Middle East Technical University   Department of Computer Engineering

CENG 460

Introduction to Robotics for Computer Engineering

Spring 2021-2022

Assignment 3 - Joint Jacobians & Mobile Robots

Due date: June 1st, Wednesday, 23:55

1 Theory (65pts)

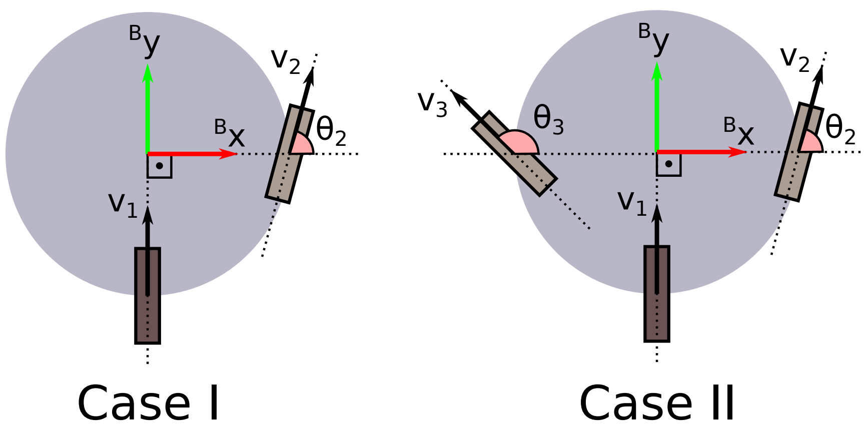

Figure 1: A top down look of the wheel configurations at the edges of a mobile robot with a cylindrical body of radius d. The unit vectors of the body frame ${}^Bx$, ${}^By$ are drawn on the center of the robot. The global orientation θ of the robot is not shown in the drawings for simplicity.

1. (15pts) Figure 1, Case I displays a robot with a fixed wheel on the back with velocity $v_1 = \omega_1 r_1$ at its center and a steered wheel with velocity v_2 at its center angled at $\theta_2 \in [0, \pi)$. Write down the following constraints in the form $f_1({}^B\dot{x}, {}^B\dot{y}, \dot{\theta}, v_1, d) = 0$ for the fixed wheel and $f_2({}^B\dot{x}, {}^B\dot{y}, \dot{\theta}, v_2, \theta_2, d) = 0$ for the steered wheel:
   (a) The wheels experience a pure rolling motion.
   (b) The wheels can not slip sideways.
2. (15pts) Figure out the sets of possible $[v_1,\ v_2]$ and $[{}^B\dot{x},\ {}^B\dot{y},\ \dot{\theta}]$ that adhere to these constraints.
3. **Ackermann Steering** A second steered wheel is attached to the robot in Figure 1, Case II.
   (a) (5pts) Write down the constraints stated in Question 1 for this wheel in the form $f_3({}^B\dot{x}, {}^B\dot{y}, \dot{\theta}, v_3, \theta_3, d) = 0$ with $\theta_3 \in [0, \pi)$.

(b) (10pts) Figure out g and h with $v_3 = g(v_1, v_2, \theta_2)$ and $\theta_3 = h(v_1, v_2, \theta_2)$. Has the set for possible values of $[{}^B\dot{x},\ {}^B\dot{y},\ \dot{\theta}]$ changed?

4. (10pts) For the robot in Assignment 2, assuming $d1 = d2 = d$, write down the Jacobian $[\dot{x},\ \dot{y}]^\top = J(\theta_1,\ \theta_2)[\dot{\theta}_1,\ \dot{\theta}_2]^\top$ for the θ_1, θ_2 shown in Assignment 2, Figure 1.

5. (10pts) Write down the set of $[\theta_1,\ \theta_2]$ where J is singular. Explain the geometric sense of this issue.

