Homework/Programming Assignment #2 **Homework/midterm Due: 03/26/2020- 5:00PM**

Name/EID:
Email:
Signature (*required*)
I/We have followed the rules in completing this Assignment.

Name/EID:
Email:
Signature (*required*)
I/We have followed the rules in completing this Assignment.

Question	Points	Total
HA 1	25	
HA 2	25	
HA 3	25	
HA 4	25	
PA	100	
PA. k (Bonus)	15	
PA. m (Bonus)	30	
Presentation* (Bonus)	20	

Instruction:

1. Remember that this is a graded assignment. It is the equivalent of a **midterm take-home exam**.
2. *** You should present the results of the PA in the class** and receive extra bonus depending on the quality of your presentation!
3. **For PA questions**, you need to write a report showing how you derived your equations, describes your approach, test functions, and discusses the results. You should show your test results for each function.

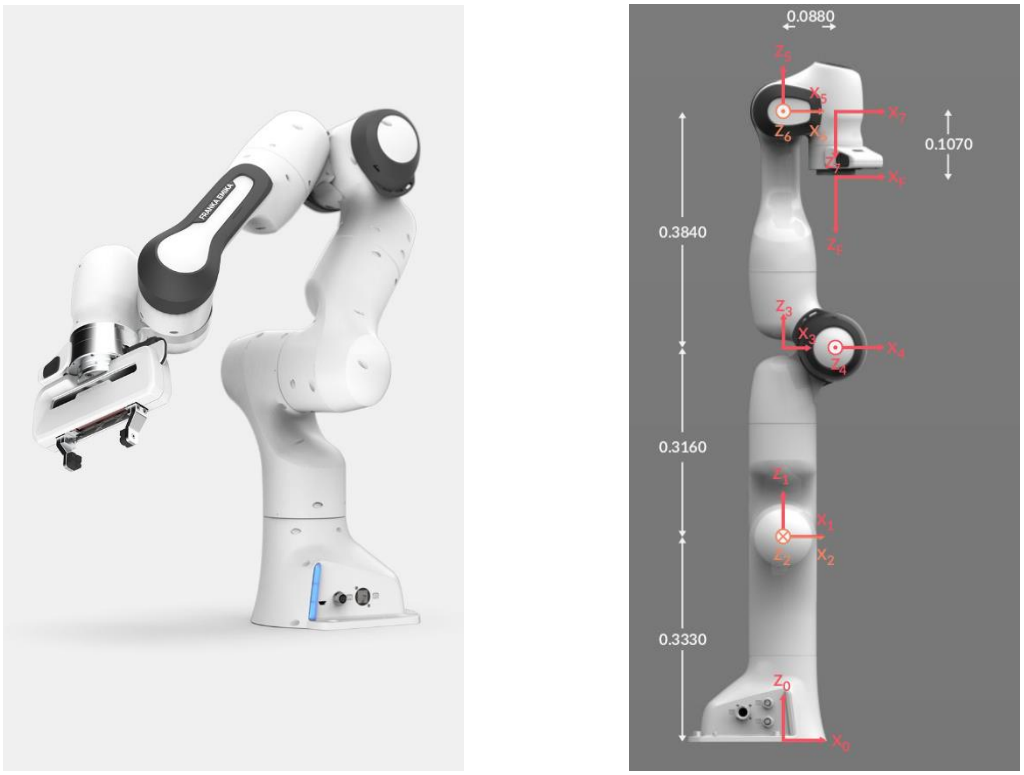
3. You are to work **alone** or **in teams of two** and are **not to discuss the problems with anyone** other than the TAs or the instructor.
4. It is open book, notes, and web. But you should cite any references you consult.
5. Unless I say otherwise in class, it is due before the start of class on the due date mentioned in the P/H Assignment.
6. **Sign and append** this score sheet as the first sheet of your assignment.
7. Remember to submit your assignment in Canvas.

Homework Assignment (HA)

1. **Exercise 4.8** (pg.161) in *Modern Robotics: Mechanics, Planning, and Control* (Lynch et al.) [1].
2. **Exercise 4.11** (pg.162) in *Modern Robotics: Mechanics, Planning, and Control* (Lynch et al.) [1].
3. **Exercise 5.12 (**pg.208) in *Modern Robotics: Mechanics, Planning, and Control* (Lynch et al.) [1].
4. **Exercise 5.13** (pg.209) in *Modern Robotics: Mechanics, Planning, and Control* (Lynch et al.) [1].

➢ Programming Assignment (PA)

Fig. 1 shows the **Franka Emika Panda** 7-DOF robotic manipulator, its pre-defined home/zero position with the assigned frames at each joint, and the corresponding D-H parameters [4].



