Exercise 1b: Differential Kinematics of the ABB IRB 120

Prof. Marco Hutter*

Teaching Assistants:
Jean-Pierre Sleiman, Maria Vittoria Minniti, Shao Zhang, Jan Preisig and Mayank Mittal

October 6, 2021

Abstract

The aim of this exercise is to calculate the differential kinematics of an ABB robot arm. You will practice on the derivation of velocities for a multi-body system, as well as derive the mapping between generalized velocities and end-effector velocities. A separate MATLAB script will be provided for the 3D visualization of the robot arm.

Figure 1: The ABB IRW 120 robot arm.

1 Introduction

The following exercise is based on an ABB IRB 120 depicted in figure 2. It is a 6-link robotic manipulator with a fixed base. During the exercise you will implement

*original contributors include Michael Blösch, Dario Bellicoso, and Samuel Bachmann

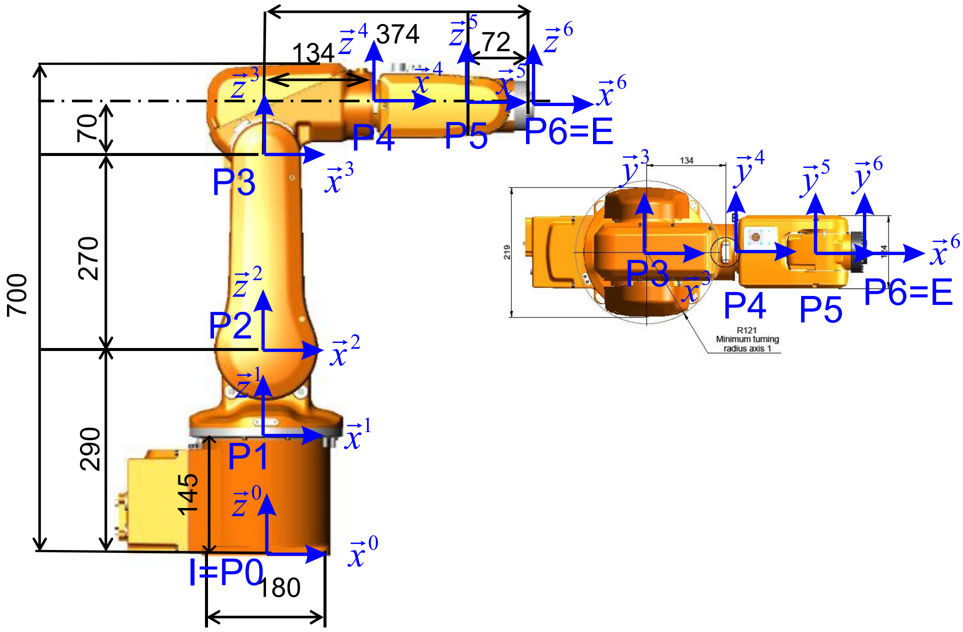

Figure 2: ABB IRB 120 with coordinate systems and joints.

several different MATLAB functions, which you should test carefully since the next exercises will depend on them. To help you with this, we have provided the script prototypes (download from Moodle).
Throughout this document, we will employ I for denoting the inertial world coordinate system (which has the same pose as the coordinate system P0 in figure 2) and E for the coordinate system attached to the end-effector (which has the same pose as the coordinate system P6 in figure 2).

2 Differential Kinematics

Exercise 2.1

In this exercise, we seek to compute an analytical expression for the twist ${}_{\mathcal{I}}\mathbf{w}_E = \begin{bmatrix} {}_{\mathcal{I}}\mathbf{v}_E^T & {}_{\mathcal{I}}\boldsymbol{\omega}_E^T \end{bmatrix}^T$ of the end-effector. To this end, find the analytical expression of the end-effector linear velocity vector ${}_{\mathcal{I}}\mathbf{v}_E$ and angular velocity vector ${}_{\mathcal{I}}\boldsymbol{\omega}_{IE}$ as a function of the linear and angular velocities of the coordinate frames attached to each link.
Hint: start by writing the rigid body motion theorem and extend it to the case of a 6DoF arm.

