ME 449 Robotic Manipulation
Fall 2015
Problem Set 3
Due Thursday November 5 at beginning of class (turn in on Canvas)

1. Write the following functions for your robotics library. These functions build on the functions written for the last assignment.

- `FixedJacobian`: Takes a set of joint angles $\theta \in \mathbb{R}^n$ and screw axes $\{\mathcal{S}_i\}$ for the robot joints expressed in the fixed space frame, and returns the space Jacobian $J_s(\theta) \in \mathbb{R}^{6\times n}$. (Another input to `FixedJacobian` could be n, the number of joints of the robot, or n could be determined implicitly from the number of screw axes and joint angles.)

- `BodyJacobian`: Similar to `FixedJacobian`, except the screw axes $\{\mathcal{B}_i\}$ are in the end-effector body frame and it returns the body Jacobian $J_b(\theta) \in \mathbb{R}^{6\times n}$.

- `IKinBody`: A numerical inverse kinematics routine based on Newton-Raphson. Takes a set of screw axes $\{\mathcal{B}_i\}$ for the robot joints expressed in the end-effector body frame, the end-effector zero configuration $M \in SE(3)$, the desired end-effector configuration T_{sd}, an initial guess $\theta_0 \in \mathbb{R}^n$ that is "close" to satisfying $T(\theta_0) = T_{sd}$, and small scalar values $\epsilon_\omega > 0$ and $\epsilon_v > 0$ controlling how close the final solution θ_k must be to the desired answer (see Chapter 6.2). Your routine should also have a maximum number of iterations `maxiterates` before stopping (e.g., 100). Returns a matrix of joint angles of the following form:
$$\begin{bmatrix} \theta_{0,1} & \theta_{0,2} & \cdots & \theta_{0,n} \\ \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,n} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{k,1} & \theta_{k,2} & \cdots & \theta_{k,n} \end{bmatrix},$$
where the top row $(\theta_{0,1}, \ldots, \theta_{0,n})$ is the initial guess vector; the bottom row $(\theta_{k,1}, \ldots, \theta_{k,n})$ is the final solution after k iterations of Newton-Raphson, such that $T(\theta_k)$ is close to T_{sd}, in the sense indicated in Chapter 6.2 (if no satisfying solution is found after `maxiterates` iterations, then the algorithm should terminate and the bottom row of the returned matrix is the last iterate); and the intermediate rows $\theta_i, i = 1, \ldots, k-1$, are the intermediate iterations.

- `IKinFixed`: Similar to `IKinBody` above, except the screw axes $\{\mathcal{S}_i\}$ are in the fixed space frame. Since this algorithm is not given in the notes, also provide a brief derivation of the relevant equations in the algorithm.

2. For the Universal Robots UR5 six-joint robot arm shown in its zero configuration in Figure 1, write the end-effector configuration $M \in SE(3)$ and the six screw axes as (a) $\{\mathcal{B}_i\}$ in the end-effector frame and (b) $\{\mathcal{S}_i\}$ in the fixed space frame.

3. For the redundant Barrett Technology seven-joint WAM robot arm shown in its home configuration in Figure 2, write the end-effector configuration $M \in SE(3)$ and seven screw axes as (a) $\{\mathcal{B}_i\}$ in the end-effector frame and (b) $\{\mathcal{S}_i\}$ in the fixed space frame.

4. Use your function `IKinBody` to find the joint variables θ_d of the UR5 satisfying
$$T(\theta_d) = T_{sd} = \begin{bmatrix} 0 & 1 & 0 & -0.6 \\ 0 & 0 & -1 & 0.1 \\ -1 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$
Distances are in meters. Use $\epsilon_\omega = 0.01$ (i.e., 0.57 degrees) and $\epsilon_v = 0.001$ (i.e., 1 mm). Use the zero configuration $\theta_0 = 0$ as your initial guess. If the configuration is outside the workspace, or if you find that

the zero configuration is too far from a final answer to converge, you may demonstrate `IKinBody` using another T_{sd}.

Note that your inverse kinematics routines do not respect joint limits, so it is possible for your routine to find solutions that are not achievable by the actual robot.

5. Use your function `IKinFixed` to find the joint variables θ_d of the WAM satisfying

$$T(\theta_d) = T_{sd} = \begin{bmatrix} 1 & 0 & 0 & 0.4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Distances are in meters. Use $\epsilon_\omega = 0.01$ (i.e., 0.57 degrees) and $\epsilon_v = 0.001$ (i.e., 1 mm). Use the zero configuration $\theta_0 = 0$ as your initial guess. If the configuration is outside the workspace, or if you find that the zero configuration is too far from a final answer to converge, you may demonstrate `IKinFixed` using another T_{sd}.

6. Use either the Matlab robotics toolbox or ROS/rviz to animate your solutions to Exercises 4 and 5. The matrices of joint angles returned by `IKinBody` and `IKinFixed` should be displayed as consecutive arm configurations, so you can see a (choppy) "movie" of the converging iterations. For each of the UR5 and WAM examples, turn in snapshots of the robot at its initial configuration, one or two intermediate configurations, and the final configuration.

Optional: To see a less choppy movie, try displaying nine configurations between consecutive joint angle vectors. These intermediate images would just be interpolated between the θ_i and θ_{i+1} iterates, at 10% of the travel, 20%, etc.

Information on generating animations in Matlab and ROS/rviz can be found on the class wiki.

