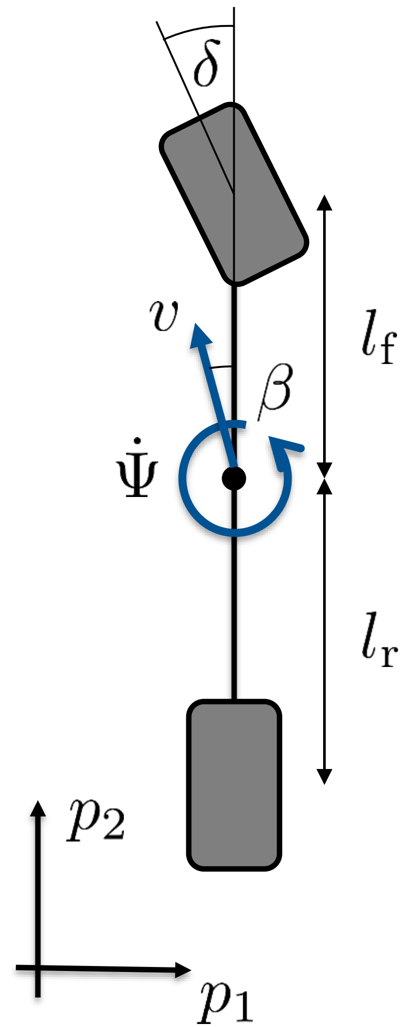
Vehicle speed

$$\dot{p}_1 = v \cos\left(\Psi + \beta(\delta)\right)$$

$$\dot{p}_2 = v \sin\left(\Psi + \beta(\delta)\right)$$

$$\dot{\Psi} = \frac{\tan(\delta)}{l_{\mathrm{f}} + l_{\mathrm{r}}} v$$

$$\beta = \arctan\left(\frac{\tan(\delta) l_{\mathrm{R}}}{l_{\mathrm{f}} + l_{\mathrm{r}}}\right)$$



Important relations in vehicle dynamics

Steady-state acceleration

$$a_y = \kappa v^2$$

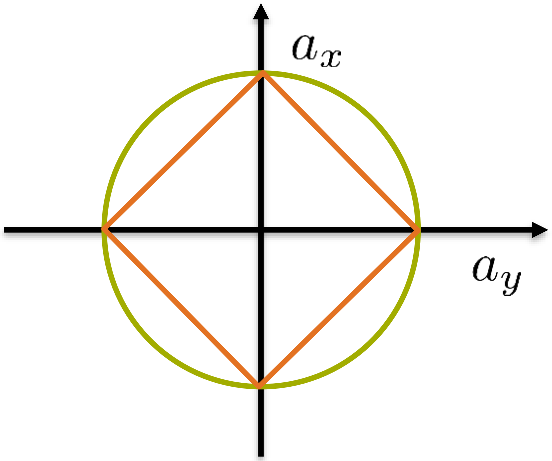
Steady-state yaw rate

$$\dot{\Psi} = \kappa v$$

Steady-state curvature

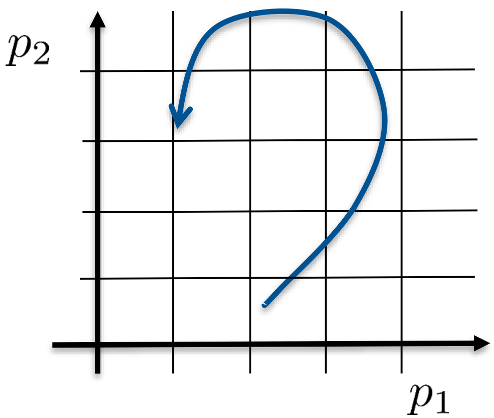
$$\kappa = \frac{\tan(\delta)}{l_\mathrm{f} + l_\mathrm{r}} \approx \frac{\delta}{l_\mathrm{f} + l_\mathrm{r}}$$