Solution 2.1

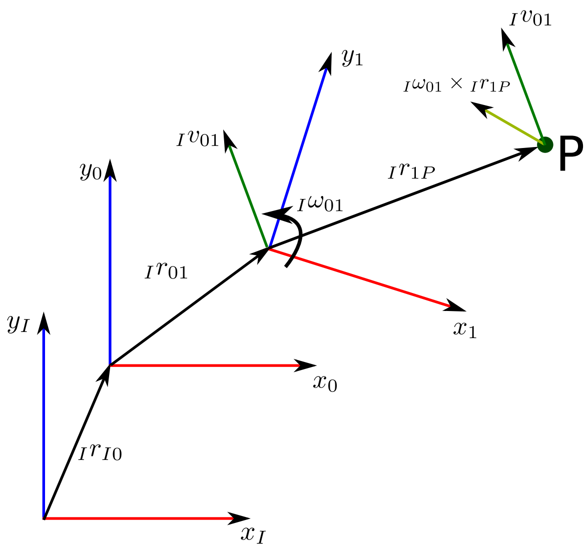

Figure 3: Linear velocity of a point in a rototranslating frame.

Consider the coordinate frames shown in Fig.2. Frame 0 is fixed with respect to the inertial frame $\mathcal{I}$, while frame 1 has a linear velocity $_\mathcal{I}\mathbf{v}_{01}$ and angular velocity $_\mathcal{I}\boldsymbol{\omega}_{01}$ with respect to frame 0. Thus, one has:

$$\begin{aligned} _\mathcal{I}\mathbf{v}_{I1} &= {_\mathcal{I}\mathbf{v}_{I0}} + {_\mathcal{I}\mathbf{v}_{01}} = {_\mathcal{I}\mathbf{v}_{01}} \\ _\mathcal{I}\boldsymbol{\omega}_{I1} &= {_\mathcal{I}\boldsymbol{\omega}_{I0}} + {_\mathcal{I}\boldsymbol{\omega}_{01}} = {_\mathcal{I}\boldsymbol{\omega}_{01}} \end{aligned} \tag{1}$$

Consider a point P that is fixed with respect to frame 1. The linear velocity $_\mathcal{I}\mathbf{v}_{IP}$ of point P with respect to the fixed frame I is given by:

$$\begin{aligned} _\mathcal{I}\mathbf{v}_{IP} &= {_\mathcal{I}\mathbf{v}_{I1}} + {_\mathcal{I}\mathbf{v}_{1P}} \\ &= {_\mathcal{I}\mathbf{v}_{I1}} + \frac{d}{dt}\left(\boldsymbol{C}_{I1} \cdot {_1\mathbf{r}_{1P}}\right) \\ &= {_\mathcal{I}\mathbf{v}_{I1}} + \boldsymbol{C}_{I1} \cdot {_1\dot{\mathbf{r}}_{1P}} + {_\mathcal{I}\boldsymbol{\omega}_{I1}} \times {_\mathcal{I}\mathbf{r}_{1P}}. \end{aligned} \tag{2}$$

If point P is fixed in frame 1, then $_1\dot{\mathbf{r}}_{1P} = 0$.
With this result in mind, consider now a planar two link robot arm with two revolute joints. The coordinate frames are chosen as in Fig.2. Reasoning as before, the linear velocity at the end of the kinematic chain can be found by propagating the linear velocity contributions from the fixed frame $\mathcal{I}$. Hence, one has:

$$\begin{aligned} _\mathcal{I}\mathbf{v}_{I1} &= {_\mathcal{I}\mathbf{v}_{I0}} + {_\mathcal{I}\boldsymbol{\omega}_{I0}} \times {_\mathcal{I}\mathbf{r}_{01}} \\ _\mathcal{I}\mathbf{v}_{I2} &= {_\mathcal{I}\mathbf{v}_{I1}} + {_\mathcal{I}\boldsymbol{\omega}_{I1}} \times {_\mathcal{I}\mathbf{r}_{12}} \\ _\mathcal{I}\mathbf{v}_{IE} &= {_\mathcal{I}\mathbf{v}_{I2}} + {_\mathcal{I}\boldsymbol{\omega}_{I2}} \times {_\mathcal{I}\mathbf{r}_{2E}} \end{aligned} \tag{3}$$

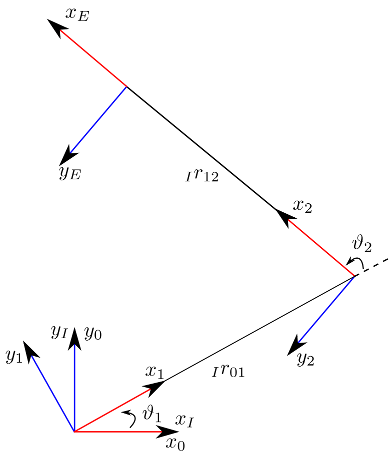

Figure 4: The kinematic structure of a planar two link robot arm. ***Correction:*** *In the figure,* r_{01} *should be replaced with* r_{12}*, and* r_{12} *should be replaced with* r_{2E}.

