Autonomous Vision Group
Prof. Dr.-Ing. Andreas Geiger

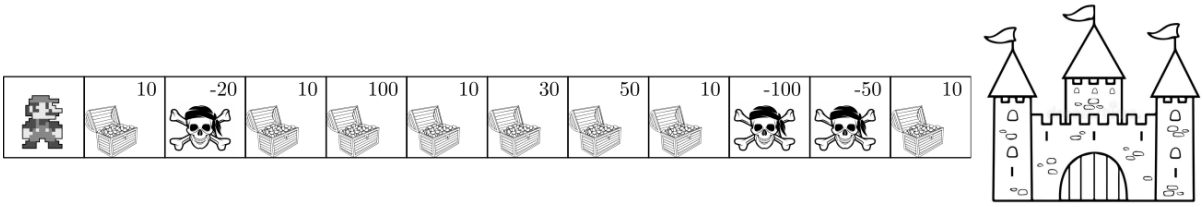
Eberhard Karls
Universität
Tübingen

Self-Driving Cars

Ex. 4 – Pen and Paper (Lecture 04 + 05 + 06)

4.1 Super Mario

There is Mario who wants to meet princess Peach. Mario's home and the palace of Peach are located on a 1D discrete coordinate system and the palace is always on the right side of Mario's home. Mario starts his journey from his home and he can move **two or three** steps to the right. At each discrete coordinate, there are several coins that Mario can collect or a thief who robs a portion of Mario's coins. Mario wants to collect many coins as Mario wants to buy a new dress for Peach. Help Mario to collect as many coins as possible during his journey.



Mario knows where he is located. The state can be defined as $\mathbf{s}_t = x_t$ where x_t is a location at time t. Note that he holds 100 coins at the beginning. Mario will enter the palace by moving over the last square.

a) Mario is greedy but fool so he always moves to a square where he makes more profit (or less loss). Define the policy $a = \pi(x)$ of greedy but fool Mario.

$\pi(0)$ = Jump 3 squares
$\pi(1)$ = Jump 3 squares
$\pi(2)$ = Jump 2 squares
$\pi(3)$ = Jump 3 squares
$\pi(4)$ = Jump 3 squares
$\pi(5)$ = Jump 2 squares
$\pi(6)$ = Jump 2 squares
$\pi(7)$ = Jump 3 squares
$\pi(8)$ = Jump 3 squares
$\pi(9)$ = Jump 2 squares
$\pi(10)$ = Jump 2 or 3 squares
$\pi(11)$ = Jump 3 or 3 squares

b) Roll out his trajectory plus his coin status until reaching the palace.

t	location	coins
0	0	100

t	location	coins
0	0	100
1	3	110
2	6	140
3	8	150
4	11	160
5	castle	160

c) Calculate the state-value function $V_\pi(x)$ and the action-value function $Q_\pi(x, a_t)$ of the greedy policy π. The discount factor γ is 1.

	0	1	2	3	4	5	6	7	8	9	10	11
V_π	60	100	100	50	0	0	20	-50	10	10	0	0
$Q_\pi(x, \text{jump 2 squares})$	80	60	100	10	50	0	20	-90	-50	10	0	0
$Q_\pi(x, \text{jump 3 squares})$	60	100	10	50	0	20	-90	-50	10	0	0	0

d) Find an optimal policy π via Q-learning. Let $Q_k(x, a_t)$ be the k-th updated $Q(x, a_t)$ function by the Bellman equality equation. To speed-up the calculations, we make two changes to the original Q-learning algorithm: Instead of random initialization, **initialize** $Q_0(x, a_t)$ to the **greedy** policy π. Furthermore, during each iteration, update all elements of the Q table **simultaneously** based on the TD error update rule, instead of executing a behavior policy with ϵ-greedy exploration and updating only a single entry. Compute $Q_k(x, a_t)$ until the induced policy becomes the optimal policy.

$J = \{j_0, j_1\} = \{\text{jump 2 squares}, \text{jump 3 squares}\}$

$Q_{k+1}(x, a_t) = r(x + a_t) + \max_{a \in J}(Q_k(x + a_t, a))$

	0	1	2	3	4	5	6	7	8	9	10	11
$r(x)$	0	10	-20	10	100	10	30	50	10	-100	-50	10
$Q_0(x, j_0)$	80	60	100	10	50	0	20	-90	-50	10	0	0
$Q_0(x, j_1)$	60	100	10	50	0	20	-90	-50	10	0	0	0
$\max_{a \in J}(Q_0(x, a))$	80	100	100	50	50	20	20	-50	10	10	0	0

	0	1	2	3	4	5	6	7	8	9	10	11
$r(x)$	0	10	-20	10	100	10	30	50	10	-100	-50	10
$Q_1(x, j_0)$	80	60	150	30	50	0	20	-90	-50	10	0	0
$Q_1(x, j_1)$	60	150	30	50	0	20	-90	-50	10	0	0	0
$\max_{a \in J}(Q_1(x, a))$	80	150	150	50	50	20	20	-50	10	10	0	0

