The major expense in calculating kinematics is often the calculation of the transcendental functions (sine and cosine). When these functions are available as part of a standard library, they are often computed from a series expansion at the cost of many multiply times. At the expense of some required memory, many manipulation systems employ table-lookup implementations of the transcendental functions. Depending on the scheme, this reduces the amount of time required to calculate a sine or cosine to two or three multiply times or less [6].

The computation of the kinematics as in (3.14) is redundant, in that nine quantities are calculated to represent orientation. One means that usually reduces computation is to calculate only two columns of the rotation matrix and then to compute a cross product (requiring only six multiplications and three additions) to compute the third column. Obviously, one chooses the two least complicated columns to compute.

BIBLIOGRAPHY

[1] J. Denavit and R.S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, pp. 215–221, June 1955.

[2] J. Lenarčič, "Kinematics," in *The International Encyclopedia of Robotics*, R. Dorf and S. Nof, Editors, John C. Wiley and Sons, New York, 1988.

[3] J. Colson and N.D. Perreira, "Kinematic Arrangements Used in Industrial Robots," *13th Industrial Robots Conference Proceedings*, April 1983.

[4] T. Turner, J. Craig, and W. Gruver, "A Microprocessor Architecture for Advanced Robot Control," 14th ISIR, Stockholm, Sweden, October 1984.

[5] W. Schiehlen, "Computer Generation of Equations of Motion," in *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, E.J. Haug, Editor, Springer-Verlag, Berlin & New York, 1984.

[6] C. Ruoff, "Fast Trigonometric Functions for Robot Control," *Robotics Age*, November 1981.

EXERCISES

3.1 [15] Compute the kinematics of the planar arm from Example 3.3.

3.2 [37] Imagine an arm like the PUMA 560, except that joint 3 is replaced with a prismatic joint. Assume the prismatic joint slides along the direction of $\hat{X}_1$ in Fig. 3.18; however, there is still an offset equivalent to d_3 to be accounted for. Make any additional assumptions needed. Derive the kinematic equations.

3.3 [25] The arm with three degrees of freedom shown in Fig. 3.29 is like the one in Example 3.3, except that joint 1's axis is not parallel to the other two. Instead, there is a twist of 90 degrees in magnitude between axes 1 and 2. Derive link parameters and the kinematic equations for ${}^B_W T$. Note that no l_3 need be defined.

3.4 [22] The arm with three degrees of freedom shown in Fig. 3.30 has joints 1 and 2 perpendicular and joints 2 and 3 parallel. As pictured, all joints are at their zero location. Note that the positive sense of the joint angle is indicated. Assign link frames {0} through {3} for this arm—that is, sketch the arm, showing the attachment of the frames. Then derive the transformation matrices ${}^0_1T$, ${}^1_2T$, and ${}^2_3T$.

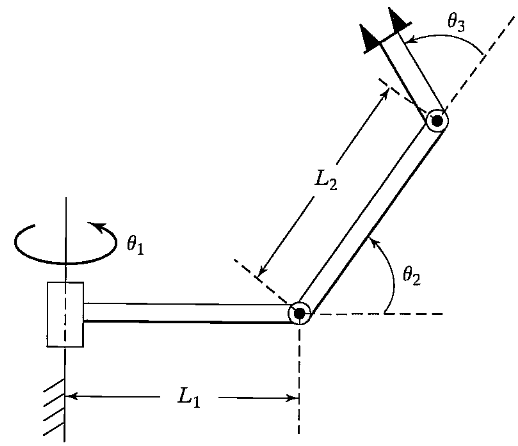

FIGURE 3.29: The $3R$ nonplanar arm (Exercise 3.3).

