ME 449 Robotic Manipulation
Fall 2014
Problem Set 2
Due Tuesday October 28 at beginning of class

**1.** Programming assignment. You can write in any language, but it will be easier if it is interactive (e.g., matlab or mathematica) and more useful if it can do symbolic math. Make sure to use sensible function names and clearly comment your code so someone reading it can understand what is going on (e.g., what are the inputs and outputs of the function?).

Write a robotics library with functions that perform the following tasks (and some of these functions should make use of others):

(i) Takes a rotation matrix as input and returns the inverse. Do not make use of a matrix inverse function in the language; calculating general matrix inverses is slow.

(ii) Takes a 3-vector $q$ and returns the 3x3 skew-symmetric representation $[q]$.

(iii) Takes a 3x3 skew-symmetric matrix $[q]$ and returns the corresponding 3-vector $q$.

(iv) Takes exponential coordinates $\omega$ (sometimes written $\omega\theta$ or $\hat{\omega}\theta$) and returns the corresponding rotation matrix $R = \exp^{[\omega]}$.

(v) Takes a rotation matrix $R \in SO(3)$ and returns the exponential coordinates $\omega$ based on the matrix logarithm $[\omega] = \log R$.

(vi) Takes an angular velocity $\omega_b$ in {b} and a rotation matrix $R_{ab}$ representing {b} in {a} and returns $\omega_a$ in {a}.

(vii) Takes a 4x4 transformation matrix as input and returns the inverse (do not make use of a matrix inverse function in the language).

(viii) Takes a spatial velocity (twist) $\mathcal{V} = [\omega, v]^T$ and returns $[\mathcal{V}]$.

(ix) Takes $[\mathcal{V}]$ and returns $\mathcal{V}$.

(x) Takes exponential coordinates $\mathcal{V}$ (sometimes written $\mathcal{S}\theta$) and returns the corresponding transformation matrix $T = \exp^{[\mathcal{V}]}$.

(xi) Takes a transformation matrix $T \in SE(3)$ and calculates the exponential coordinates $\mathcal{V}$ based on the matrix logarithm $[\mathcal{V}] = \log T$.

(xii) Takes a transformation matrix $T$ and returns the matrix form of the adjoint map $[\mathrm{Ad}_T]$.

(xiii) Takes a twist $\mathcal{V}_b$ in {b} and a transformation matrix $T_{ab}$ representing {b} in {a} and returns $\mathcal{V}_a$ in {a}.

(xiv) Takes the home configuration $M = T_{sb}(0)$ of a manipulator's end-effector, a list of world-fixed screw axes $\mathcal{S}_i$ in the space frame corresponding to the joint motions, and a list of joint displacements $\theta = (\theta_1, \ldots \theta_n)$, and calculates the configuration of the end-effector when the robot is at these joint coordinates $T_{sb}(\theta)$ (i.e., the forward kinematics). If your programming language supports symbolic math, then this function can also give you the forward kinematics expressions for symbolic joint angles.

(xv) Takes the home configuration $M = T_{sb}(0)$ of a manipulator's end-effector, a list of end-effector-fixed screw axes $\mathcal{B}_i$ in the end-effector body frame corresponding to the joint motions, and a list of joint displacements $\theta = (\theta_1, \ldots \theta_n)$, and calculates the configuration of the end-effector when the robot is at these joint coordinates $T_{sb}(\theta)$ (i.e., the forward kinematics). If your programming language supports symbolic math, then this function can also give you the forward kinematics expressions for symbolic joint angles.

You will turn in your code electronically and also demonstrate it in class. Be on time!

**2.** You will test your software using the RRP robot shown in Figure 1 at its home position. $M$ is $T_{sb}(0)$, the end-effector configuration when the arm is at its home position. First write $M$. Then

(i) Find $M^{-1} = T_{bs}(0)$ using your code.

(ii) Given $\mathcal{V}_s = [0, 0, 1, 0, 0, 0]$, use your code to find $[\mathcal{V}_s]$.

(iii) Given $\mathcal{V}_s = [0, 0, 1, 0, 0, 0]$, use your code to find $\mathcal{V}_b$.

(iv) Given $\mathcal{V}_s = [0, 0, 1, 0, 0, 0]$, use your code to find $\exp^{[\mathcal{V}_s]}$.

(v) Given $M$, use your code to find $\log M$.

(vi) Write the screw axes in the space frame. Then use your code to evaluate the forward kinematics when $\theta = (90°, 90°, 1)$.

(vii) Write the screw axes in the end-effector body frame. Then use your code to evaluate the forward kinematics when $\theta = (90°, 90°, 1)$.

Turn in a screen shot of your code solving these problems.

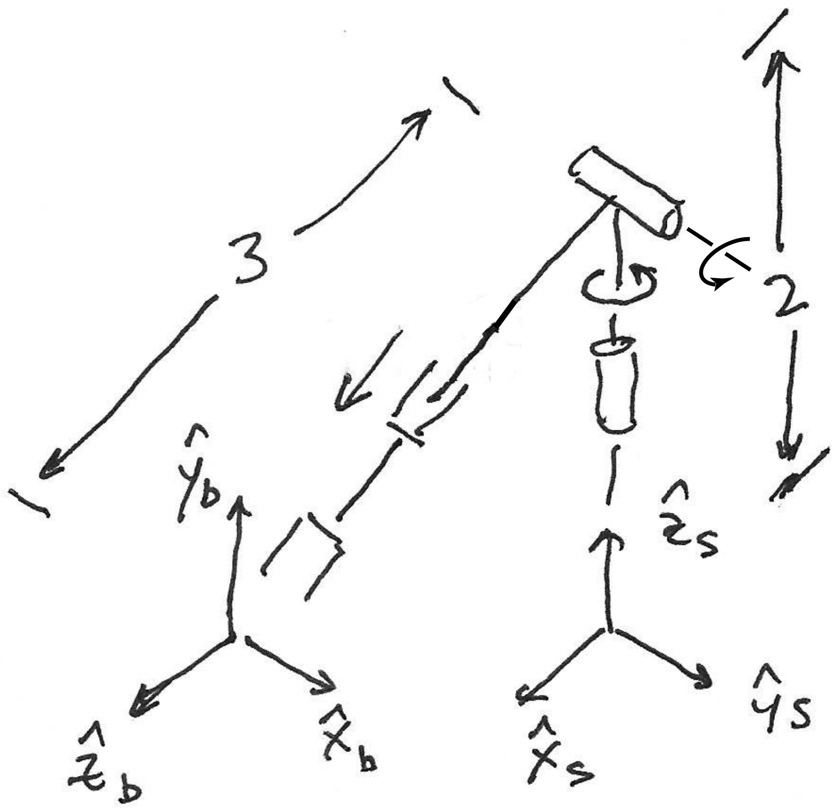


Figure 1: RRP robot in its home configuration. The arrows indicate the sense of positive motion about or along each joint axis.