	0	1	2	3	4	5	6	7	8	9	10	11
$r(x)$	0	10	-20	10	100	10	30	50	10	-100	-50	10
$Q_2(x, j_0)$	130	60	150	30	50	0	20	-90	-50	10	0	0
$Q_2(x, j_1)$	60	150	30	50	0	20	-90	-50	10	0	0	0
$\max_{a \in J}(Q_2(x, a))$	130	150	150	50	50	20	20	-50	10	10	0	0

	0	1	2	3	4	5	6	7	8	9	10	11
$r(x)$	0	10	-20	10	100	10	30	50	10	-100	-50	10
$Q_3(x, j_0)$	130	60	150	30	50	0	20	-90	-50	10	0	0
$Q_3(x, j_1)$	60	150	30	50	0	20	-90	-50	10	0	0	0
$\max_{a \in J}(Q_3(x, a))$	130	150	150	50	50	20	20	-50	10	10	0	0

No change after Q_2.

e) Induce the optimal policy from the optimal state-value function Q^*.

$\pi(0)$ = Jump 2 squares
$\pi(1)$ = Jump 3 squares
$\pi(2)$ = Jump 2 squares
$\pi(3)$ = Jump 3 squares
$\pi(4)$ = Jump 2 squares
$\pi(5)$ = Jump 3 squares
$\pi(6)$ = Jump 2 squares
$\pi(7)$ = Jump 3 squares
$\pi(8)$ = Jump 3 squares
$\pi(9)$ = Jump 2 squares
$\pi(10)$ = Jump 2 squares
$\pi(11)$ = Jump 2 squares

4.2 Holonomic and Non-Holonomic Constraints

a) What are the definitions of holonomic and non-holonomic constraints?

Holonomic constraints are constraints on the configuration.

Non-holonomic constraints are constraints on the velocity.

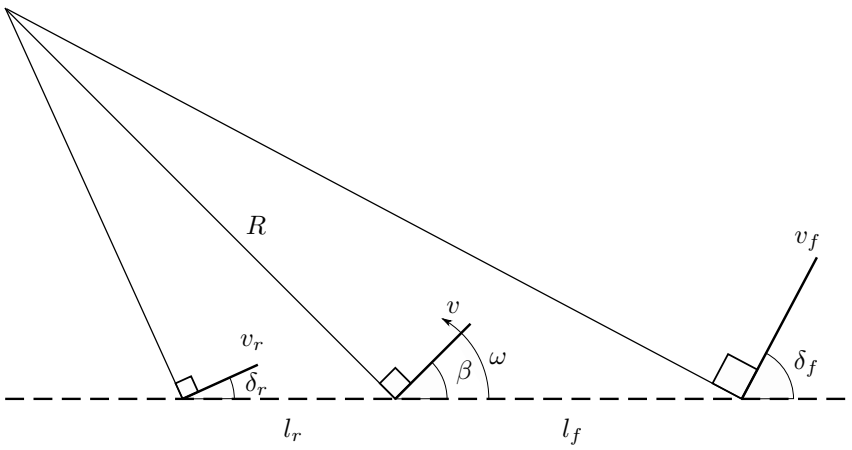
b) Specify examples for holonomic and non-holonomic systems others than those provided in the lecture.

Holonomic system - a gantry crane (2D plane), a forklane (3D), a drop tower (1D), an elevator (1D), etc.

Non-holonomic system - Segway, a plane, ski, etc.

4.3 The Bicycle Model

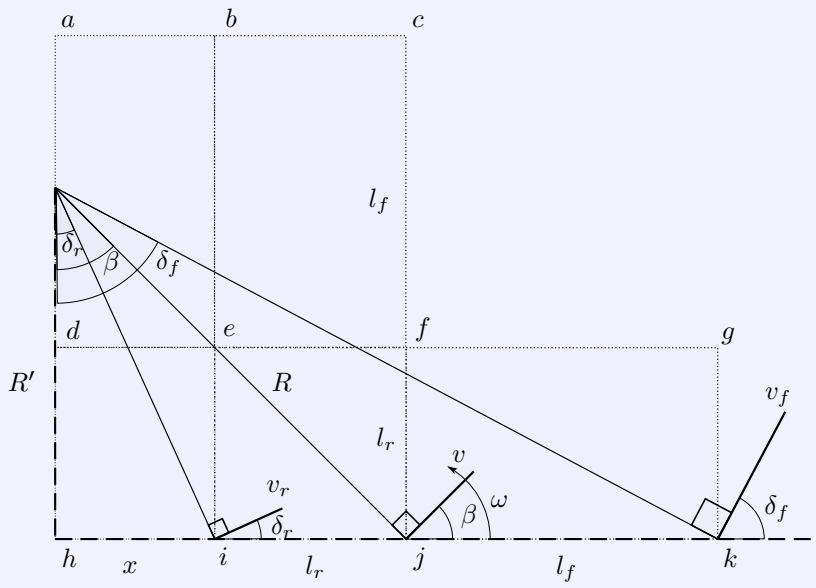
In Lecture 5, we derive equations of the bicycle model with a fixed rear wheel. We can formulate those equations also for a steering rear wheel. The following figure shows the graphical overview of the bicycle model.



