



## Introduction to Robotics

### Assignment #4

Due: 30.06.2019, 23:59

Preferred tools/languages for programming, plotting and solving are

python, matlab, octave or gnuplot.

Always provide source code as well as all intermediate steps. Make sure to annotate your code in detail in order to provide readability and fast understanding. "Here, some magic happens" will cost you points.

**Task 4.1 (4 points) Basis-Splines – Non-uniform Visualization:** Draw the basis spline and the resulting spline for a B-Spline of  $4^{th}$  order for the knot vector  $[0 \ .3 \ .4 \ .8 \ 1]$  (time intervals). Provide functions and plot within your solution as well as the attached source code.

**Task 4.2 (4 points) Lagrange polynomial:** Given a set of data points  $(2, 1), (5, -4), (10, -2), (11, -3)$ . Calculate the Lagrange polynomial  $P_3(x)$  through the data points.

Simplify the equation as much as possible. Use a tool of your choice (gnuplot is preferred) to visualize the four Lagrange basis polynomials and the polynomial  $p_3(x)$ .

**Task 4.3 (6 points) PID-controller:** Consider a DC motor with

**J** moment of inertia of the rotor =  $0.1 \text{ kg m}^2$

**b** motor viscous friction constant =  $0.1 \text{ N m s}$

**K** motor torque constant =  $0.01 \text{ N m/Amp}$

**R** electric resistance =  $1 \text{ Ohm}$

**L** electric inductance =  $0.5 \text{ H}$

**4.3.1 (4 points):** Calculate the response of a PID-controller with the factors  $k_p$ ,  $k_i$  and  $k_d$  for the DC motor<sup>1</sup> with a 1-rad/sec step reference and plot the function for both subtasks with

1.  $k_p = 20, k_i = 0, k_d = 0,$
2.  $k_p = 500, k_i = 0, k_d = 0$  and
3.  $k_p = 100, k_i = 50, k_d = 0$

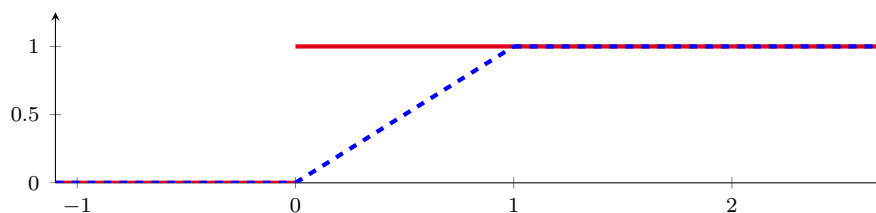
using a tool of your choice (matlab should be preferred).

to the step function:

$$y(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$

the ramp function

$$y(t) = \begin{cases} 0, & t < 0 \\ t, & 0 \leq t < 1 \\ 1, & t \geq 1 \end{cases}$$



**4.3.2 (2 points):** Explain the impact of  $k_p$  and  $k_i$  and their combination. What will be improved if  $k_d \neq 0$ ?

**4.3.3 \*Bonus\* (4 points):** find  $k_p, k_i, k_d$  for the step function such that

- settling time is less than 2 seconds
- overshoot is less than 5%
- steady-state error less than 1%

<sup>1</sup>Find the transfer function of a DC motor to be  $\frac{K}{(J \cdot s + b) \cdot (L \cdot s + R) + K^2}$  and consider using the `feedback()` function in matlab

#### Task 4.4 (8 points) Configuration Space – Programming Task:

Given is the planar 2-DOF manipulator as seen in Figure 1.

- The base is mounted at position  $(500, 500)$  of the reference coordinate system.
- The manipulator is defined as
  - link length:  $l_1 = l_2 = 200$
  - joint range:  $\rho_1, \rho_2 \in [0, 2\pi)$
- The reachable workspace has a radius of 400
- The workspace holds two circular obstacles
- Obstacle  $o_1$ 
  - origin:  $(270, 620)$
  - radius: 50
- Obstacle  $o_2$ 
  - origin:  $(250, 200)$
  - radius: 200

**4.4.1 (6 points):** Use the mechanisms of configuration space to shrink the manipulator to a single point. Assume the manipulator to have no physical links and the TCP to be a circle with radius 2. Plot the configuration space with the two C-obstacles and the point-sized manipulator. Discretize the configuration space to a precision of at least  $1^\circ$ .

**4.4.2 (2 points):** Plot the start area (circle with radius 10 around  $s = (900, 500)$ ) and the two goal areas (circles with radius 10 around  $g_1 = (580, 150)$  and  $g_2 = (230, 470)$ ) in the workspace and in the configuration space. Draw a path from  $s$  to  $g_1$  and from  $s$  to  $g_2$ .