Combining these results with the fact the frame 0 is fixed with respect to frame $\mathcal{I}$ (i.e. ${}_{\mathcal{I}}\boldsymbol{\omega}_{I0} = \mathbf{0}$, ${}_{\mathcal{I}}\mathbf{v}_{I0} = \mathbf{0}$), the end-effector linear velocity is given by:

$$ {}_{\mathcal{I}}\mathbf{v}_{IE} = {}_{\mathcal{I}}\boldsymbol{\omega}_{I1} \times {}_{\mathcal{I}}\mathbf{r}_{12} + {}_{\mathcal{I}}\boldsymbol{\omega}_{I2} \times {}_{\mathcal{I}}\mathbf{r}_{2E} \tag{4} $$

This result can be extended to the case of the ABB IRB 120, yielding:

$$ \begin{aligned} {}_{\mathcal{I}}\mathbf{v}_{IE} &= {}_{\mathcal{I}}\boldsymbol{\omega}_{I1} \times {}_{\mathcal{I}}\mathbf{r}_{12} + {}_{\mathcal{I}}\boldsymbol{\omega}_{I2} \times {}_{\mathcal{I}}\mathbf{r}_{23} + \cdots + {}_{\mathcal{I}}\boldsymbol{\omega}_{I5} \times {}_{\mathcal{I}}\mathbf{r}_{56} + {}_{\mathcal{I}}\boldsymbol{\omega}_{I6} \times {}_{\mathcal{I}}\mathbf{r}_{6E} \\ &= {}_{\mathcal{I}}\mathbf{v}_{12} + {}_{\mathcal{I}}\mathbf{v}_{23} + \cdots + {}_{\mathcal{I}}\mathbf{v}_{56} + {}_{\mathcal{I}}\mathbf{v}_{6E} \end{aligned} \tag{5} $$

The end-effector rotational velocity ${}_{\mathcal{I}}\boldsymbol{\omega}_{IE}$ is obtained by summing the single joint velocity contributions:

$$ {}_{\mathcal{I}}\boldsymbol{\omega}_{IE} = {}_{\mathcal{I}}\boldsymbol{\omega}_{I0} + {}_{\mathcal{I}}\boldsymbol{\omega}_{01} + {}_{\mathcal{I}}\boldsymbol{\omega}_{12} + \cdots + {}_{\mathcal{I}}\boldsymbol{\omega}_{56} + {}_{\mathcal{I}}\boldsymbol{\omega}_{6E} \tag{6} $$

Exercise 2.2

This exercise focuses on deriving the mapping between the generalized velocities $\dot{\mathbf{q}}$ and the end-effector twist ${}_{\mathcal{I}}\mathbf{w}_E$, namely the *basic* or *geometric* Jacobian ${}_{\mathcal{I}}\mathbf{J}_{e0} = \begin{bmatrix} {}_{\mathcal{I}}\mathbf{J}_P^T & {}_{\mathcal{I}}\mathbf{J}_R^T \end{bmatrix}^T$. To this end, you should derive the translational and rotational Jacobians of the end-effector, respectively ${}_{\mathcal{I}}\mathbf{J}_P$ and ${}_{\mathcal{I}}\mathbf{J}_R$. To do this, you can start from the derivation you found in exercise 1. The Jacobians should depend on the minimal coordinates $\mathbf{q}$ only. Remember that Jacobians map joint space generalized velocities to operational space generalized velocities:

$$
{}_{\mathcal{I}}\mathbf{v}_{IE} = {}_{\mathcal{I}}\mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}} \tag{7}
$$

$$
{}_{\mathcal{I}}\boldsymbol{\omega}_{IE} = {}_{\mathcal{I}}\mathbf{J}_R(\mathbf{q})\dot{\mathbf{q}} \tag{8}
$$