a) Prove that angular velocity ω at the gravity center of the car is $\frac{v\cos(\beta)}{l_f+l_r}(\tan(\delta_f)-\tan(\delta_r))$.

$$
\begin{aligned}
\tan(\delta_r) &= \frac{x}{R'} \\
\tan(\beta) &= \frac{x+l_r}{R'} \\
\tan(\delta_f) &= \frac{x+l_r+l_f}{R'} \\
\tan(\delta_f)-\tan(\delta_r) &= \frac{l_r+l_f}{R'} \\
\frac{1}{R'} &= \frac{\tan(\delta_f)-\tan(\delta_r)}{l_r+l_f} \\
R &= R'/\cos\beta \\
\omega &= \frac{v}{R} = \frac{v\cos\beta}{R'} \\
&= \frac{v\cos\beta}{l_r+l_f}(\tan(\delta_f)-\tan(\delta_r))
\end{aligned}
$$

b) Prove $\beta = tan^{-1}\left(\frac{l_f\tan(\delta_r)+lr\tan(\delta_f)}{l_f+l_r}\right)$.



Way 1.

$$
\begin{aligned}
\tan(\delta_r) &= \frac{x}{R'} \\
\tan(\beta) &= \frac{x + l_r}{R'} \\
\tan(\delta_f) &= \frac{x + l_r + l_f}{R'} \\
\tan(\delta_f) - \tan(\delta_r) &= \frac{l_r + l_f}{R'} \\
\frac{1}{R'} &= \frac{\tan(\delta_f) - \tan(\delta_r)}{l_r + l_f} \\
\tan(\beta) &= \frac{x + l_r}{R'} = \frac{x}{R'} + \frac{l_r}{R'} \\
&= \tan\delta_r + l_r \frac{\tan(\delta_f) - \tan(\delta_r)}{l_r + l_f} \\
&= \frac{l_r \tan(\delta_f) - l_r \tan(\delta_r) + (l_r + l_f)\tan\delta_r}{l_r + l_f} \\
&= \frac{l_r \tan(\delta_f) + l_f \tan(\delta_r)}{l_r + l_f}
\end{aligned}
$$

Way 2.

$$
\begin{aligned}
\mathrm{Rect}(a, c, j, h) &= (l_r + l_f) R' \tan\beta \\
\mathrm{Rect}(a, b, e, d) &= l_f R' \tan\delta_r \\
\mathrm{Rect}(d, g, k, h) &= l_r R' \tan\delta_f \\
(l_r + l_f) R' \tan\beta &= l_f R' \tan\delta_r + l_r R' \tan\delta_f \\
\tan\beta &= \frac{l_f \tan\delta_r + l_r \tan\delta_f}{l_r + l_f} \\
\beta &= \tan^{-1}\left(\frac{l_f \tan\delta_r + l_r \tan\delta_f}{l_r + l_f}\right)
\end{aligned}
$$

4.4 A Bread Toaster

Andreas has a one-button bread toaster that heats a bread for a certain amount of time. The toaster makes perfect toasts so he takes the toaster on travels to enjoy every time when he goes abroad to enjoy the perfect toast. One day, he went to Egypt to attend a seminar called "The Future of Self-Driving Cars" and made a toast with his reliable machine at the morning. However, the bread was burned and Andreas had a bad start into the day.

a) What could cause the problem?

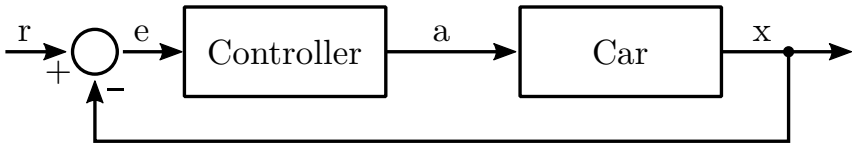
> The air temperture could be higher than Germany, the electric power is higher than Germany or the bread in Egypt is apt to be burned.

b) How can we help Andreas sustain his successful morning start? How could we improve the toaster in terms of control?

> Make a closed-loop controlled toaster.
>
> **i)** The toaster measures the color of the bread. If it is dark, pop the toast.
>
> **ii)** The toaster measures the temperature of the bread. If it is higher than the threshold, stop heating the air (not pop the toast until the heating time ends).