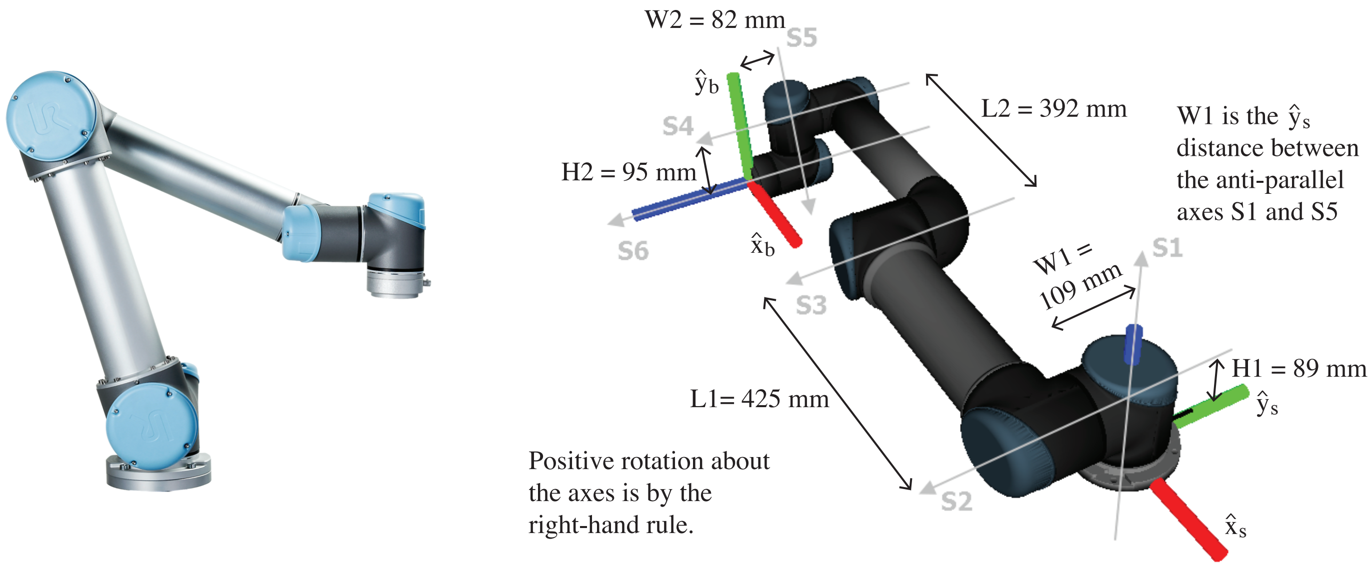
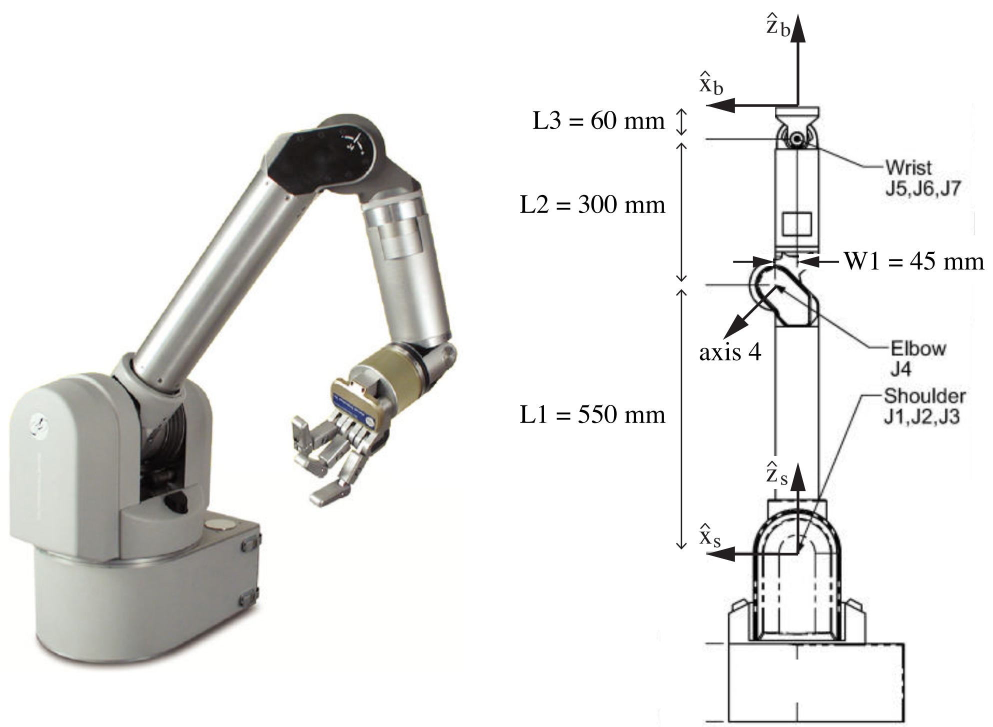


Figure 1: (Left) The Universal Robots UR5 6R arm. (Right) The UR5 at its home configuration.



$\hat{y}_s$ and $\hat{y}_b$ axes are aligned and out of the page. Axes 1, 2, and 3 intersect at the origin of {s}. Axes 5, 6, and 7 intersect at a common point 60 mm from the origin of {b}. Axes 1, 3, 5, and 7 are aligned with $\hat{z}_s$, and axes 2, 4, and 6 are out of the page at the zero configuration. Positive rotation about the axes is by the right-hand rule.

Figure 2: (Left) The Barrett Technology WAM 7R arm. (Right) The WAM at its home configuration.