The following sections describe your distant lab assignment

2 Calibration of the robot

Calibrate your robot : select appropriate commands, perform measures and solve the corresponding calibration optimization problem using `scipy.optimize.least_squares`

Perform the same tasks as in Section 1.2 to assess your calibration. To do so, you need to modify the minibex model `5R.mbx` to reflect your calibration outcome.

3 Path following

The target path in the workspace is defined as the circle centered at $(0, -20)$ with radius 5. Display it in dotted line in the robot workspace. Discretize it appropriately using the parametric formulation : $x = (0, -20) + r(\cos t, \sin t)$ with $t \in [0, 2\pi]$.

Make your robot follow this path : Use IBEX to obtain all possible commands corresponding to the initial pose in the target path ; How many solutions do you obtain ? Why ? Starting from each solution, solve the path following optimization problem using `scipy.optimize.root`. Actuate the robot and display the followed path. Repeat the process two times : using nominal and calibrated geometric parameter values. Can you comment ?

4 Path-planning

The goal is to plan a shortest distance path linking the origin pose $(0, -15)$ to the destination pose $(0, 15)$ while avoiding some obstacles defined as circles with following centers and radii (x, y, r) :

$(-12, -6, 1.5)$	$(-12, 0, 1.1)$	$(-12, 6, 1.8)$	$(-6, -18, 2.8)$	$(-6, -12, 2.9)$	$(-6, -6, 0.6)$	$(-6, 0, 1.6)$	$(-6, 6, 0.4)$	$(-6, 12, 1.9)$
$(-6, 18, 0.4)$	$(0, -24, 2.4)$	$(0, -12, 1.9)$	$(0, -6, 0.5)$	$(0, 0, 1.8)$	$(0, 6, 2.9)$	$(0, 12, 1.7)$	$(0, 24, 0.5)$	$(6, -18, 1.9)$
$(6, -12, 2.8)$	$(6, -6, 0.4)$	$(6, 0, 1.9)$	$(6, 6, 0.3)$	$(6, 12, 2.3)$	$(6, 18, 1.2)$	$(12, -6, 2.8)$	$(12, 0, 1.3)$	$(12, 6, 2.1)$

Compute this path : Use IBEX to compute a sufficiently precise, yet not too verbose, paving of the obstacle and singularity free manifold of the robot ; Build the corresponding neighborhood graph in python and identify the boxes corresponding to the origin and destination poses ; Use `scipy.sparse.csgraph.djikstra` to solve the shortest path planning optimization problem for each pair of origin/destination boxes. How many valid paths do you obtain ? Why ? Display the boxes in each of these paths, and actuate the robot with the center commands of the boxes. Again, repeat the process with nominal and with calibrated architecture parameters.

5 Work deposit

Your work must be deposited via the corresponding link on Madoc by

Saturday 23 December 2017, 12 :00 PM (noon)

It must consist of :

— A well commented python script, fully executable as a whole to solve in sequence the problems described in the previous sections (calibration, then path following, then path planning)[4]. The comments will explain the details of the process used to solve each problem and provide evidences about the solution quality ; in case a problem could not be solved, they will explain the difficulties encountered ;

— The well commented IBEX models used for the various problems solved. The comments will in particular state the options set for the solving command `ibexsolve`, justifying their appropriateness for the considered problem.

You may also include a brief PDF report in case you feel it is more appropriate to render some of the prescribed information (figures, data sets, ...)

[4]You can pause a python script by asking the user to press a key, e.g., `input("Press <ENTER> to continue...")`.