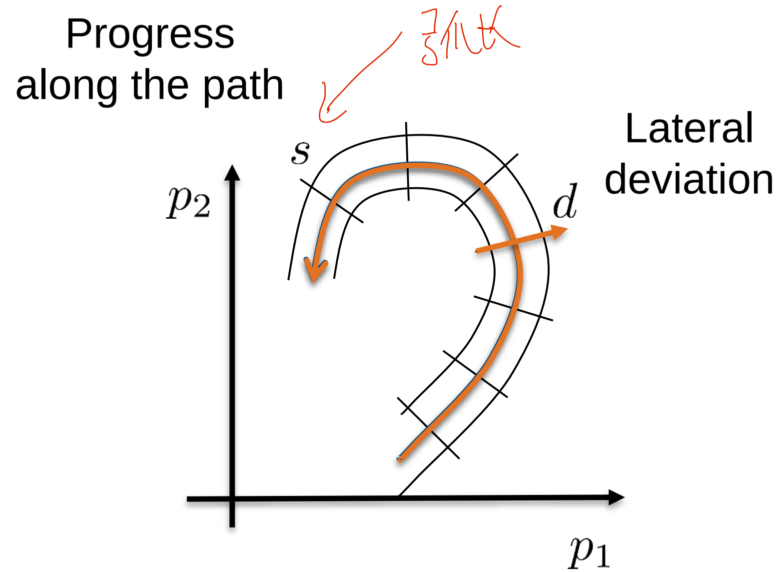
GG-Diagram



- Max. combined accelerations for overall vehicle
- Different forms are possible depending on the vehicle

Frenet coordinate frame

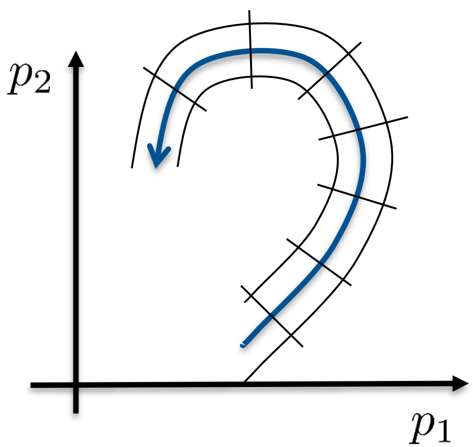


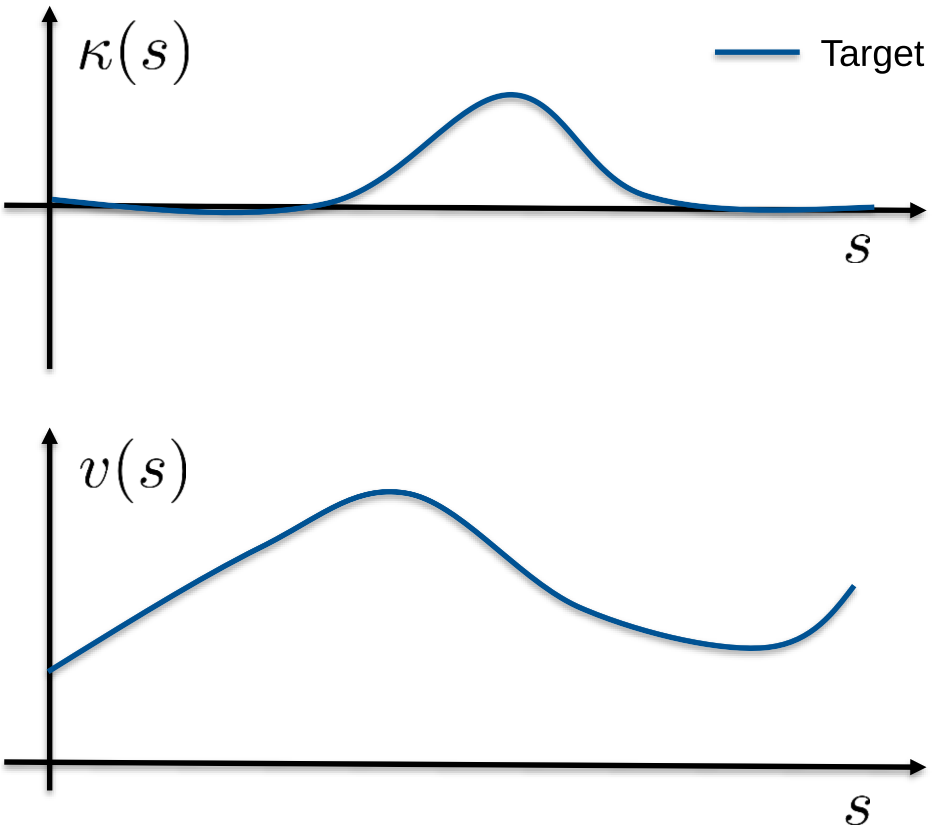


$$\dot{p}_1 = v\cos(\Psi + \beta)$$
$$\dot{p}_2 = v\sin(\Psi + \beta)$$
$$\dot{\Psi} = \dot{\Psi}$$

$$\dot{s} = \frac{1}{1 - d\kappa(s)} v\cos\left(\Delta\Psi + \beta\right)$$
$$\dot{d} = v\sin\left(\Delta\Psi + \beta\right)$$
$$\Delta\dot{\Psi} = \kappa(s)\dot{s} - \dot{\Psi}$$