#### Task 4.5 (5 points) Arbitrary C-Obstacles – Programming Task:

Extend the program to support arbitrary polygon obstacles. Assume two new obstacles

- Obstacle  $o_3$ : a rectangle with the following corner points
  - $(650, 450), (650, 200), (800, 200), (800, 450)$
- Obstacle  $o_4$ : a polygon with the following corner points
  - $(600, 800), (550, 900), (750, 900), (700, 800), (650, 750), (600, 800)$

Plot the configuration space with C-obstacles  $o_1 - o_4$ . Draw a path from  $s$  to  $g_1$  and from  $s$  to  $g_2$ .

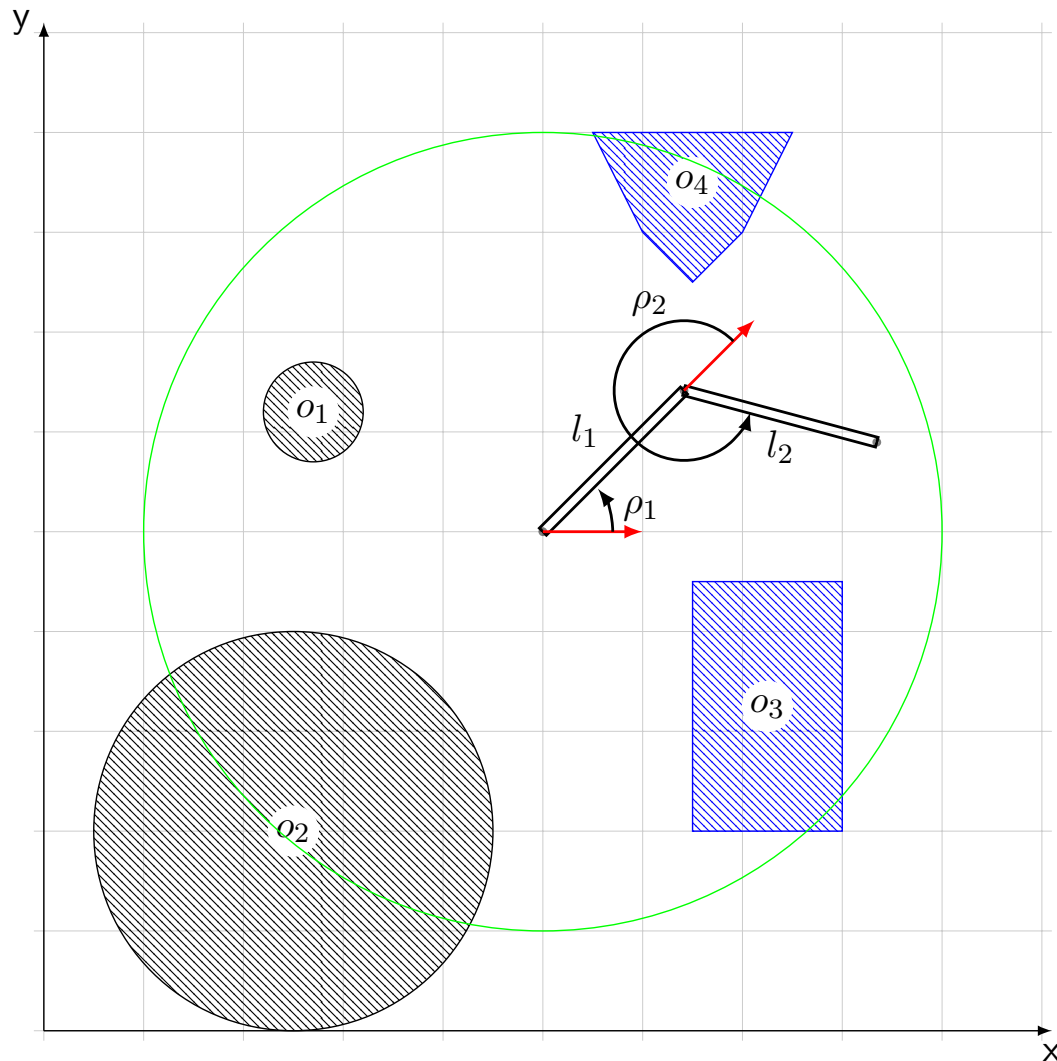


Figure 1: 2-DOF planar manipulator in workspace with obstacles. Grid distance is 100

**Pay attention to the depicted direction of  $\rho_1$  and  $\rho_2$**

The red arrows represent the zero position of the joints, the black arrows point in the positive rotation direction

Present the resulting plots as a PDF and pack your executable code (with all required libraries) as a ZIP file.

**Hints:**

When using python you might find useful:

**Shapely** library allows for polygons, circles (buffered points), affine transformations and intersections of objects <https://pypi.python.org/pypi/Shapely>

**numpy** is a powerful scientific library with fast array transformations and math functions <http://www.numpy.org/>

**math** math library grants access to sine, cosine, degree and radians <https://docs.python.org/2/library/math.html>

**PIL(low)** is the python imaging library which can export arrays to images <https://pypi.python.org/pypi/Pillow/2.2.1>

**multiprocessing** allows to speed up your code by using multiple processes <https://docs.python.org/3/library/multiprocessing.html>