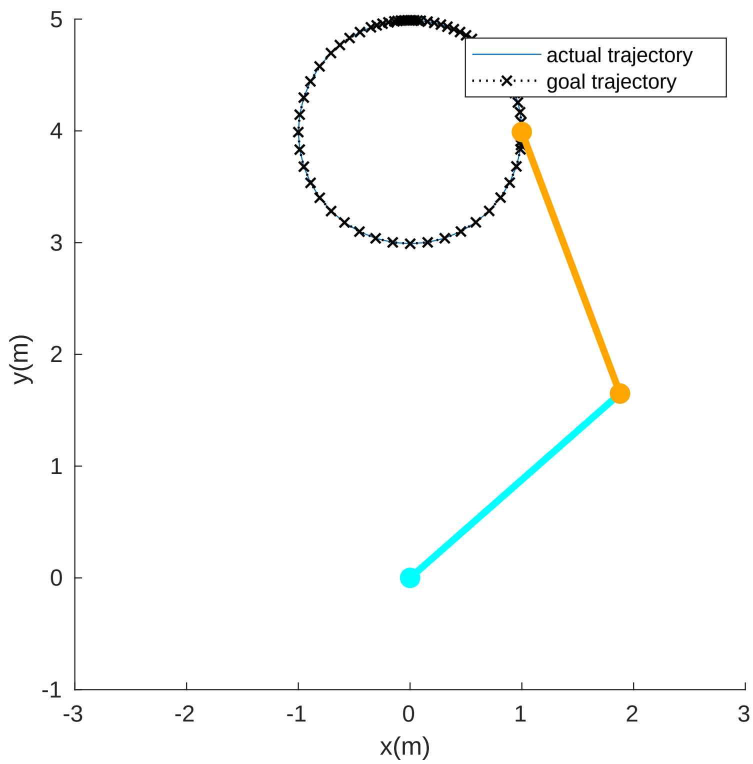

Figure 2: A Sneak Peek of the Programming Part.

2 Programming (35pts)

In this part, We will utilize $J(\theta_1,\ \theta_2)$ to do the programming task in Assignment 2 with numeric integration of joint velocities instead of calculating IK at each point using `ode45`.

1. (10pts) Implement `line_segment_with_jacobians`. Some hints are provided for you throughout the stub file.

2. (5pts) Animate the implementation and plot joint values. How does the solution differ when we lower the tolerance values, compared to sampling few points with IK? Why are some points the solver picked are dense with respect to others? What is the behaviour of joint variables as the arm extends?

3. (20pts) Implement `circle_with_jacobians` *(Hint: model the path as a parametric function of t)*. Animate and plot the joints, similar to Figure 2.

3 Other Specifications

- All of the rotation matrices are to be assumed right handed.
- You can complete the programming assignment without relying on any toolbox, including the robotics toolbox. If you require a toolbox utility, please publicly ask in ODTUCLASS. We may or may not allow it.
- Your programming assignment will be reviewed in a glass-box fashion, therefore please be diligent with your comments in your implementation (you do not have to put comments in every line, just put comments to places that you think it is complicated for the reader to understand. Your comments should provide information about *why* rather than *what*. Avoid "`a = a.*b % element-wise multiply a and b`" sort of comments).
- Please ask your questions regarding the assignment publicly in ODTUCLASS, unless your question reveals part of your solution. In that case, send a mail to `onem@ceng.metu.edu.tr`.
- **This is an individual assignment. Using any piece of code that is not your own is strictly forbidden and constitutes as cheating. This includes friends, previous homeworks, or the Internet. The violators will be punished according to the department regulations.**

4 Submission

Submit your solution as a compressed zip file name `hw3_eXXXXXXX.zip` to ODTUCLASS, containing:

- a pdf scan (you can take this scan with a camera as well using Adobe Scan etc.) of your hand-written answers or a typeset pdf document named `theory_hw3_eXXXXXXX.pdf`.
- All of your matlab code regarding your plots and filled-in stubs provided with your homework. All of your work should be reproducable with the code you supplied. Supply a readme file that explains which file contains what if your file collection is too complex.