Please implement the following two functions:

```
function J_P = jointToPosJac(q)
  % Input: vector of generalized coordinates (joint angles)
  % Output: Jacobian of the end-effector translation which maps joint
  % velocities to end-effector linear velocities in I frame.

  % Compute the translational jacobian.
  J_P = zeros(3, 6);
end

function J_R = jointToRotJac(q)
  % Input: vector of generalized coordinates (joint angles)
  % Output: Jacobian of the end-effector orientation which maps joint
  % velocities to end-effector angular velocities in I frame.

  % Compute the rotational jacobian.
  J_R = zeros(3, 6);
end
```

Solution 2.2

The translation and rotation Jacobians can be evaluated starting from the results that were obtained in the previous exercises. By combining the analytical expressions of the linear and angular end-effector velocities, one has:

$$
\begin{aligned}
{}_{\mathcal{I}}\mathbf{v}_{IE} &= {}_{\mathcal{I}}\mathbf{v}_{01} + {}_{\mathcal{I}}\mathbf{v}_{12} + \cdots + {}_{\mathcal{I}}\mathbf{v}_{56} + {}_{\mathcal{I}}\mathbf{v}_{6E} \\
&= {}_{\mathcal{I}}\boldsymbol{\omega}_1 \times {}_{\mathcal{I}}\mathbf{r}_{12} + {}_{\mathcal{I}}\boldsymbol{\omega}_2 \times {}_{\mathcal{I}}\mathbf{r}_{23} + \cdots + {}_{\mathcal{I}}\boldsymbol{\omega}_5 \times {}_{\mathcal{I}}\mathbf{r}_{56} + {}_{\mathcal{I}}\boldsymbol{\omega}_E \times {}_{\mathcal{I}}\mathbf{r}_{6E} \\
&= {}_{\mathcal{I}}\boldsymbol{\omega}_1 \times ({}_{\mathcal{I}}\mathbf{r}_{I2} - {}_{\mathcal{I}}\mathbf{r}_{I1}) + {}_{\mathcal{I}}\boldsymbol{\omega}_2 \times ({}_{\mathcal{I}}\mathbf{r}_{I3} - {}_{\mathcal{I}}\mathbf{r}_{I2}) + \cdots + {}_{\mathcal{I}}\boldsymbol{\omega}_5 \times ({}_{\mathcal{I}}\mathbf{r}_{I6} - {}_{\mathcal{I}}\mathbf{r}_{I5}) \\
&= ({}_{\mathcal{I}}\boldsymbol{\omega}_0 + {}_{\mathcal{I}}\boldsymbol{\omega}_{01}) \times ({}_{\mathcal{I}}\mathbf{r}_{I2} - {}_{\mathcal{I}}\mathbf{r}_{I1}) \\
&+ ({}_{\mathcal{I}}\boldsymbol{\omega}_1 + {}_{\mathcal{I}}\boldsymbol{\omega}_{12}) \times ({}_{\mathcal{I}}\mathbf{r}_{I3} - {}_{\mathcal{I}}\mathbf{r}_{I2}) \\
&+ \ldots \\
&+ ({}_{\mathcal{I}}\boldsymbol{\omega}_5 + {}_{\mathcal{I}}\boldsymbol{\omega}_{56}) \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I6})
\end{aligned} \tag{9}
$$

Since the joints are of the revolute type, the relative motion between frames $k-1$ and k will be defined by ${}_{\mathcal{I}}\boldsymbol{\omega}_{k\text{-}1,k} = {}_{\mathcal{I}}\mathbf{n}_k\dot{\theta}_k$, where ${}_{\mathcal{I}}\mathbf{n}_k$ is a vector expressed in $\mathcal{I}$ frame which defines the current rotation direction of joint k and $\dot{\theta}$ is the rate of change in the angular position of joint k. Recalling that the composition rule of angular velocities is:

$$
{}_{\mathcal{I}}\boldsymbol{\omega}_k = {}_{\mathcal{I}}\boldsymbol{\omega}_{k-1} + {}_{\mathcal{I}}\boldsymbol{\omega}_{k-1,k}, \tag{10}
$$

one has:

