Albert-Ludwigs-Universität Freiburg
Lecture: Introduction to Mobile Robotics
Summer term 2019

Institut für Informatik

Prof. Dr. Wolfram Burgard
Dr. Daniel Büscher
Lukas Luft
Marina Kollmitz
Iman Nematollahi

mobilerobotics@informatik.uni-freiburg.de

Sheet 1

Topic: Linear Algebra

Due date: 03.05.2019

Exercise 1: Linear Algebra

(a) Consider the matrices

$$\mathbf{A} = \begin{pmatrix} 0.25 & 0.1 \\ 0.2 & 0.5 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0.25 & -0.3 \\ -0.3 & 0.5 \end{pmatrix}.$$

Are they symmetric positive definite?

(b) For

$$\mathbf{C} = \begin{pmatrix} -3 & 0 \\ 0 & 1 \end{pmatrix},$$

find the largest value for $\mu \in \mathbb{R}$ for which $C + \mu I$ is not symmetric positive definite.

(c) Write a program in Python that determines whether a matrix is orthogonal.

(d) Use this program to investigate whether

$$\mathbf{D} = \frac{1}{3}\begin{pmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 2 & 2 \end{pmatrix}$$

is orthogonal.

Exercise 2: 2D Transformations as Affine Matrices

Transformations between coordinate frames play an important role in robotics. As background for exercises 2 and 3 on this sheet, please refer to the linear algebra slides on affine transformations and transformation combination.

Considering a robot moving on a plane, its pose 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

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

(a) While being at pose $\mathbf{x_1} = (x_1, y_1, \theta_1)^T$, the robot senses a landmark l at position $\mathbf{l} = (l_x, l_y)$ w.r.t. to its local frame. Use the matrix T_1 to calculate the coordinates of $\mathbf{l}$ w.r.t. the global frame.

(b) 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 its local frame?

(c) 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.

(d) The robot is at position $\mathbf{x_2}$. Where is the landmark $\mathbf{l} = (l_x, l_y)$ w.r.t. the robot's local frame now?

Exercise 3: Sensing

A robot is located at $x = 1.0\,\text{m}$, $y = 0.5\,\text{m}$, $\theta = \frac{\pi}{4}$. Its laser range finder is mounted on the robot at $x = 0.2\,\text{m}$, $y = 0.0\,\text{m}$, $\theta = \pi$ 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 an 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')
```