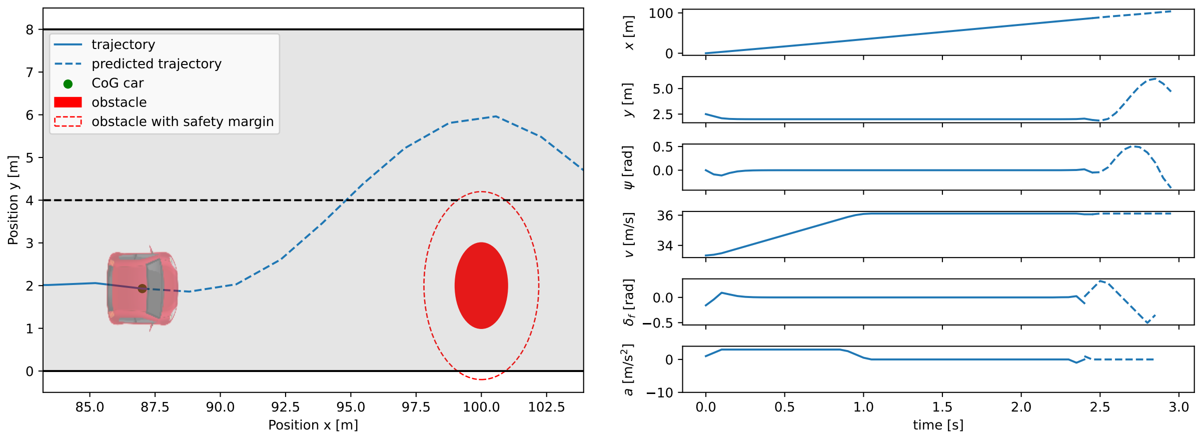
P1: Self-driving obstacle avoidance on the highway using MPC

Self-driving cars are becoming more present on our streets every day. MPC is applied in autonomous cars for satisfying all the demands a self-driving car has to fulfill. Since autonomous driving is still primarily used in less complex environments such as highways, we will design a MPC controller that is able to drive on a highway and avoid obstacles safely.



(a) Highway (b) states and inputs

Figure 1: Highway obstacle avoidance

System description

The system used to represent the car is the kinematic bicycle model, as presented in [1].

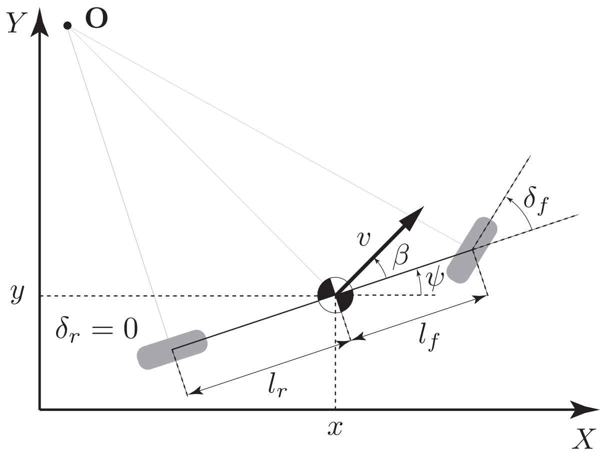


Figure 2: Kinematic bicycle model

This model is kinematic, therefore neglecting inertias, masses and forces presented to the car and the road, but considers a length of the car $l_{car} = l_r + l_f$, where l_r is the distance of the rear axis to the center of gravity (CoG), and l_f is the distance between the CoG and the front axis. Notice that in general the CoG depends on the load distribution of the vehicle, but we assume for simplicity a constant CoG in the middle of the car ($l_r = l_f$). With the above schematic the following nonlinear model can be derived:

$$\begin{aligned}
\dot{x} &= v\cos(\psi+\beta) \\
\dot{y} &= v\sin(\psi+\beta) \\
\dot{\psi} &= \frac{v}{l_r}\sin(\beta) \\
\dot{v} &= a \\
\beta &= \arctan\left(\frac{l_r}{l_f+l_r}\tan\left(\delta_f\right)\right)
\end{aligned}$$

Here, x and y are simply the coordinates of the vehicle in an inertial frame. The velocity v is the current norm of the velocity vector presented in the picture. We assume that we can manipulate our car by breaking and accelerating our vehicle (acceleration a) and steering it using the front axis angle δ_f.

The slip side angle β is the angle between the velocity vector of the car and its current heading and is in a kinematic model directly dependent on the steering angle δ_f. The current heading in the inertial system can be described with the yaw angle ψ. Full state feedback of the system is assumed.

