

Sheet 2

Topic: Linear Algebra, Locomotion, and Sensing

Submission deadline: May 07, 2015

Submit to: mobilerobotics@informatik.uni-freiburg.de

Transformations between coordinate frames play an important role in robotics. As background for exercises 1 and 3 on this sheet, please refer to the linear algebra slides on affine transformations and transformation combination.

Exercise 1: 2D Transformations as Affine Matrices

The 2D pose of a robot w.r.t. a global coordinate frame is commonly written as $\mathbf{x} = (x, y, \theta)^T$, where (x, y) denotes its position in the xy -plane and θ its orientation. The homogeneous transformation matrix that represents a pose $\mathbf{x} = (x, y, \theta)^T$ w.r.t. to the origin $(0, 0, 0)^T$ of the global coordinate system is given by

$$T = \begin{pmatrix} \mathbf{R}(\theta) & \mathbf{t} \\ 0 & 1 \end{pmatrix}, \mathbf{R}(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}, \mathbf{t} = \begin{pmatrix} x \\ y \end{pmatrix}$$

- While being at pose $\mathbf{x}_1 = (x_1, y_1, \theta_1)^T$, the robot senses a landmark l at position (l_x, l_y) w.r.t. to its local frame. Use the matrix T_1 to calculate the coordinates of l w.r.t. the global frame.
- Now imagine that you are given the landmark's coordinates w.r.t. to the global frame. How can you calculate the coordinates that the robot will sense in his local frame?
- The robot moves to a new pose $\mathbf{x}_2 = (x_2, y_2, \theta_2)^T$ w.r.t. the global frame. Find the transformation matrix T_{12} that represents the new pose w.r.t. to \mathbf{x}_1 . *Hint:* Write T_{12} as a product of homogeneous transformation matrices.
- Again, the robot measures the position of the landmark l . What will be the result, given the coordinates (l_x, l_y) w.r.t. to \mathbf{x}_1 ?

Exercise 2: Locomotion

A robot equipped with a differential drive starts at position $x = 1.0m$, $y = 2.0m$ and with heading $\theta = \frac{\pi}{2}$. It has to move to the position $x = 1.5m$, $y = 2.0m$, $\theta = \frac{\pi}{2}$ (all angles in radians). The movement of the vehicle is described by steering commands (v_l = speed of left wheel, v_r = speed of right wheel, t = driving time).

- (a) What is the minimal number of steering commands (v_l, v_r, t) needed to guide the vehicle to the desired target location?
- (b) What is the length of the shortest trajectory under this constraint?
- (c) Which sequence of steering commands guides the robot on the shortest trajectory to the desired location if an arbitrary number of steering commands can be used?
- (d) What is the length of this trajectory?

Note: the length of a trajectory refers to the travelled distance along the trajectory.

Exercise 3: Sensing

A robot is located at $x = 1.0m$, $y = 0.5m$, $\theta = \frac{\pi}{4}$. Its laser range finder is mounted on the robot at $x = 0.2m$, $y = 0.0m$, $\theta = \pi$ (with respect to the robot's frame of reference).

The distance measurements of one laser scan can be found in the file `laserscan.dat`, which is provided on the website of this lecture. The first distance measurement is taken in the angle $\alpha = -\frac{\pi}{2}$ (in the frame of reference of the laser range finder), the last distance measurement has $\alpha = \frac{\pi}{2}$ (i.e., the field of view of the sensor is π), and all neighboring measurements are in equal angular distance (all angles in radians).

Note: You can load the data file and calculate the corresponding angles in Python using

```
import math
import numpy as np
scan = np.loadtxt('laserscan.dat')
angle = np.linspace(-math.pi/2, math.pi/2, np.shape(scan)[0], endpoint='true')
```

- (a) Use Python to plot all laser end-points in the frame of reference of the laser range finder.
- (b) The provided scan exhibits an unexpected property. Identify it and suggest an explanation.
- (c) Use homogeneous transformation matrices in Python to compute and plot the center of the robot, the center of the laser range finder, and all laser end-points in world coordinates.

Note: You can equally scale the x and y -axis of a plot using

```
plt.gca().set_aspect('equal', adjustable='box')
```