Advanced Robotic Manipulation — Handout

CS327A (Spring 2016)

Solution Set #1

Problem 1 - Revision: Introduction to Robotics

(a) D-H Convention

Draw the schematic of one manipulator of each type below. Clearly label the base frame {0} as well as each joint frame. Indicate the joint co-ordinates on the schematic.

i. RR (planar)

ii. RR (non-planar)

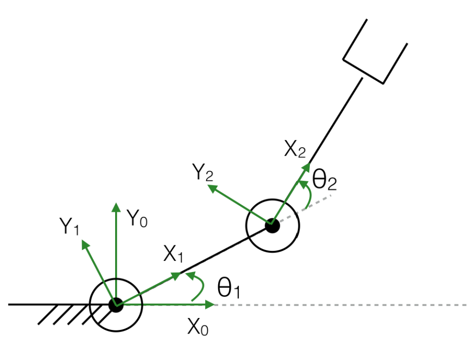
iii. RRP (planar)

iv. PRR (non-planar)

With respect to a positioning task at the end effector, which of the manipulators above are redundant?

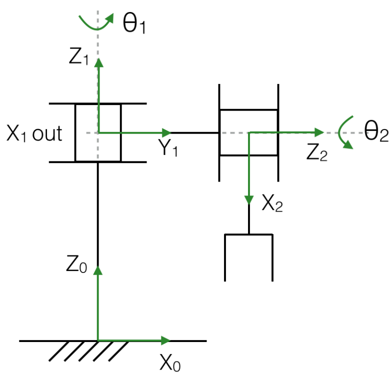
Solution:

i. RR (planar)



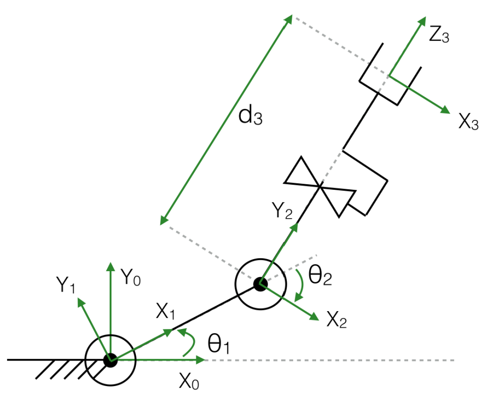
Above manipulator is non-redundant with respect to planar positioning task as the task requires 2 degrees of freedom (DOFs) and the manipulator has 2 DOFs.

ii. RR (non-planar)



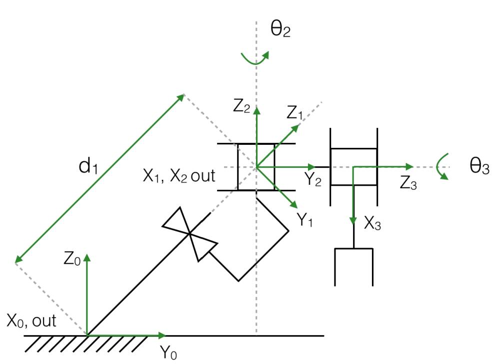
Above manipulator is non-redundant with respect to spatial positioning task as the task requires 3 DOFs and the manipulator has 2 DOFs.

iii. RRP (planar)



Above manipulator is redundant with respect to planar positioning task as the task requires only 2 DOFs and the manipulator has 3 DOFs.

iv. PRR (non-planar)

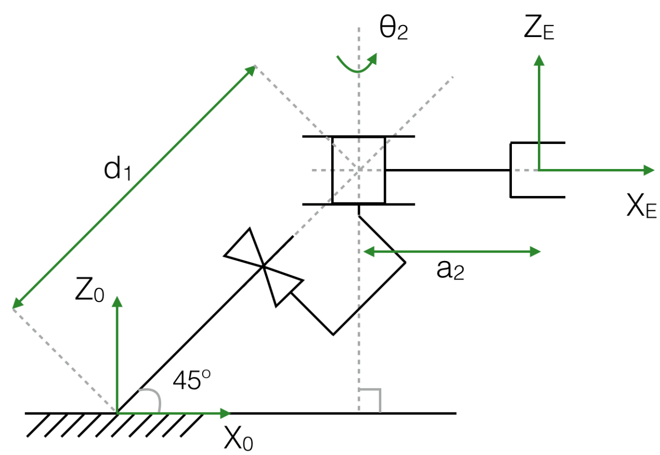


Above manipulator is non-redundant with respect to spatial positioning task as the task requires 3 DOFs and the manipulator has 3 DOFs.