Trajectory specification in Frenet coordinates





Improved lateral control with model knowledge

- Reformulation of the lateral path tracking problem in Frenet-coordinates
- Differentiating twice leads to double-integrator dynamics in the curvilinear coordinate frame

$$\begin{bmatrix} \dot{d} \\ \ddot{d} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d \\ \dot{d} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_{y,\mathrm{C}}$$

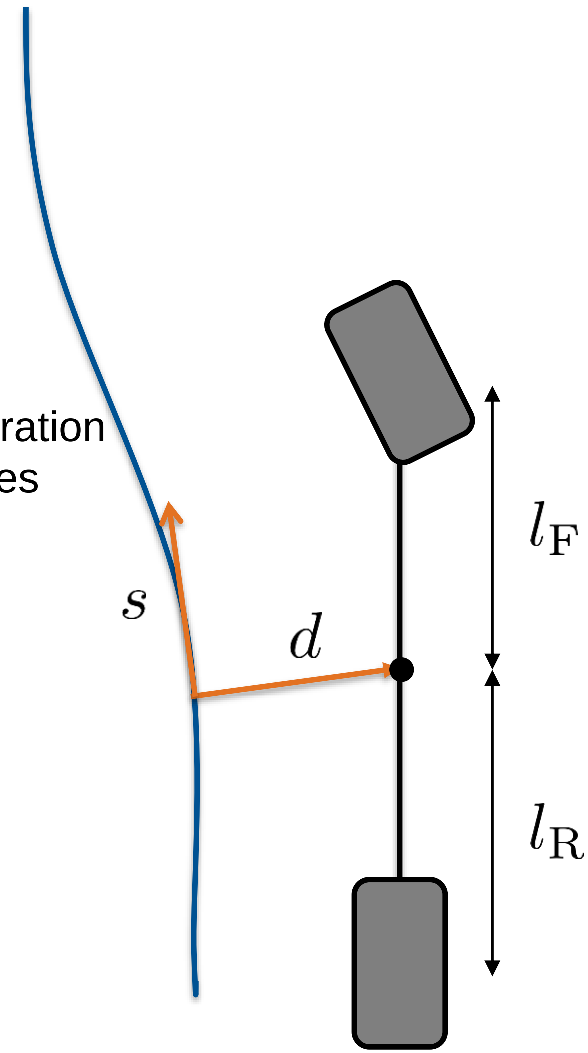
corrective acceleration in path coordinates

- For small errors this is linear w.r.t. the acceleration in vehicle coordinates

$$a_y \approx a_{y,\mathrm{P}} + a_{y,\mathrm{C}}$$

Lateral acceleration required to stay on the path (feedforward)

Corrective lateral acceleration (feedback)



Improved lateral control with model knowledge

- Control law has the structure of a 2-DOF gain-scheduling PD controller

$$\kappa_c = \frac{a_{y,\mathrm{P}} + a_{y,\mathrm{C}}}{v^2} = \kappa_\mathrm{P} - \frac{k_1 d + k_2 \dot{d}}{v^2}$$