In the modeled situation the car is driving on the right lane of a two-line highway. It is initially not positioned in the center of the right lane, which is the preferred position. In a distance of 100 m, a non-moving obstacle is placed. This obstacle hast o be avoided with a safety margin obs_Δ. The initial state of the system is $\boldsymbol{x_0} = [x_0, y_0, \psi_0, v_0] = [0\,\text{m}, 2.5\,\text{m}, 0\,\pi, 120$ km/h] and the initial inputs are $\boldsymbol{u_0} = [a_0, \delta_{f,0}] = [0, 0]$. The model parameters of the car and the environment are given in Table 1 and the bounds on the states and inputs are given in Table 2.

Table 1: Model parameters.

Parameter	Description	Value
l_{car}	length of car	3 m
w_{car}	width of car	2 m
w_{lane}	lane width	4 m
obs_x	x-position of obstacle	100 m
obs_y	y-position of obstacle	2 m
obs_r	radius of obstacle	1 m
obs_Δ	safety margin	1.2 m

Table 2: State and input bounds.

Parameter	Description	Value
ψ_{min}	maximal heading angle	$-\pi/2$
ψ_{max}	maximal heading angle	$\pi/2$
v_{max}	maximal velocity	130 km/h
a_{min}	maximal deceleration	$-10\,\text{m/s}^2$
a_{max}	maximal acceleration	$3\,\text{m/s}^2$
$\delta_{f,min}$	minimal steering angle	$-\pi/2$
$\delta_{f,max}$	maximal steering angle	$\pi/2$

Mandatory Tasks

Important: Please use only the Python packages that have been referenced in this task description. Through the entire project you are allowed to use: **CasADi**, **NumPy**, and **Matplotlib**. We have referenced additional packages that are allowed in specific subtasks. If you want to use a different package, please contact your supervisor.

The following tasks **have to be completed** in order to pass the project.

- Implement the system model in Python using CasADi. Discretize the system via orthogonal collocation on finite elements with $\Delta\, t = 0.05$ s and simulate it for $N_{sim} = 100$ timesteps. Apply for the first half of the simulation the constant input $\boldsymbol{u_1} = [0.1, 5 \cdot 10^{-4}\pi]$ and for the second half $\boldsymbol{u_2} = [-0.1, -5 \cdot 10^{-4}\pi]$. Explain shortly the system behavior.
- Implement an MPC controller with a prediction horizon $N = 3$. The following sub-tasks of increasing complexity shall be considered by the MPC controller. Discuss for each subtask shortly how you enforced the respective behavior and what changes you observed in the trajectories. **Hint:** Neglect the dimension of the car for subtasks i)-iii) and model the car as a point source.

i) Minimize traveling time
ii) Keep the car in general at the center of the right lane
iii) Since in the kinematic model all inertias are neglected, limit the input change rates $\Delta \boldsymbol{u} = |\, \boldsymbol{u}_i - \boldsymbol{u}_{i-1} \,|$ to $\Delta \boldsymbol{u}_{max} = [1, \pi/20]$ as a hard constraint in order to ensure a safe and comfortable ride
iv) Avoid the obstacle without interfering with the safety margin or leaving the highway
   - How can you over-approximate the shape of the car? Adapt the state bounds and implement the obstacle avoidance.
   - Is the prediction horizon of $N = 3$ suitable? State the minimal prediction horizon N for successfull obstacle avoidance. Which behavior could be improved by a further increase of N?

Additional tasks

Below are **suggested** additional tasks to obtain an excellent grade for the project. We want to emphasize that students are encouraged to come up with their own ideas for additional investigations and not all of the suggestions below must be included for an excellent grade.

- Avoid multiple obstacles
- How could you further promote a smooth drive?
- Think about different shapes for representing the car and the respective advantages and disadvantages
- Is the MPC with your proposed N applicable for real-time performance?

Deliverables

The following materials have to be submitted in order to pass the semester project:

- **Recorded final presentation** (video screencast). The presentation must be **5-7 minutes** (for the entire group) and the file should not exceed **200 mb**. Highlight on the slides which group member(s) are responsible.
- **Written report** to present and discuss the obtained results. You must use the **supplied template on Moodle** (Tex or Word) and write no more than **3-4 pages** (for the entire group). Highlight which group member worked on which section.
- **Source code** of your project. Please ensure that the code is executable and optionally add a short explanation of the structure (readme).

Please ensure that all formal conditions are (e.g. page limits, highlight responsible author) are satisfied, as we will deduct points for significant violations. Deadline for the submission is 18.07.2025. Please submit all deliverables via moodle.

Supervisor

Please address questions to:

Name	Contact
Tobias Brockhoff	tobias.brockhoff@tu-dortmund.de

References

[1] Jason Kong et al. “Kinematic and dynamic vehicle models for autonomous driving control design”. In: (2015), pp. 1094–1099. DOI: 10.1109/IVS.2015.7225830.