(b) Basic Jacobian

Write down the linear (3 x n) and angular velocity (3 x n) Jacobians at the end-effector for the following manipulators *in the given configuration* in frame $\{0\}$. Try to do so by using the explicit form and visualizing the resulting motion at the end-effector when each joint is moved with unit speed.

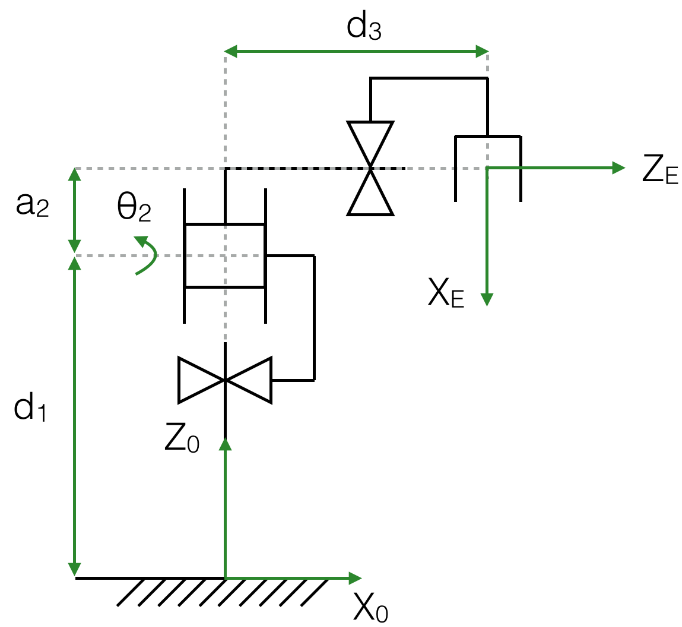
i. PR (non-planar): As shown, $d_1 = 1$, $\theta_2 = \pi/2$ such that X_E is parallel to X_0.



Solution:

$$J_v = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & a_2 \\ \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \quad J_w = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

ii. PRP (non-planar): As shown, $d_1 = 1$, $\theta_2 = \pi/2$, $d_3 = 1$ such that X_E is parallel to Z_0.



Solution:

$$J_v = \begin{bmatrix} 0 & 0 & 1 \\ 0 & a_2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad J_w = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Problem 2 - Quaternion algebra

(a) Conjugation

The conjugate of a quaternion h is defined as $\bar{h}$, such that

$$\bar{h} \equiv w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}$$

Show that $h\bar{h} = \bar{h}h = (w^2 + x^2 + y^2 + z^2) + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$.

Solution:

$$\begin{aligned} h\bar{h} &= (w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k})(w - x\mathbf{i} - y\mathbf{j} - z\mathbf{k}) \\ &= a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \end{aligned}$$

where

$$\begin{aligned} a &= w^2 + x^2 + y^2 + z^2 \\ b &= -wx + wx - yz + yz = 0 \\ c &= -wy + wy - xz + xz = 0 \\ d &= -wz + wz - xy + xy = 0 \end{aligned}$$

(b) Norm of a quaternion

Similar to complex numbers, the norm of a quaternion is defined to be the scalar component of the product of the quaternion with its conjugate.

$$||h|| \equiv \sqrt{w^2 + x^2 + y^2 + z^2} \equiv \sqrt{h\bar{h}}$$

Quaternions which possess the property that $(w^2 + x^2 + y^2 + z^2) = 1$ are called **unit quaternions** and have unit norm. Show that the product of two unit quaternions is a unit quaternion.

