Assignment on Inverse Kinematics

Assignment

Introduction

You are provided with the **3-links planar manipulator** shown in the sketch below, whose link lengths $\mathbf{l_1}$, $\mathbf{l_2}$, $\mathbf{l_3}$ are all equal and whose tip E is identified by the triplet $[\mathbf{x_e},\ \mathbf{y_e},\ \phi_\mathbf{e}]$, being $(x_e,\ y_e)$ the Cartesian coordinates of the tip, whereas ϕ_e is the angle between the link 3 and the x-axis of the root frame.

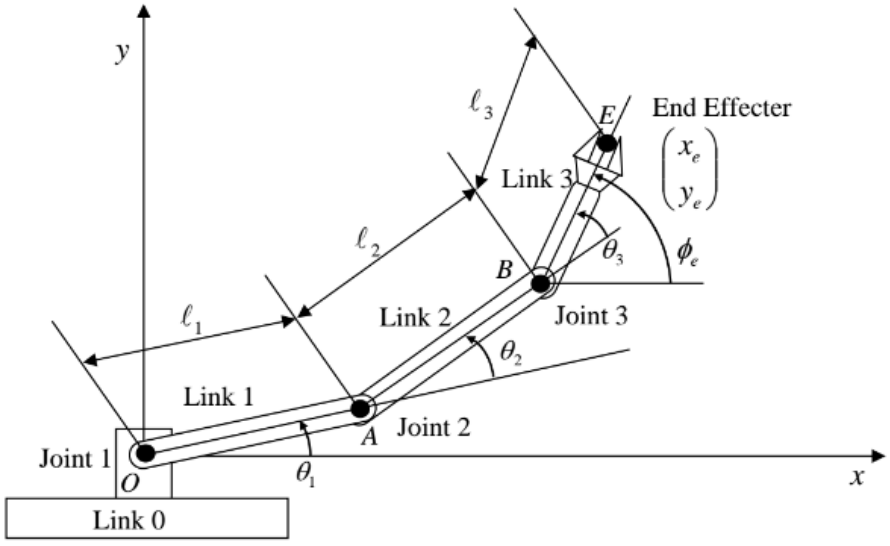

Figure 1: kinematics

Moreover, the manipulator is actuated by means of 3 motors controlled in velocity that move the joints angles $\theta_\mathbf{1}$, $\theta_\mathbf{2}$, $\theta_\mathbf{3}$.

Goal

Design a controller by filling in the missing code in the `src/main.cpp` module, which implements a suitable Inverse Kinematics algorithm to meet the following requirements given in descending order of importance:

- **Primary task**: reach the target position with the tip of the manipulator.
- **Secondary task**: align the tip frame along with the target frame.

The test will verify the capabilities of your controller in different conditions, assigning **scores** according to the recorded performance, considering in particular cases where the primary and secondary tasks cannot be achieved at the same time or even cannot be fully completed.

In the latter setting, the test will evaluate the **best effort** made to attain the primary task solely.

An example of a successful control is visible in the animation below:

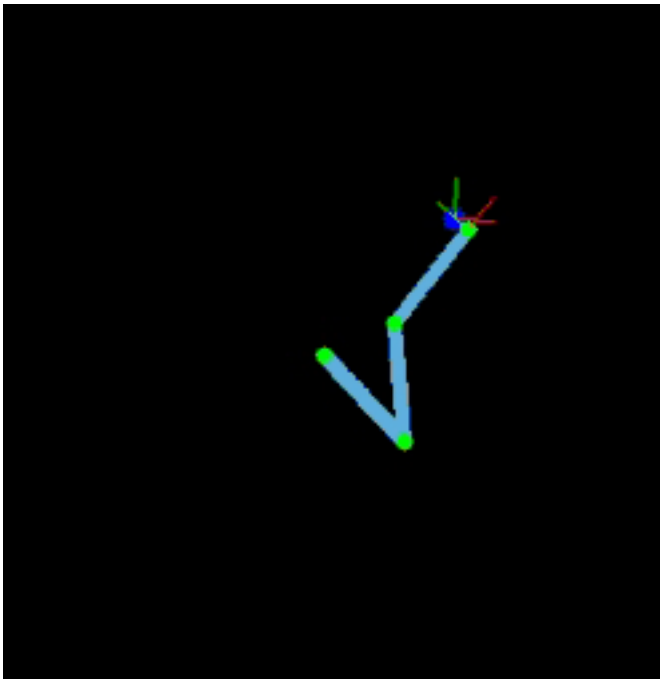

Figure 2: robot

While running the script **test.sh** in the **smoke-test** directory, you might find helpful resorting to the `yarplogger` utility to catch all the printouts produced through `yDebug`, `yWarning`, `yInfo` and the like. Thus, do the following: - Make sure that `yarpserver` is running. - Launch the `yarplogger`. - Launch the smoke test.