$$
\begin{aligned}
{}_{\mathcal{I}}\mathbf{v}_{IE} &= ({}_{\mathcal{I}}\boldsymbol{\omega}_0 + {}_{\mathcal{I}}\boldsymbol{\omega}_{01}) \times ({}_{\mathcal{I}}\mathbf{r}_{I2} - {}_{\mathcal{I}}\mathbf{r}_{I1}) \\
&+ ({}_{\mathcal{I}}\boldsymbol{\omega}_1 + {}_{\mathcal{I}}\boldsymbol{\omega}_{12}) \times ({}_{\mathcal{I}}\mathbf{r}_{I3} - {}_{\mathcal{I}}\mathbf{r}_{I2}) \\
&+ \dots \\
&+ ({}_{\mathcal{I}}\boldsymbol{\omega}_5 + {}_{\mathcal{I}}\boldsymbol{\omega}_{56}) \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I6}) \\
&= ({}_{\mathcal{I}}\mathbf{n}_1\dot{\theta}_1) \times ({}_{\mathcal{I}}\mathbf{r}_{I2} - {}_{\mathcal{I}}\mathbf{r}_{I1}) \\
&+ ({}_{\mathcal{I}}\mathbf{n}_1\dot{\theta}_1 + {}_{\mathcal{I}}\mathbf{n}_2\dot{\theta}_2) \times ({}_{\mathcal{I}}\mathbf{r}_{I3} - {}_{\mathcal{I}}\mathbf{r}_{I2}) \\
&+ \dots \\
&+ ({}_{\mathcal{I}}\mathbf{n}_1\dot{\theta}_1 + \dots + {}_{\mathcal{I}}\mathbf{n}_6\dot{\theta}_6) \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I6})
\end{aligned} \tag{11}
$$

Expanding and reordering the terms in the last equation, one has

$$
\begin{aligned}
{}_{\mathcal{I}}\mathbf{v}_{IE} &= {}_{\mathcal{I}}\mathbf{n}_1\dot{\theta}_1 \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I1}) \\
&+ {}_{\mathcal{I}}\mathbf{n}_2\dot{\theta}_2 \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I2}) \\
&+ \dots \\
&+ {}_{\mathcal{I}}\mathbf{n}_6\dot{\theta}_6 \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I6}),
\end{aligned} \tag{12}
$$

which, rewritten in matrix from, gives

$$
\begin{aligned}
{}_{\mathcal{I}}\mathbf{v}_{IE} &= \begin{bmatrix} {}_{\mathcal{I}}\mathbf{n}_1 \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I1}) & \dots & {}_{\mathcal{I}}\mathbf{n}_6 \times ({}_{\mathcal{I}}\mathbf{r}_{IE} - {}_{\mathcal{I}}\mathbf{r}_{I6}) \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix} \\
&= {}_{\mathcal{I}}\mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}},
\end{aligned} \tag{13}
$$

where ${}_{\mathcal{I}}\mathbf{J}_P(\mathbf{q})$ is the translation Jacobian matrix that projects a vector from the joint velocity space to the cartesian linear velocity space.
Using the results obtained by solving Exercise 1, and taking into account that ${}_{\mathcal{I}}\boldsymbol{\omega}_{I0}$ and ${}_{\mathcal{I}}\boldsymbol{\omega}_{6E}$ are both equal to zero, one has

$$
\begin{aligned}
{}_{\mathcal{I}}\boldsymbol{\omega}_{IE} &= {}_{\mathcal{I}}\boldsymbol{\omega}_{01} + {}_{\mathcal{I}}\boldsymbol{\omega}_{12} + \dots + {}_{\mathcal{I}}\boldsymbol{\omega}_{56} \\
&= {}_{\mathcal{I}}\mathbf{n}_1\dot{\theta}_1 + {}_{\mathcal{I}}\mathbf{n}_2\dot{\theta}_2 + \dots + {}_{\mathcal{I}}\mathbf{n}_6\dot{\theta}_6 \\
&= \begin{bmatrix} {}_{\mathcal{I}}\mathbf{n}_1 & {}_{\mathcal{I}}\mathbf{n}_2 & \dots & {}_{\mathcal{I}}\mathbf{n}_6 \end{bmatrix} \cdot \dot{\mathbf{q}} \\
&= {}_{\mathcal{I}}\mathbf{J}_R(\mathbf{q}) \cdot \dot{\mathbf{q}},
\end{aligned} \tag{14}
$$

