Assignment 3 – Manipulator Kinematics

Assignment 3 covers manipulator kinematics. Problem 1 will help you revisit three main concepts of manipulator kinematics: the forward (FK) and inverse kinematics (IK), and the manipulator Jacobian. Problem 2 highlights the important difference between forces that a manipulator can generate and forces that it can resist. In problem 3, you will learn about numerical techniques for solving the IK problem of redundant manipulators with singular configurations. Finally, problem 4 will introduce the concept of virtual model control for mobility and manipulation platforms, and cover a technique for resolving parallel chains.

After completion of this assignment you should have a solid understanding of the forward and inverse kinematics problems of manipulators including the important role that Jacobians play, know analytic solution techniques to these problems, and be able to program numerical methods solving the IK for redundant manipulators with singular configurations.

1. Main Concepts of Manipulator Kinematics (6pts)

For the three degree of freedom manipulator shown in Figure 1:

(a) (2pts) Find the forward kinematics map.

(b) (2pts) Solve the inverse kinematics problem using the Paden-Kahan subproblems.

(c) (2pts) Derive the spatial and body Jacobians.

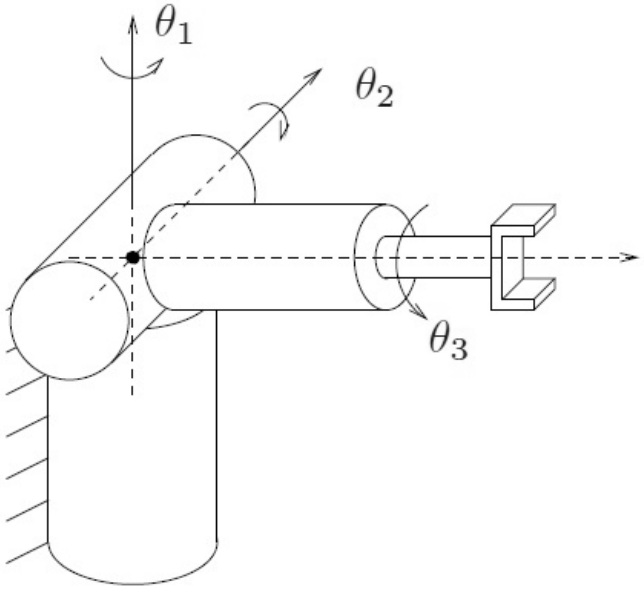

Figure 1: A simple three degree of freedom manipulator.

2. Resisting Force at Singular Configuration (3pts)

Show that if a manipulator is at a singular configuration, then there exists an end-effector wrench F which can be balanced without applying any joint torques. How is the wrench related to the twists which form the columns of the Jacobian?

3. Iterative Inverse Kinematics Methods (6pts+2pts)

In assignment 2, we developed for the Barrett WAM™ arm the forward kinematics map $x = \text{FK}(\theta)$, which maps from the joint configuration of the robot $\theta \in \mathcal{C}_R$ to the pose of the end-effector $x \in SE(3)$. Whereas the FK map is unique and easy to compute analytically, the inverse kinematics map

$$\theta = \text{IK}(x)$$

can be quite difficult to compute analytically, and is generally not unique. In this problem, we will explore numerical methods which use the manipulator's Jacobian matrix to iteratively solve the IK problem of the Barrett WAM™ arm.

The document `iksurvey.pdf` serves as a useful overview of these methods. Another good resource for inverse kinematics algorithms is SSVO pp. 132–147.

Each of these methods relies on the manipulator Jacobian. This matrix is the *derivative* of the forward kinematics map function FK. In other words, this matrix maps from joint velocities $\dot{\theta}$ to end-effector velocities v:

$$v = J(\theta)\,\dot{\theta}$$

In fact, there are different Jacobians, depending on the representation of the velocity v. J depends on the reference point (e.g. the marker tip), whether the rotational and/or translational components of velocity are reported, and how the velocities themselves are represented (e.g. velocity vectors, spatial twists, euler angle/quaternion parameter derivatives, etc).

For this problem, the arm is mounted to a tabletop as in problem 3 of assignment 2 and we will ignore joint limits.