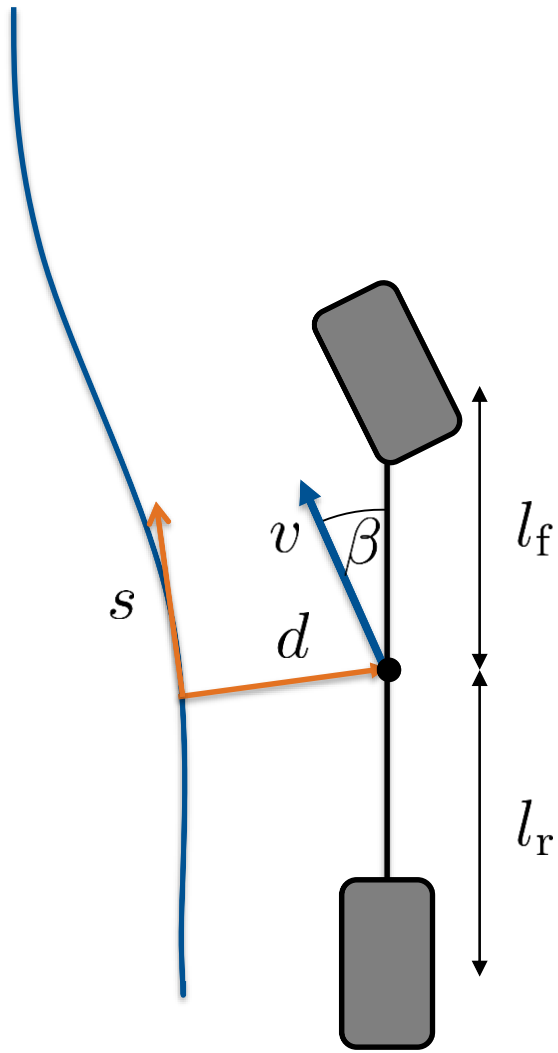
- Calculate steering angle from kinematic bicycle model

$$\delta = \kappa_c \left(l_\mathrm{F} + l_\mathrm{R} \right)$$

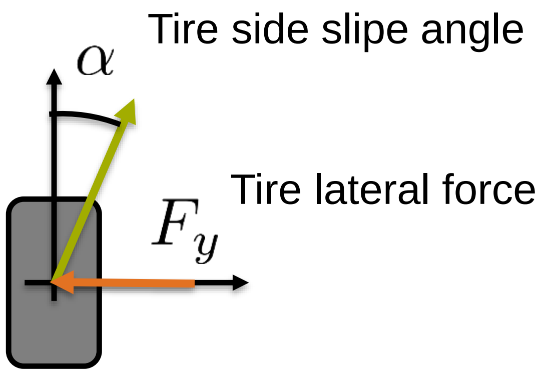
- Calculate lateral error derivative from

$$\dot{d} = v \sin\left(\Delta\Psi + \beta \right)$$

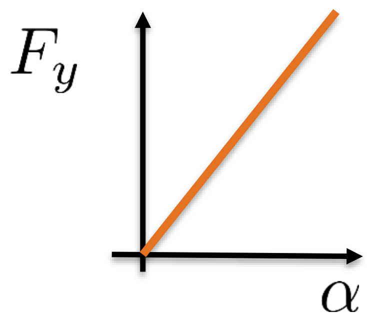
Velocity heading error



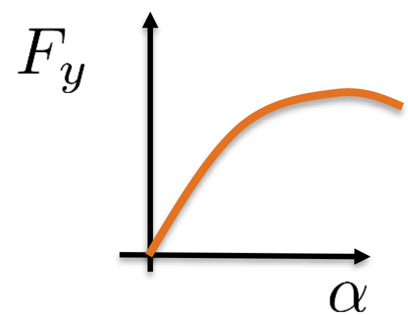
Advanced motion control – I



Linear tire model



Nonlinear tire model

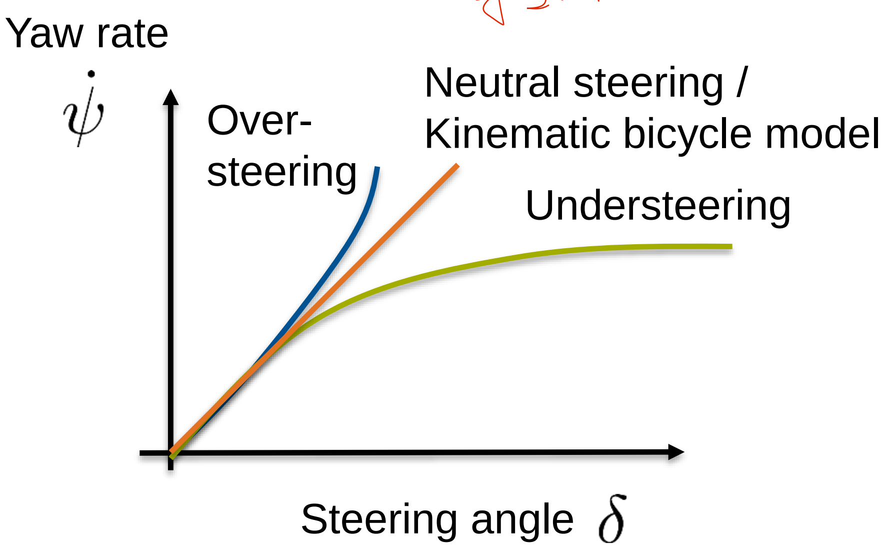


Cornering stiffness $$c_\alpha = \frac{\partial F_y}{\partial \alpha}$$

- Advanced vehicle models require detailed modelling of the tire
- Kinematic rolling condition is replaced with tire slip and tire models