where $\mathbf{J}_R(\mathbf{q})$ is the rotation Jacobian matrix that projects a vector in the joint velocity space to the Cartesian angular velocity space.

```
function J_P = jointToPosJac(q)
  % Input: vector of generalized coordinates (joint angles)
  % Output: Jacobian of the end-effector orientation which maps joint
  % velocities to end-effector linear velocities in I frame.

  % Compute the relative homogeneous transformation matrices.
  T_I0 = getTransformI0();
  T_01 = jointToTransform01(q(1));
  T_12 = jointToTransform12(q(2));
  T_23 = jointToTransform23(q(3));
  T_34 = jointToTransform34(q(4));
  T_45 = jointToTransform45(q(5));
  T_56 = jointToTransform56(q(6));

  % Compute the homogeneous transformation matrices from frame k to the
  % inertial frame I.
```

```
  T_I1 = T_I0*T_01;
  T_I2 = T_I1*T_12;
  T_I3 = T_I2*T_23;
  T_I4 = T_I3*T_34;
  T_I5 = T_I4*T_45;
  T_I6 = T_I5*T_56;

  % Extract the rotation matrices from each homogeneous transformation
  % matrix.
  R_I1 = T_I1(1:3,1:3);
  R_I2 = T_I2(1:3,1:3);
  R_I3 = T_I3(1:3,1:3);
  R_I4 = T_I4(1:3,1:3);
  R_I5 = T_I5(1:3,1:3);
  R_I6 = T_I6(1:3,1:3);

  % Extract the position vectors from each homogeneous transformation
  % matrix.
  r_I_I1 = T_I1(1:3,4);
  r_I_I2 = T_I2(1:3,4);
  r_I_I3 = T_I3(1:3,4);
  r_I_I4 = T_I4(1:3,4);
  r_I_I5 = T_I5(1:3,4);
  r_I_I6 = T_I6(1:3,4);

  % Define the unit vectors around which each link rotates in the ...
      precedent
  % coordinate frame.
  n_1 = [0 0 1]';
  n_2 = [0 1 0]';
  n_3 = [0 1 0]';
  n_4 = [1 0 0]';
  n_5 = [0 1 0]';
  n_6 = [1 0 0]';

  % Compute the end-effector position vector.
  r_I_IE = jointToPosition(q);

  % Compute the translational jacobian.
  J_P = [   cross(R_I1*n_1, r_I_IE - r_I_I1) ...
            cross(R_I2*n_2, r_I_IE - r_I_I2) ...
            cross(R_I3*n_3, r_I_IE - r_I_I3) ...
            cross(R_I4*n_4, r_I_IE - r_I_I4) ...
            cross(R_I5*n_5, r_I_IE - r_I_I5) ...
            cross(R_I6*n_6, r_I_IE - r_I_I6) ...
        ];

end


function J_R = jointToRotJac(q)
  % Input: vector of generalized coordinates (joint angles)
  % Output: Jacobian of the end-effector orientation which maps joint
  % velocities to end-effector angular velocities in I frame.

  % Compute the relative homogeneous transformation matrices.
  T_I0 = getTransformI0();
  T_01 = jointToTransform01(q(1));
  T_12 = jointToTransform12(q(2));
  T_23 = jointToTransform23(q(3));
  T_34 = jointToTransform34(q(4));
  T_45 = jointToTransform45(q(5));
  T_56 = jointToTransform56(q(6));

  % Compute the homogeneous transformation matrices from frame k to the
  % inertial frame I.
  T_I1 = T_I0*T_01;
```

```
    T_I2 = T_I1*T_12;
    T_I3 = T_I2*T_23;
    T_I4 = T_I3*T_34;
    T_I5 = T_I4*T_45;
    T_I6 = T_I5*T_56;

    % Extract the rotation matrices from each homogeneous transformation
    % matrix.
    R_I1 = T_I1(1:3,1:3);
    R_I2 = T_I2(1:3,1:3);
    R_I3 = T_I3(1:3,1:3);
    R_I4 = T_I4(1:3,1:3);
    R_I5 = T_I5(1:3,1:3);
    R_I6 = T_I6(1:3,1:3);

    % Define the unit vectors around which each link rotates in the ...
        precedent
    % coordinate frame.
    n_1 = [0 0 1]';
    n_2 = [0 1 0]';
    n_3 = [0 1 0]';
    n_4 = [1 0 0]';
    n_5 = [0 1 0]';
    n_6 = [1 0 0]';

    % Compute the rotational jacobian.
    J_R = [    R_I1*n_1 ...
               R_I2*n_2 ...
               R_I3*n_3 ...
               R_I4*n_4 ...
               R_I5*n_5 ...
               R_I6*n_6 ...
          ];

end
```