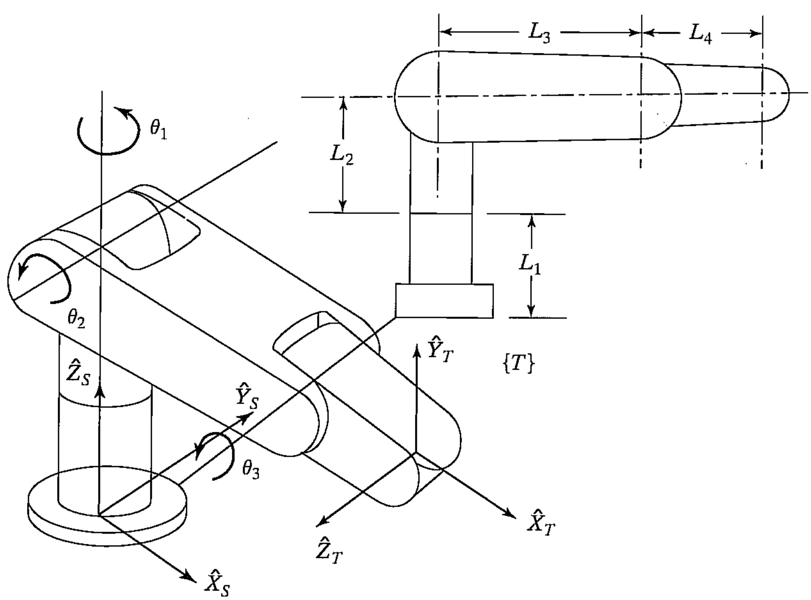

FIGURE 3.30: Two views of a $3R$ manipulator (Exercise 3.4).

3.5 [26] Write a subroutine to compute the kinematics of a PUMA 560. Code for speed, trying to minimize the number of multiplications as much as possible. Use the procedure heading (or equivalent in C)

```
Procedure KIN(VAR theta: vec6; VAR wrelb: frame);
```

Count a sine or cosine evaluation as costing 5 multiply times. Count additions as costing 0.333 multiply times and assignment statements as 0.2 multiply times.

Count a square-root computation as costing 4 multiply times. How many multiply times do you need?

3.6 [20] Write a subroutine to compute the kinematics of the cylindrical arm in Example 3.4. Use the procedure heading (or equivalent in C)

```
Procedure KIN(VAR jointvar: vec3; VAR wrelb: frames);
```

Count a sine or cosine evaluation as costing 5 multiply times. Count additions as costing 0.333 multiply times and assignment statements as 0.2 multiply times. Count a square-root computation as costing 4 multiply times. How many multiply times do you need?

3.7 [22] Write a subroutine to compute the kinematics of the arm in Exercise 3.3. Use the procedure heading (or equivalent in C)

```
Procedure KIN(VAR theta: vec3; VAR wrelb: frame);
```

Count a sine or cosine evaluation as costing 5 multiply times. Count additions as costing 0.333 multiply times and assignment statements as 0.2 multiply times. Count a square-root computation as costing 4 multiply times. How many multiply times do you need?

3.8 [13] In Fig. 3.31, the location of the tool, ${}^{W}_{T}T$, is not accurately known. Using force control, the robot feels around with the tool tip until it inserts it into the socket (or Goal) at location ${}^{S}_{G}T$. Once in this "calibration" configuration (in which $\{G\}$ and $\{T\}$ are coincident), the position of the robot, ${}^{B}_{W}T$, is figured out by reading the joint angle sensors and computing the kinematics. Assuming ${}^{B}_{S}T$ and ${}^{S}_{G}T$ are known, give the transform equation to compute the unknown tool frame, ${}^{W}_{T}T$.

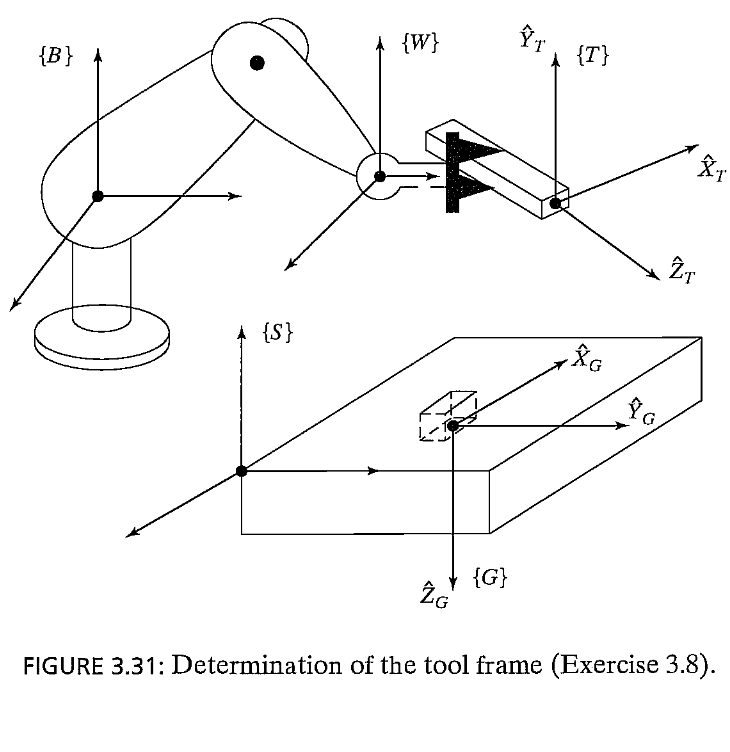

FIGURE 3.31: Determination of the tool frame (Exercise 3.8).