Joint	a (m)	d (m)	α (rad)	θ (rad)
Joint 1	0	0.333	0	θ_1
Joint 2	0	0	$-\frac{\pi}{2}$	θ_2
Joint 3	0	0.316	$\frac{\pi}{2}$	θ_3
Joint 4	0.0825	0	$\frac{\pi}{2}$	θ_4
Joint 5	-0.0825	0.384	$-\frac{\pi}{2}$	θ_5
Joint 6	0	0	$\frac{\pi}{2}$	θ_6
Joint 7	0.088	0	$\frac{\pi}{2}$	θ_7
Flange	0	0.107	0	0

Fig.1 Panda’s kinematic chain (DH-parameters) and assigned coordinate frames at zero position.

Name	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Unit
q_{max}	2.8973	1.7628	2.8973	-0.0698	2.8973	3.7525	2.8973	rad
q_{min}	-2.8973	-1.7628	-2.8973	-3.0718	-2.8973	-0.0175	-2.8973	rad
$\dot{q}_{max}$	2.1750	2.1750	2.1750	2.1750	2.6100	2.6100	2.6100	$\frac{\text{rad}}{\text{s}}$
$\ddot{q}_{max}$	15	7.5	10	12.5	15	20	20	$\frac{\text{rad}}{\text{s}^2}$

Fig.2 Panda's joint space limits.

Also, Fig. 2 summarizes the Joint space limits of all 7 DoF of Panda. More information about the Panda can be found in its data-sheet.

Considering Fig. 1 and Fig. 2 and the Panda's datasheet:

- **a)** Find the forward kinematics (FK) of the robot using the **space form of the exponential products.**
- **b)** Write the function **"FK_space.m"** that calculates the *space form FK* of the robot and represents the defined frames and screw axis graphically. **Note**: your program should be modular and generic such that it can be used for any type of serial open-chain manipulator!
- **c)** Repeat (a) and (b) for the **body form FK** and write a function **"FK_body.m"**.
- **d)** Find the **space and body form** Jacobian of the robot.
- **e)** Write functions **"J_space.m"** and **"J_body.m"** that calculate the **space and body-form Jacobians** of Panda, respectively.
- **f)** Write a function **"singularity.m"** that calculates the **singularity configurations** of the robot based on the derived Jacobians **for both and space frames.**
- **g)** Write functions that based on the Jacobian of the robot at each configuration:
  - a. shows/plots the manipulability ellipsoid and its axes (i.e., **"ellipsoid_plot.m"**).
  - b. Calculates the isotropy **"J_isotropy.m"**, condition number **"J_condition.m"**, and volume of the ellipsoid **"J_ellipsoid_volume.m"**.
- **h)** Using the derived forward kinematics and Jacobians, write a function **"J_inverse_kinematics.m"**, that uses the iterative numerical inverse kinematics algorithm to control the robot from arbitrary **configuration a** to desired **configuration b**. Test your algorithm in various configurations.
- **i)** Use **Jacobian Transpose algorithm** (**"J_transpose_kinematics.m"**), and repeat **part h**.
- **j)** Using the redundancy resolution approach that we discussed in the class, extend your function in **part h** to define a secondary objective function of the joint variables that **maximizing this manipulability measure and** exploits redundancy to move away from singularities (**"redundancy_resolution.m"**).

- **k)** (**Bonus**) extend your written function **("DLS_inverse_kinematics.m")** in **part h** such that utilizing the **Damped Least Square Approach** can control the robot at configurations **near the singularity situations. Hint:** you may use your written functions in g) and h) to detect singularity situations and switch to this mode for controlling the robot.
- **l)** **For each function**, you should write test functions and discuss the results you obtained in your report. **Do not hand in** only functions, you will receive half of the points if you do not hand in appropriate reports and modular functions!
- **m)** (**Bonus**) Graphical simulation of the robot in ROS, Matlab, or similar software has extra bonus.

➢ References:

1. Lynch and Park, "***Modern Robotics***," Cambridge U. Press, 2017, **Chapter 3.**
2. Bruno, Siciliano, Sciavicco Lorenzo, Villani Luigi, and Oriolo Giuseppe. "*Robotics: modelling, planning and control*." Advanced Textbooks in Control & Signal Processing 4 (2009): 76-82, **Chapter 3**.
3. https://www.franka.de/
4. https://frankaemika.github.io/docs/control_parameters.html
5. https://www.youtube.com/channel/UCjvmirH4BRbWoQwYO4EmXFQ