Advanced motion control – II

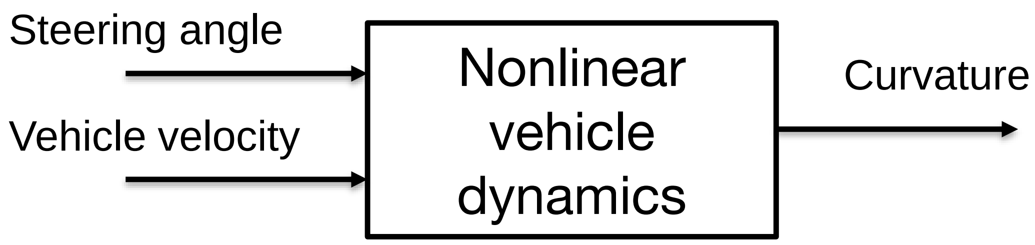
转向不足/过度

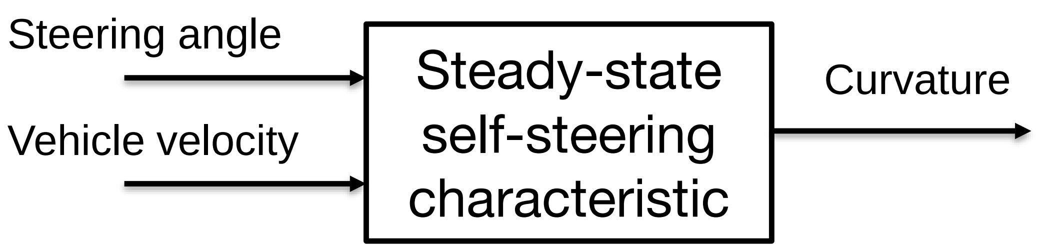


$$\dot{\Psi} = \frac{\tan(\delta)}{l_{\mathrm{f}} + l_{\mathrm{r}}} v$$

- Self-steering characteristic is most important factor
- Depends heavily on tires, suspension setup and vehicle speed
- Different reasons for understeer (e.g. front-rear tire balance, approaching the limits of the tire, …)

