Assignment 4: Null Space and Grasp Maps

MM8803 Mobile Manipulation, 2020 Edition

February 28, 2020

Part I
Motions in the Null Space

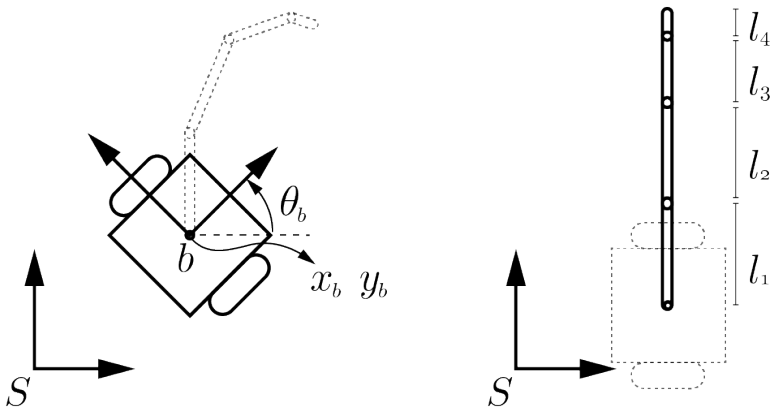
In the previous assignemnt we look at the forward kinematics of a manipulator which gives us the position of the end-effector in the worlds given the joint positions. Then, from the Homogeneous transformation, we are able to compute the Jacobian, such that

$$\dot{x}_{ee} = J\dot{q}$$

The Jacobian transforms the joints rates to a velocity of the end-effector. In this context, with redundant robots is interesting to look at the Null space of the Jacobian ($\mathcal{N}(J)$) which describes a space and if a given $\dot{q}$ lives in this space, then, even though the joints are moving, the end-effector stays in place.

1 Jacobian of a Mobile Manipulator

For Part I of the assignment we will be working with a mobile manipulator which moves on the plane. The base moves in SE(2) and the manipulator is also planar with 4 links.



The mobile base will be a differential drive robot and its motion is defined as:

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta}_b \end{bmatrix} = \begin{bmatrix} \cos(\theta_b) & 0 \\ \sin(\theta_b) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}$$

We will also consider that the manipulator is attached to the center of our mobile base. The first exercise is to compute the Jacobian of the mobile manipulator. For the mobile manipulator, we will have something of the form:

$$\begin{bmatrix} \dot{x}_{ee} \\ \dot{y}_{ee} \\ \dot{\theta}_{ee} \end{bmatrix} = \begin{bmatrix} J_b & | & J_m \end{bmatrix}_{3x7} \begin{bmatrix} \cos(\theta_b)v \\ \sin(\theta_b)v \\ \omega \\ q_m \end{bmatrix}_{7x1}$$

For this Part we will be working with the attached ".ipynb" file (or follow the given Colab Link).

Question 1: You have to implement the Jacobian for this mobile manipulator. You can check if you have the rigth Jacobian with the given unittest.

2 Motion in the Null Space

We are collaborating with our mobile manipulator, which is holding a heavy object that we are working with. You realize that the base of the robot is bothering you, thus you ask the robot to move with some linear and angular velocity to move the base away, but you still want to have the object being held, stay in position. Thus we want this motion to live in the null space of the jacobian we computed.

Question 2: Given the current position of the robot and a given linear (v) and angular (ω) velocity, compute $q_m$which keeps the end-effector in position

Note 1: Complete the "Velocity_in_NullSpace" function for the manipulator. We have added a unittest to check this function, but we will not use it to grade, since the solution might not be unique. Thus, you will be graded on run the animation code on question 2 of the Colab file and the end-effector should stay in place (it might move a few cm, but it should be almost inperceptable.). You might want to use a smaller number of N (number of steps to animate), to make it run faster.

Note 2: We have added the library "from sympy import Matrix" which is usefull to work with matrixes and has functions like: pinv() for the psudoinverse.

Note 3: We are taking a dt time step between each animation frame, thus, we multiply the linear and angular velocity by dt.

3 Null Space Projection

Other usefull tool regarding Null Space is the Null Space Projection. This matrix will take a vector and project it into the null space. This can be use if you have a rough idea in what direction you want to move, and the projector will "find" a vector that lives in the null space. The null space projector is given by

$N = I - J^{\dagger}J$

where $J^{\dagger}$is the psudoinverse of the Jacobian. Implement the null space projector in the function "null_space_projector" and see the animation. For this animation, we have the base being commanded with the control from the null space projector in blue, while in red is the base with the desired linear and angular velocity.

Question 3: Implement the null_space_projector function. Explain why the base motions do not match.

4 Modified Null Space Projection

As we saw for section 2, there is a vector that lives in the null space of the jacobian that is able to keep the manipulator in place while moving the base with the required linear and angular velocity of the base. Can we modify the Null space Projector, such that we can make this happen?