Solution: Let h_1 and h_2 be unit quaternions. First, we show the intermediate result that holds for any two quaternions: $\overline{h_1 h_2} = \bar{h}_2 \bar{h}_1$.

$$\begin{aligned}
\overline{h_1 h_2} =& \\
& (w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\
& (-w_1 x_2 - w_2 x_1 - y_1 z_2 + y_2 z_1)\mathbf{i} + \\
& (-w_1 y_2 - w_2 y_1 + x_1 z_2 - x_2 z_1)\mathbf{j} + \\
& (-w_1 z_2 - w_2 z_1 - x_1 y_2 + x_2 y_1)\mathbf{k}
\end{aligned}$$

$$\begin{aligned}
\bar{h}_2 \bar{h}_1 &= (w_2 - x_2\mathbf{i} - y_2\mathbf{j} - z_2\mathbf{k})(w_1 - x_1\mathbf{i} - y_1\mathbf{j} - z_1\mathbf{k}) \\
&= \\
& (w_2 w_1 - x_2 x_1 - y_2 y_1 - z_2 z_1) + \\
& (-w_2 x_1 - w_1 x_2 + y_2 z_1 - y_1 z_2)\mathbf{i} + \\
& (-w_2 y_1 - w_1 y_2 - x_2 z_1 + x_1 z_2)\mathbf{j} + \\
& (-w_2 z_1 - w_1 z_2 + x_2 y_1 - x_1 y_2)\mathbf{k}
\end{aligned}$$

which on rearranging yields

$$\begin{aligned}
&= \\
& (w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + \\
& (-w_1 x_2 - w_2 x_1 - y_1 z_2 + y_2 z_1)\mathbf{i} + \\
& (-w_1 y_2 - w_2 y_1 + x_1 z_2 - x_2 z_1)\mathbf{j} + \\
& (-w_1 z_2 - w_2 z_1 - x_1 y_2 + x_2 y_1)\mathbf{k} \\
&= \overline{h_1 h_2}
\end{aligned}$$

Now, the norm of the product $\|h_1 h_2\|$ yields

$$\|h_1 h_2\| = \sqrt{h_1 h_2 \overline{h_1 h_2}}$$

which, using the property shown above, yields

$$= \sqrt{h_1 h_2 \bar{h}_2 \bar{h}_1}$$

which, since h_2 is a unit quaternion, yields

$$= \sqrt{h_1 \bar{h}_1}$$

which, since h_1 is a unit quaternion, yields

$$= 1$$

(c) Rotation operation quaternion product

Let $r = [r_x \, r_y \, r_z]^T$ be a vector in the 3-D vector space, $\mathbb{R}^3$. Now, we construct a quaternion $h_r = 0 + r_x\mathbf{i} + r_y\mathbf{j} + r_z\mathbf{k}$ whose vector component is identical to the components of r. Also consider a *unit* quaternion $\lambda = \lambda_0 + \lambda_1\mathbf{i} + \lambda_2\mathbf{j} + \lambda_3\mathbf{k}$.

Show that the vector component of the triple product $h_{r'} = \lambda h_r \bar{\lambda}$ is identical to the components of the (3-dimensional) matrix-vector product

$$\begin{bmatrix} 2(\lambda_0^2+\lambda_1^2)-1 & 2(\lambda_1\lambda_2-\lambda_0\lambda_3) & 2(\lambda_1\lambda_3+\lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2+\lambda_0\lambda_3) & 2(\lambda_0^2+\lambda_2^2)-1 & 2(\lambda_2\lambda_3-\lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3-\lambda_0\lambda_2) & 2(\lambda_2\lambda_3+\lambda_0\lambda_1) & 2(\lambda_0^2+\lambda_3^2)-1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}$$

Also give the scalar component of the triple product $h_{r'}$.

Solution:

$$\begin{aligned} \lambda h_r =& (-\lambda_1 r_x - \lambda_2 r_y - \lambda_3 r_z) + \\ & (\lambda_0 r_x + \lambda_2 r_z - \lambda_3 r_y)\mathbf{i} + \\ & (\lambda_0 r_y - \lambda_1 r_z + \lambda_3 r_x)\mathbf{j} + \\ & (\lambda_0 r_z + \lambda_1 r_y - \lambda_2 r_x)\mathbf{k} + \end{aligned}$$

$$\begin{aligned} h_{r'} &= \lambda h_r \bar{\lambda} \\ &= h_{r'w} + h_{r'x}\mathbf{i} + h_{r'y}\mathbf{j} + h_{r'z}\mathbf{k} \end{aligned}$$