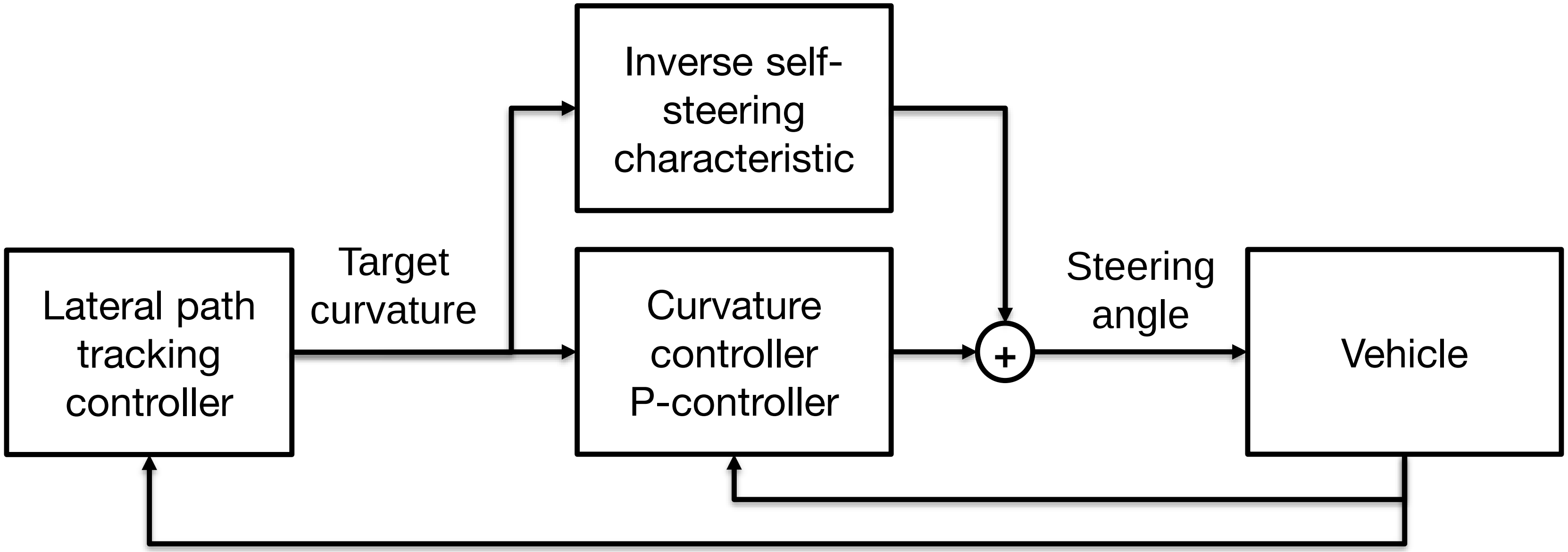
Advanced motion control – III





While operating in the **stable driving region** it is sufficient to know about the steady-state self-steering characteristics to obtain a reasonable good feedforward control law!

Advanced motion control – IV



- Nonlinear feedforward controller based on self-steering characteristic
- Curvature feedback can be done using a simple P-controller

Application to autonomous racing @ Roborace 2019

- ✓ Successful application at Roborace competition 2019
- ✓ Close to human driver performance on qualifying lap
- ✓ Moderate implementation efforts required
- ✗ Depends on curvature and acceleration signal quality
- ✗ Difficulties with dynamic trajectory planning

Source: https://www.youtube.com/watch?v=-vqQBuTQhQw

More details in:
Minimum curvature trajectory planning and control for an autonomous race car
(https://doi.org/10.1080/00423114.2019.1631455)

A software architecture for the dynamic path planning of an autonomous racecar at the limits of handling
(https://doi.org/10.1109/ICCVE45908.2019.8965238)

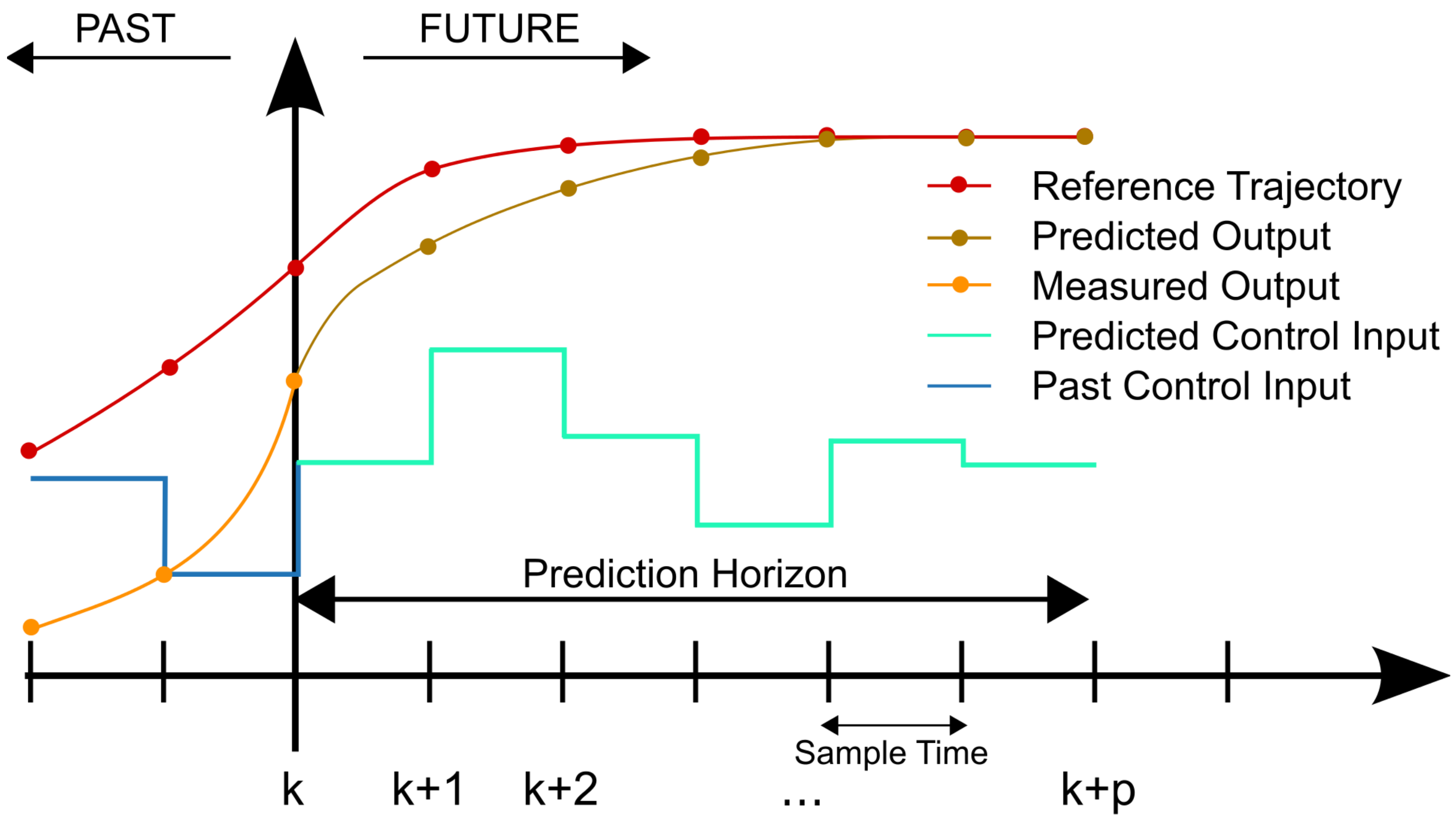
Autonomous Driving Software Engineering
Prof. Dr. Markus Lienkamp