1. (2pts) Implement a function which computes a Jacobian matrix for the marker tip of the Barrett WAM™arm. The Jacobian matrix must produce both the linear and angular components of the marker tip velocity. Indicate the representation of velocities that you used. Evaluate your function at the following joint configuration,

   $$\theta_s = [\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7\ ]^T,$$

   and include the result in your submission.

2. (2pts) The marker tip location x_s for the joint configuration θ_s above is given by

   $$x_s = [\ 0.44543,\ 1.12320,\ 2.22653,\ -0.29883,\ 0.44566,\ 0.84122,\ -0.06664\ ]^T.$$

Note that x_s is given as a position plus a quaternion $[x, y, z, q_i, q_j, q_k, q_0]^T$ equivalent to the homogeneous transformation

$$g_s = \begin{bmatrix} -0.81252 & -0.15424 & -0.56216 & 0.44543 \\ -0.37847 & -0.59390 & 0.70996 & 1.12320 \\ -0.44337 & 0.78962 & 0.42418 & 2.22653 \\ 0.00000 & 0.00000 & 0.00000 & 1.00000 \end{bmatrix}.$$

Suppose the *desired* marker tip location is

$$x_d = [\ 0.46320,\ 1.16402,\ 2.22058,\ -0.29301,\ 0.41901,\ 0.84979,\ 0.12817\ ]^T$$

and compute a velocity v, in the same representation as your Jacobian, which would move the marker tip from x_s in the direction of x_d.

3. (2pts) Implement the Jacobian pseudoinverse iterative method using your results from questions 1 and 2. Indicate any choices for parameters that you chose (step sizes, stopping conditions, etc). Use your implementation to move from the starting configuration θ_s above to each of these goal marker tip poses:

$$x_{d1} = [\ 0.46320,\ 1.16402,\ 2.22058,\ -0.29301,\ 0.41901,\ 0.84979,\ 0.12817\ ]^T$$

$$x_{d2} = [\ 0.49796,\ 0.98500,\ 2.34041,\ -0.11698,\ 0.07755,\ 0.82524,\ 0.54706\ ]^T$$

For each goal pose, submit a joint trajectory file, giving the joint values at each iteration (each file should start with θ_s). Indicate the final joint configurations θ_{di} for each problem. Briefly discuss the performance of your algorithm.

4. (+2pts) Implement the damped least squares method. Use your implementation to solve each of the goal poses from the previous question. Again, submit a joint trajectory file and indicate the final joint configurations for each. Briefly discuss its performance.

4. Virtual Model Control, Kineto-Statics Duality, and Parallel Chains (6pts+2pts)

In the 2001 IJRR article "Virtual Model Control: An Intuitive Approach for Bipedal Locomotion" Pratt and colleagues introduce the virtual model concept for the control of legged robots. Virtual model control assumes that imagined forces act on the center of mass of the robot and stabilize its position. The method then uses Jacobians to map these center of mass forces into joint torques. Please read the paper through section 4 to familiarize yourself with the concept and to understand how you can solve problem 4 in a straightforward way. The paper can be downloaded with the assignment.

Figure 2 shows a 2D seven-link robot with a trunk and two legs. Each leg consists of foot, shank and thigh connected by ankle and knee joints. Both legs are connected to the trunk by hip joints. **We only consider double stance** in which both legs are on the ground and the robot describes a parallel chain mechanism. The segment lengths are $l_1 = l_2 = 0.5m$. The legs are very light. Their mass can be ignored. The trunk has a mass of $m = 80kg$ that is concentrated at the hips, and a moment of inertia

about this point of $2kgm^2$.

The body frame $\{B\}$ is attached to the trunk and the world frame $\{A\}$ is located on the ground. We represent the state of the robot body using (x, z, θ) where (x, z, θ) is the forward kinematic map from A to B. The initial state of the robot given by $z_0 = 0.8m, \dot{z_0} = -1m/s, x_0 = 0, \dot{x_0} = 0.1m/s, \theta_0 = 0, \dot{\theta_0} = 0.1rad/s$. The distances from the centers of the feet to the origin of the world frame $\{A\}$ are $ldis = rdis = 0.2m$.