where

$$\begin{aligned} h_{r'w} =& \lambda_0(-\lambda_1 r_x - \lambda_2 r_y - \lambda_3 r_z) - (-\lambda_1)(\lambda_0 r_x + \lambda_2 r_z - \lambda_3 r_y) - \\ & (-\lambda_2)(\lambda_0 r_y - \lambda_1 r_z + \lambda_3 r_x) - (-\lambda_3)(\lambda_0 r_z + \lambda_1 r_y - \lambda_2 r_x) \\ =& 0 \\ h_{r'x} =& (-\lambda_1)(-\lambda_1 r_x - \lambda_2 r_y - \lambda_3 r_z) + \lambda_0(\lambda_0 r_x + \lambda_2 r_z - \lambda_3 r_y) + \\ & (-\lambda_3)(\lambda_0 r_y - \lambda_1 r_z + \lambda_3 r_x) - (-\lambda_2)(\lambda_0 r_z + \lambda_1 r_y - \lambda_2 r_x) \\ =& (2(\lambda_0^2 + \lambda_1^2) - 1)r_x + 2(\lambda_1\lambda_2 - \lambda_0\lambda_3)r_y + 2(\lambda_3\lambda_1 + \lambda_0\lambda_2)r_z \\ h_{r'y} =& (-\lambda_2)(-\lambda_1 r_x - \lambda_2 r_y - \lambda_3 r_z) + \lambda_0(\lambda_0 r_y - \lambda_1 r_z + \lambda_3 r_x) - \\ & (-\lambda_3)(\lambda_0 r_x + \lambda_2 r_z - \lambda_3 r_y) + (-\lambda_1)(\lambda_0 r_z + \lambda_1 r_y - \lambda_2 r_x) \\ =& 2(\lambda_2\lambda_1 + \lambda_3\lambda_0)r_x + (2(\lambda_2^2 + \lambda_0^2) - 1)r_y + 2(\lambda_2\lambda_3 - \lambda_0\lambda_1)r_z \\ h_{r'z} =& (-\lambda_3)(-\lambda_1 r_x - \lambda_2 r_y - \lambda_3 r_z) + (\lambda_0)(\lambda_0 r_z + \lambda_1 r_y - \lambda_2 r_x) + \\ & (-\lambda_2)(\lambda_0 r_x + \lambda_2 r_z - \lambda_3 r_y) - (\lambda_1)(\lambda_0 r_y - \lambda_1 r_z + \lambda_3 r_x) \\ =& 2(\lambda_1\lambda_3 - \lambda_0\lambda_2)r_x + 2(\lambda_2\lambda_3 + \lambda_0\lambda_1)r_y + (2(\lambda_0^2 + \lambda_3^2) - 1)r_z \end{aligned}$$

where we have used the fact that $-\lambda_0^2 - \lambda_1^2 - \lambda_2^2 - \lambda_3^2 = -1$.

The above representation clearly matches the result of the product of the rotation matrix representation of the Euler parameters, and the original vector r. The scalar component of the resultant quaternion $h_{r'}$ is zero which shows that it is a pure vector in the 3D space.

(d) Composition of two rotations

Suppose that unit quaternions λ_1 and λ_2 represent two rotation operations. Given a 3-dimensional vector r, show that rotating r first by λ_1, and then by λ_2 is equivalent to rotating r by the resultant of the product $\lambda_2\lambda_1$.

Solution: Let h_r be the quaternion representation of the vector r, $h_r = 0 + r_x\mathbf{i} + r_y\mathbf{j} + r_z\mathbf{k}$. Consider the intermediate result of rotation r by λ_1:

$$h_{r,1} = \lambda_1 h_r \bar{\lambda}_1$$

Rotating $h_{r,1}$ by λ_2 yields the desired final vector r' represented by the quaternion

$$\begin{aligned} h_{r'} &= h_{r,1,2} \\ &= \lambda_2 h_{r,1} \bar{\lambda}_2 \\ &= \lambda_2 \lambda_1 h_r \bar{\lambda}_1 \bar{\lambda}_2 \end{aligned}$$

Using the property $\bar{\lambda}_1\bar{\lambda}_2 = \overline{\lambda_2\lambda_1}$, we obtain

$$h_{r'} = (\lambda_2\lambda_1) h_r (\overline{\lambda_2\lambda_1})$$

which is a rotation of r by the quaternion product $\lambda_2\lambda_1$.