Simon Sagmeister, M. Sc.

Agenda

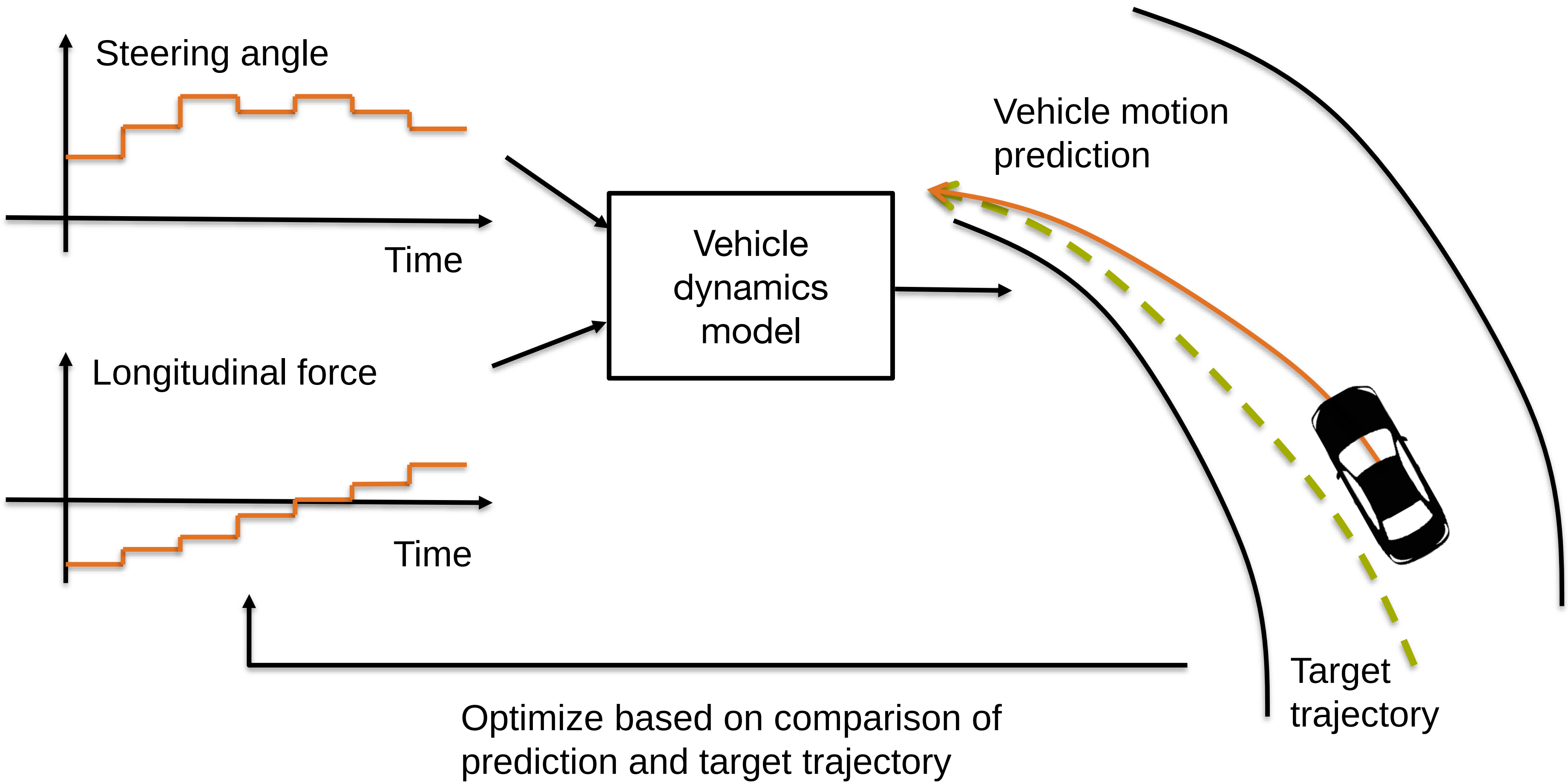
1. Introduction
2. Basic control (Geometric & PID)
3. Model-based control
4. **Model predictive control**
5. Summary

