Lab 2

Solutions and Grading Guidelines

Forward Kinematics

- 1.1 (15 Points) Write a translate() function that takes a 2D numpy vector as an argument and returns an homogeneous transform corresponding to a translation in the direction of the input.

- This is a pure translation, hence the angle being rotated = 0.

- $T = \begin{bmatrix} R(0) & p \\ 0 & 1 \end{bmatrix}$. Note that $R(0) = I$, where I is an identity matrix.

Forward Kinematics

- 1.2 (15 Points) Write a rotate() function that takes an angle as an argument and returns an homogeneous transform corresponding to a rotation of that angle.

- This is a pure rotation, hence the translation = 0.

- $T = \begin{bmatrix} R(\theta) & 0 \\ 0 & 1 \end{bmatrix}$, where $R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$.

Forward Kinematics

- 1.3 (15 Points) How would you use these functions to compute T_{SH}, T_{HK}, and T_{KF}?
- A homogeneous transform can be decomposed into a pure translation and a pure rotation, hence $T = \begin{bmatrix} R(\theta) & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R(0) & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R(\theta) & 0 \\ 0 & 1 \end{bmatrix}$.
- T = translate(p) @ rotate(theta)

Forward Kinematics

- 1.4 (10 Points) Compute T_{SH} when $\theta_1 = \dfrac{\pi}{3}$ and verify your result by comparing it with the theoretical result.
- See the notebook `solution.ipynb`

Forward Kinematics

- 2.1 (15 Points) Write a function forward_kinematics() that takes 2 angles as input (the 2 DOFs of the planar robot) and returns T_{SF}. You may use the function template below.
- See the notebook `solution.ipynb`

Forward Kinematics

- 2.2 (5 Points) Verify that the function is working correctly by computing the position/orientation of the foot
- when 1) $\theta_1 = \theta_2 = 0$
- and when 2) $\theta_1 = \pi$ and $\theta_2 = -\dfrac{\pi}{2}$.
- Compare with the theoretical result.
- See the notebook `solution.ipynb`

Forward Kinematics

- 3.1 (15 Points) Use your forward_kinematics function in the following code to compute the position of the foot in frame {S} during the movement of the robot. Plot the motion of the foot in 2D using the function plot_foot_trajectory() below
- See the notebook `solution.ipynb`

Forward Kinematics

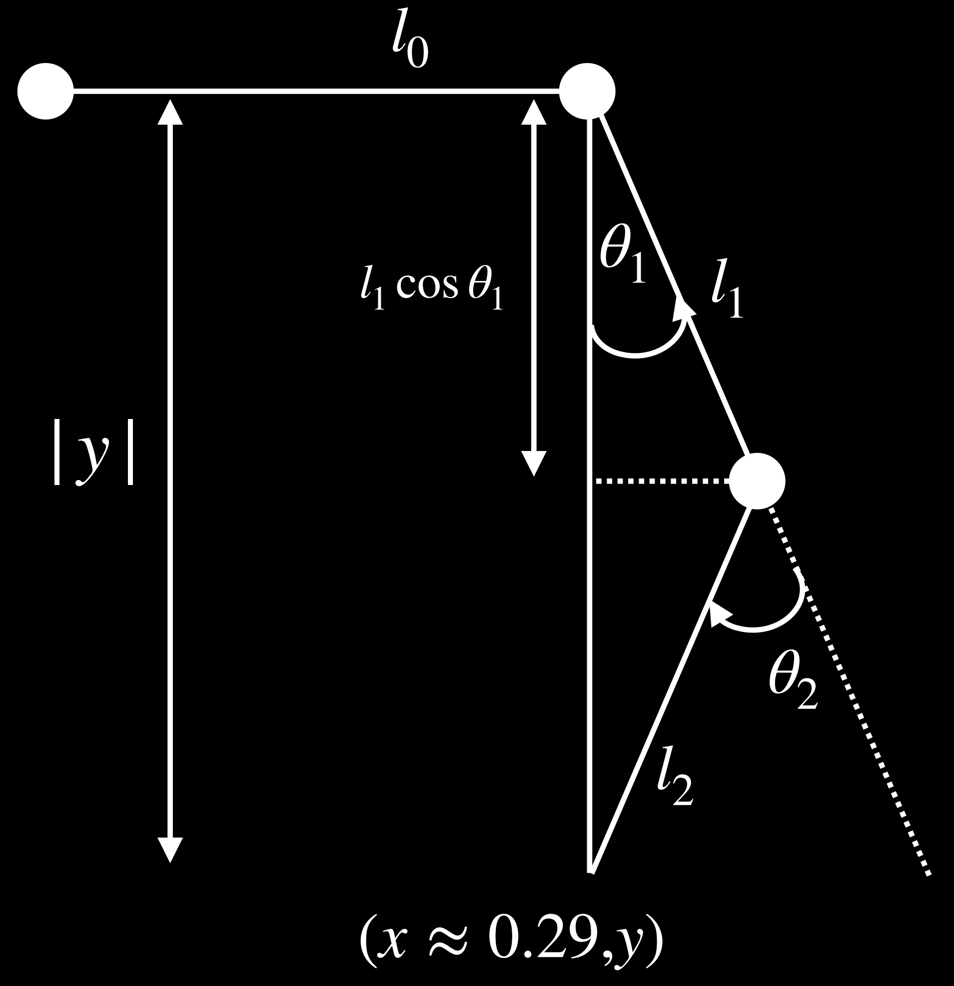
- 3.2 (5 Points) What is the maximum and minimum height reached by the foot during the movement? And the maximum/minimum width?
- See the notebook `solution.ipynb`

Forward Kinematics

- 3.3 (5 Points) Which motion type reaches the highest point? and the lowest?
- Highest: Trajectory 2; Lowest: Trajectory 0

Forward Kinematics

- 3.4 (Bonus 10 Points) Using the results of motion type 1, can you infer (approximately) joint angles of the robot that allows the foot to reach the position (0.29, -0.24)? How can you do this?
- Programmatically: See the notebook `solution.ipynb`
- Geometrically:



Since the trajectory is (approximately) vertical at $x \approx 0.29$ and $l_1 = l_2$, we have an equilateral triangle as shown in the figure. It can be inferred that

$|y| = 2l_1 \cos\theta_1$, and
$|\theta_2| = 2|\theta_1|$

Solving the equations above and noting that $\theta_2 < 0$ (clockwise) gives us

$\theta_1 = 0.72$ and $\theta_2 = -1.44$

(The negated solution can be discarded as it is not on Trajectory 1)