4.5 PID Control

Consider the problem of longitudinal vehicle control. Let $x \in \mathbb{R}$ denote the vehicle's longitudinal position, $v \in \mathbb{R}$ its velocity and $a \in \mathbb{R}$ its acceleration. Let the initial values be $x_0 = 2$, $v_0 = 0$ and $a_0 = 0$. Consider a closed-loop controller with set point $r = 0$ as illustrated in the following figure where a is the control variable and x is the control state. The goal of the controller is to minimize error $e = r - x$.



a) Write down control laws $a = f_{\{\text{Bang,P,PD}\}}(e)$ for a Bang-Bang controller, a proportional controller (P control) and a proportional differential controller (PD control), replacing $f(\cdot)$ with the correct functions assuming gain $K = 1$ for both coefficients (for a Bang-Bang controller, the magnitude of $f(\cdot)$ is 1). Also write the control law in terms of the position x for each controller.

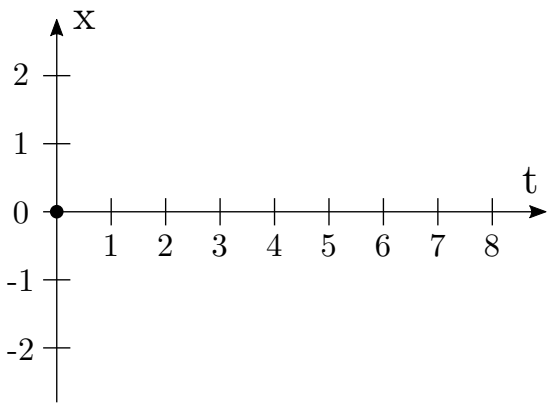
$$a = f_{\text{Bang}}(e) = \begin{cases} 1 & \text{if } 0 > x, \\ -1 & otherwise. \end{cases}$$

$$a = f_{\text{P}}(e) = Ke = -x$$

$$a = f_{\text{PD}}(e) = Ke + K\frac{de}{dt} = -x - \frac{dx}{dt}$$

b) Execute each controller for 8 time steps by completing the table and plot below.

t	a	v	x
0	0	0	2
1			
2			
3			
4			
5			
6			
7			
8			



Bang-Bang Control

t	a	v	x
0	0	0	2
1	-1	-1	1
2	-1	-2	-1
3	1	-1	-2
4	1	0	-2
5	1	1	-1
6	1	2	1
7	-1	1	2
8	-1	0	2

Proportional Control

t	a	v	x
0	0	0	2
1	-2	-2	0
2	0	-2	-2
3	2	0	-2
4	2	2	0
5	0	2	2
6	-2	0	2
7	-2	-2	0
8	0	-2	-2

Proportional Derivative Control

t	a	v	x
0	0	0	2
1	-2	-2	0
2	2	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0

c) Comment on the output of each controller. What happens? Which controller do you desire in terms of the behavior?

A Bang-Bang controller and a proportional controller suffer from the oscillation.

A proportional derivative control doesn't lead to the oscillation.

4.6 Pure Pursuit Control

A pure pursuit controller controls a car to drive towards the target point. Suppose the rotation center is at $(0, 10)$ and the back wheel of a car is at $(0, 0)$ and aligned with the x axis (unit is meter). The car is moving towards the target point $(10, 10)$ using a pure pursuit controller. Using the following equations, we can compute the next position of the car and the angle between the car axis and the x-axis.

$$
\begin{aligned}
X_{t+1} &= X_t + v\cos(\psi_t)\Delta t \\
Y_{t+1} &= Y_t + v\sin(\psi_t)\Delta t \\
\psi_{t+1} &= \psi_t + \frac{v\delta_t}{l_f + l_r}\Delta t \\
\delta_t &\approx \frac{2L}{d_t^2} e_t
\end{aligned}
$$

Consider a timestep of $\Delta t = 1$ second and a velocity $v = 1$ meter/second. The length of the car is 2 meter and the center of gravity of the car is its middle point. Distance d_t is the distance between the target point and the postion of the rear wheel.

a) Roll out the state of the car including the location, the angle between the car and the x-axis for 8 time steps.

t	x	y	ψ
0	1.000	0	0
1	2.000	0	5.65 deg
2	2.995	0.099	11.32 deg
3	3.976	0.295	17.01 deg
4	4.932	0.587	22.72 deg
5	5.854	0.974	28.49 deg
6	6.733	1.451	34.31 deg
7	7.559	2.014	40.22 deg
8	8.323	2.660	46.24 deg

b) Draw the trajectory of the center of the gravity of the car on the xy-coordinate system. How does the car behave? How does the trajectory look like?

The blue dots represent the trajectory of the center of gravity of the car.

The car turns less than the spherical trajectory (orange dots).