Basics of model predictive control – I



Source: https://de.wikipedia.org/wiki/Model_Predictive_Control

Basics of model predictive control – II



Basics of model predictive control – III

optimization problem

- Solution to this optimization problem is computational expensive
- Typical problem sizes are around 20-50 discretization steps

Quadratic Programming

- Quadratic cost function, linear constraints
- Works well for linear dynamics
- Efficient and reliable solvers available with <10ms computation time

Nonlinear Programming

- Arbitrary cost functions and constraints possible
- Nonlinear dynamics
- Solvers are based on linearization schemes
- Solution times range from a few milliseconds to multiple seconds

Basics of model predictive control – IV

- ✓ Considers constraints and is suited for multi-input and multi-output systems
- ✓ Nonlinear dynamics and complex cost functions
- ✗ Implementation and solution of numerical problem is a complicated challenge
- ✗ Computation times get large for many active constraints or nonlinear dynamics
- ✗ Solver might not converge → backup strategy needed

Autonomous Driving Software Engineering
Prof. Dr. Markus Lienkamp

Simon Sagmeister, M. Sc.

Agenda

1. Introduction
2. Basic control (Geometric & PID)
3. Model-based control
4. Model predictive control
5. **Summary**



Summary – What did we learn today

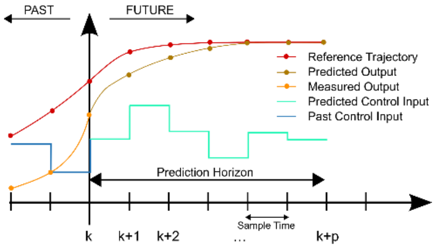
- Responsibilities of the controller within the autonomous driving stack
- Recap of control theory and vehicle dynamics modeling

Performance

Stanley &
Velocity PI

Model-based feedforward
and feedback control

Model predictive control



Complexity

Helpful literature for this lecture

Required control engineering basics:

- Lecture „Regelungstechnik“ – Prof. Dr.-Ing. Boris Lohmann – Chapter 6 & 7 (Reglerentwurf & Erweiterte Regelungsstrukturen und Zustandsregelung)

Further control engineering materials:

- Otto Föllinger, *Regelungstechnik*, VDE Verlag GmbH
- Heinz Unbehauen, *Regelungstechnik I*, Springer Vieweg
- John Doyle, Bruce Francis, Allen Tannenbaum, *Feedback Control Theory*, Macmillan Publishing Co.

Further vehicle dynamics materials:

- Lecture „Dynamik der Straßenfahrzeuge“ – Prof. Dr.-Ing. Markus Lienkamp
- William F. Milliken and Douglas L. Milliken – *Race Car Vehicle Dynamics*

Related literature

A list of papers for the controllers presented within the literature and more interesting concepts:

Brian Paden, Michal Cap, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli, *A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles*, 2016

Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun, *Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing,* 2009

Moritz Werling, *Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien,* 2011

Nitin R. Kapania and J. Christian Gerdes, *Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling*, 2015

Alexander Heilmeier, Alexander Wischnewski, Leonhard Hermansdorfer, Johannes Betz, Markus Lienkamp, and Boris Lohmann, *Minimum curvature trajectory planning and control for an autonomous race car*, 2019

Alexander Liniger, Alexander Domahidi, and Manfred Morari, *Optimization-based autonomous racing of 1: 43 scale RC cars*, 2015