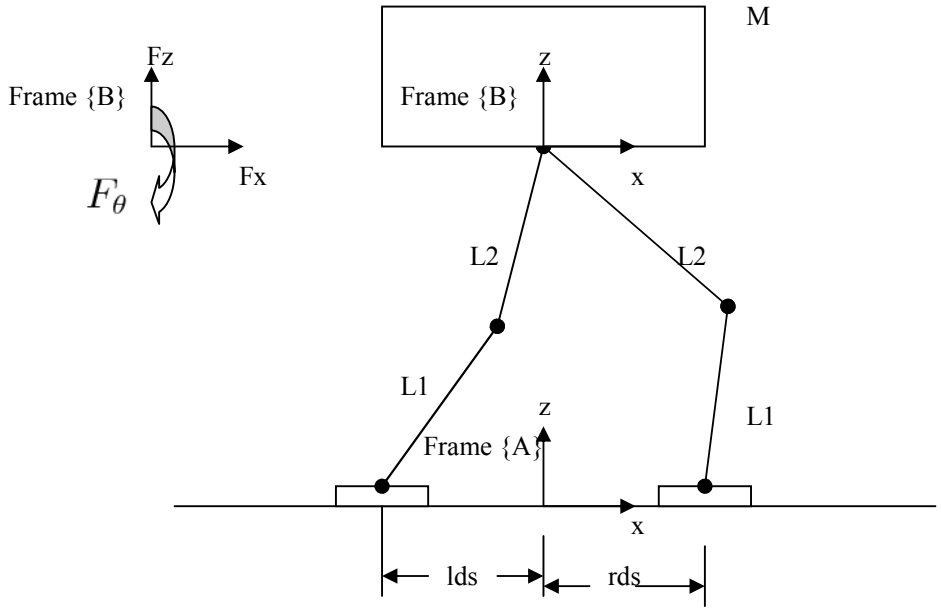

Figure 2: Biped in double support.

(a) (2pts) Given the initial state of the robot body, what are the initial ankle and knee joint angles $\theta_{la}, \theta_{lk}, \theta_{ra}, \theta_{rk}$, as defined in Fig. 4. in the paper? (The inverse kinematics should give you 4 solutions, choose the one matching Figure 2.)

Now imagine virtual spring-dampers acting between frames $\{A\}$ and $\{B\}$ that try to stabilize the trunk of the biped. The virtual forces generated by the spring-dampers can be computed as follows:

$$F^z_{des} = k_1(z_0 - z) - D_1\dot{z} + Mg \quad (1)$$

$$F^x_{des} = -k_2 x - D_2\dot{x} \quad (2)$$

$$F^\theta_{des} = -k_3\theta - D_3\dot{\theta} \quad (3)$$

Viewed from another angle, these virtual forces represent PD controllers that attract the robot's trunk to a desired state $(z_{des} = z_0, \dot{z}_{des} = 0, x_{des} = \dot{x}_{des} = \theta_{des} = \dot{\theta}_{des} = 0)$, where $k_1, k_2, k_3, D_1, D_2, D_3$ are the control parameters we can adjust. Here we set the control parameters to $k1 = 1000, k2 = k3 = 100, D1 = 300, D2 = D3 = 75$.

(b) (2pts) Create a program which numerically integrates the simple dynamics of the trunk, given its initial state $(x_0, z_0, \theta_0, \dot{x}_0, \dot{z}_0, \dot{\theta}_0)$ and the virtual forces in Equations (1-3). For each time step, generate the virtual forces according to the current state $(x, z, \theta, \dot{x}, \dot{z}, \dot{\theta})$, and then compute the new

state for next time step according to the generated virtual forces. Plot figures to show how (x, z, θ) change with time. Does the state eventually converge to the desired state?

To actually generate the desired virtual forces, we need to apply corresponding joint torques. Section 4 of the IJRR paper explains how to relate the desired virtual forces and joint torques using Jacobian matrices and constraints.

(c) (2pts) Neglecting the motion of the legs, add to your program a part to compute estimated joint torques which generate the virtual forces during each time step. Show a figure with the resulting joint torque trajectories. The maximum joint torques that the motors of the biped can produce are 150Nm. Which motors will saturate and endanger the stabilization?

(c) (+2pts) Repeat (c) but now use the IK to adjust the configuration of each leg. Submit an animation that shows the robot moving. Does the stabilization still work? Does it exceed torque limits?