Problem 3 - Instantaneous inverse kinematics

In this problem, you will program a "position-controlled" 7-DOF KUKA-IIWA manipulator in SAI2 to follow a desired end-effector trajectory. The desired position trajectory to be followed is represented by

$$\begin{aligned} x_d &= 0 \\ y_d &= 0.5 + 0.1 \cos \frac{2\pi t}{5} \\ z_d &= 0.65 - 0.05 \cos \frac{4\pi t}{5} \end{aligned}$$

where t is time in seconds. The desired orientation trajectory is represented in Euler parameters as $\lambda_d = (\lambda_{0,d}, \lambda_{1,d}, \lambda_{2,d}, \lambda_{3,d})$ where

$$\begin{aligned}
\lambda_{0,d} &= \frac{1}{\sqrt{2}} \sin\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \\
\lambda_{1,d} &= \frac{1}{\sqrt{2}} \cos\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \\
\lambda_{2,d} &= \frac{1}{\sqrt{2}} \sin\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \\
\lambda_{3,d} &= \frac{1}{\sqrt{2}} \cos\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right)
\end{aligned}$$

The desired operational space coordinates are thus given by $\mathbf{x}_d = [x_d \; y_d \; z_d \; \lambda_{0,d} \; \lambda_{1,d} \; \lambda_{2,d} \; \lambda_{3,d}]^T$.

(a) Desired operational space velocity

Find an expression for the desired operational space velocity vector $\dot{\mathbf{x}}_d = [\dot{x}_d \; \dot{y}_d \; \dot{z}_d \; \dot{\lambda}_{0,d} \; \dot{\lambda}_{1,d} \; \dot{\lambda}_{2,d} \; \dot{\lambda}_{3,d}]^T$. It is sufficient to report just the individual components of the above vector.

Solution:

$$\begin{aligned}
\dot{x}_d &= 0 \\
\dot{y}_d &= -0.1\frac{2\pi}{5} \sin \frac{2\pi t}{5} \\
\dot{z}_d &= 0.05\frac{4\pi}{5} \sin \frac{4\pi t}{5} \\
\dot{\lambda}_{0,d} &= -\frac{\pi^2}{10\sqrt{2}} \cos\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \sin \frac{2\pi t}{5} \\
\dot{\lambda}_{1,d} &= \frac{\pi^2}{10\sqrt{2}} \sin\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \sin \frac{2\pi t}{5} \\
\dot{\lambda}_{2,d} &= -\frac{\pi^2}{10\sqrt{2}} \cos\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \sin \frac{2\pi t}{5} \\
\dot{\lambda}_{3,d} &= \frac{\pi^2}{10\sqrt{2}} \sin\left(\frac{\pi}{4} \cos \frac{2\pi t}{5}\right) \sin \frac{2\pi t}{5}
\end{aligned}$$

(b) Desired end-effector linear and angular velocities

Give an expression for the E and E^+ matrices in terms of x, y, z, λ_0, λ_1, λ_2 and λ_3.

Solution:

$$E = \begin{bmatrix} E_p & 0 \\ 0 & E_r \end{bmatrix}$$

where

$$E_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_r = \frac{1}{2} \begin{bmatrix} -\lambda_1 & -\lambda_2 & -\lambda_3 \\ \lambda_0 & \lambda_3 & -\lambda_2 \\ -\lambda_3 & \lambda_0 & \lambda_1 \\ \lambda_2 & -\lambda_1 & \lambda_0 \end{bmatrix}$$

The inverse of E is given by

$$E^+ = \begin{bmatrix} E_p^{-1} & 0 \\ 0 & E_r^+ \end{bmatrix}$$

where

$$E_p^{-1} = E_p, \quad E_r^+ = 4E_r^T$$

(d) Simulation: implementation

Implement the above algorithm in the space marked as `"FILL ME IN"` in `cs327a/hw1/p1-main.cpp`. Submit your completed `cs327a/hw1/p1-main.cpp` file and any other source files you added along with your homework write-up.

Solution: The solution code is provided under `cs327a/hw1_sol/p1-main-sol.cpp`. To compile and run it, from within the `cs327a/bin` folder, run

```
$ git stash
$ git pull --rebase
$ git stash pop
$ pushd ../build && cmake -DCMAKE_BUILD_TYPE=Release .. && make && popd
$ ./hw1-p1-sol
```