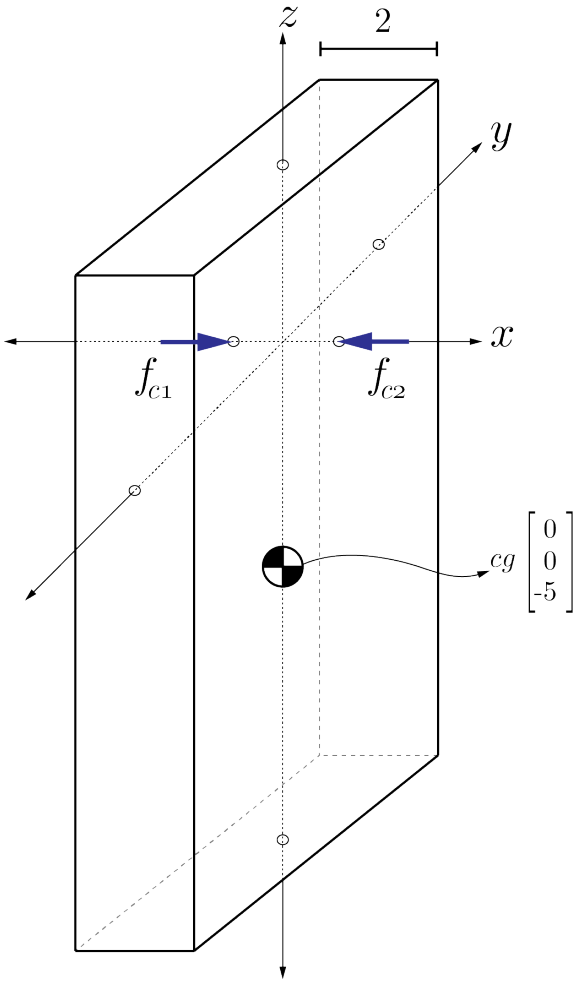
Question 4: Make the Null Space Projection match the linear and angular velocity

Part II

Grasp Maps

Carefully read the MM8803-contacts.pdf note. If you want more detail, check out Chapter 5 from the 1994 book by Murray, Li, and Sastry which is available online at https://www.cds.caltech.edu/∼murray/books/MLS/pdf/mls94-complete.pdf.

The goal of this small assignment is to review what we learned about grasping and apply it to the smartphone example we discussed in class. The coordinate system we will use has Z up, X pointing out of the screen, and Y to the right, as shown in the following figure



We will model two finger contacts, contact 1 at $(-1, 0, 0)$ and contact 2 at $(1, 0, 0)$. The center of gravity is assumed to be at $(0, 0, -5)$, which is where an external force $f_{ext} = -mg$ is applied. Below we investigate what happens for two different contact types: frictionless contacts, and point contacts with friction. Soft finger contacts is left as an exercise if you so please :-)

5 Reuleaux' method for frictionless contacts

First, let us assume frictionless contacts.We can analyze this in the X-Z plane.

Question 5:

1. Please sketch, in the X-Z plane, which IRC's are possible according to Reuleaux' method.
2. Discuss whether the method, which is based on a first-order analysis, actually gives the right answer.
3. If not, discuss which IRC's are actually possible.

6 Grasp map for frictionless contacts

For frictionless point contacts we have, for each contact the following **wrench basis** B_i and **friction cone** FC_i

$$B_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \; FC_i = \{f_1 \geq 0\}$$

and the **contact map** defined as $G_i \triangleq \left[Ad_{T_b^c}\right]^T B_i$, a 6×1 matrix.

Question 6:

1. Calculate $Ad_{T_b^c}$ for each contact.
2. Calculate G_i for each contact.
3. Calculate the 6×2 grasp map $G = [G_1 G_2]$
4. Give an example of a possible wrench $\mathcal{F} = Gf$ and an impossible wrench $\mathcal{F} = Gf$ applied by the contacts on the phone. Give both f and $\mathcal{F}$ in both cases and sketch the situation.

7 Bonus Questions: Point contacts with friction

Next, if we have contacts with friction, we have, for each contact:

$$B_i = \begin{bmatrix} & & \\ & & \\ 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \; F \left\{ C_i = \sqrt{f_1^2 + f_2} \leq \mu f_3, \; f_3 \geq 0 \right\}$$

and the contact map $G_i \triangleq \left[Ad_{T_b^c}\right]^T B_i$ is now a 6×3 matrix.

Question 7:

1. Calculate G_i for each contact.
2. Calculate the 6×6 grasp map $G = [G_1 G_2]$
3. Given a mass m and gravitational constant g, give a wrench F that balances gravity and can be applied by the contacts.
4. Give a wrench $\mathcal{F}$ for which the phone will slide down.

In 3 and 4 above, give both f and $\mathcal{F}$ in both cases, and sketch the situation.

Deliverables

Deliver the ipynb file with the

- Question 1: Passing the Jacobian Unittest. (20 pts)
- Question 2: Show the animation of the end-effector in place while the base moves. (10 pts)
- Question 3: Show the animation of the end-effector in place but the base not doing the correct motion. (20 pts)
- Question 4 (Bonus): Show the animation of the end-effector in place and the base moving correctly. (+ 15 pts)

Deliver a pdf file where you answer:

- Answer to Question 2, why the the base does not match the set velocities (10 pts)
- Answer to Question 5 (you can put a picture of the sketch) (20 pts)
- Answer to Question 6 (20 pts)
- Answer to Question 7 (Bonus) (+ 15 pts)