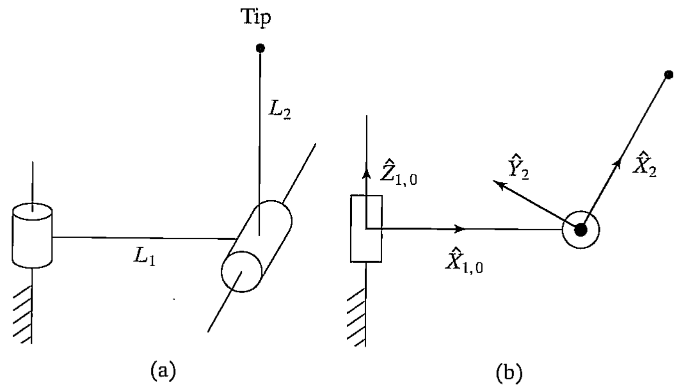

FIGURE 3.32: Two-link arm with frame assignments (Exercise 3.9).

3.9 [11] For the two-link manipulator shown in Fig. 3.32(a), the link-transformation matrices, ${}^0_1T$ and ${}^1_2T$, were constructed. Their product is

$$
{}^0_2T = \begin{bmatrix} c\theta_1 c\theta_2 & -c\theta_1 s\theta_2 & s\theta_1 & l_1 c\theta_1 \\ s\theta_1 c\theta_2 & -s\theta_1 s\theta_2 & -c\theta_1 & l_1 s\theta_1 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
$$

The link-frame assignments used are indicated in Fig. 3.32(b). Note that frame {0} is coincident with frame {1} when $\theta_1 = 0$. The length of the second link is l_2. Find an expression for the vector ${}^0P_{tip}$, which locates the tip of the arm relative to the {0} frame.

3.10 [39] Derive kinematic equations for the Yasukawa Motoman robot (see Section 3.7) that compute the position and orientation of the wrist frame directly from actuator values, rather than by first computing the joint angles. A solution is possible that requires only 33 multiplications, two square roots, and six sine or cosine evaluations.

3.11 [17] Figure 3.33 shows the schematic of a wrist which has three intersecting axes that are not orthogonal. Assign link frames to this wrist (as if it were a 3-DOF manipulator), and give the link parameters.

3.12 [08] Can an arbitrary rigid-body transformation always be expressed with four parameters (a, α, d, θ) in the form of equation (3.6)?

3.13 [15] Show the attachment of link frames for the 5-DOF manipulator shown schematically in Fig. 3.34.

3.14 [20] As was stated, the relative position of any two lines in space can be given with two parameters, a and α, where a is the length of the common perpendicular joining the two and α is the angle made by the two axes when projected onto a plane normal to the common perpendicular. Given a line defined as passing through point p with unit-vector direction $\hat{m}$ and a second passing through point q with unit-vector direction $\hat{n}$, write expressions for a and α.

3.15 [15] Show the attachment of link frames for the 3-DOF manipulator shown schematically in Fig. 3.35.

3.16 [15] Assign link frames to the RPR planar robot shown in Fig. 3.36, and give the linkage parameters.

3.17 [15] Show the attachment of link frames on the three-link robot shown in Fig. 3.37.

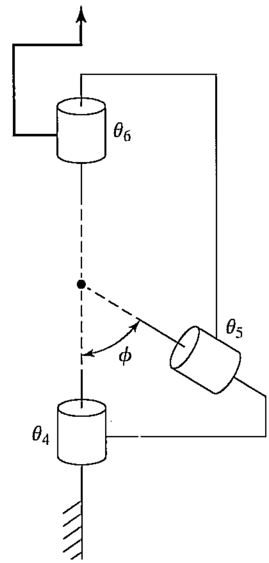

FIGURE 3.33: $3R$ nonorthogonal-axis robot (Exercise 3.11).

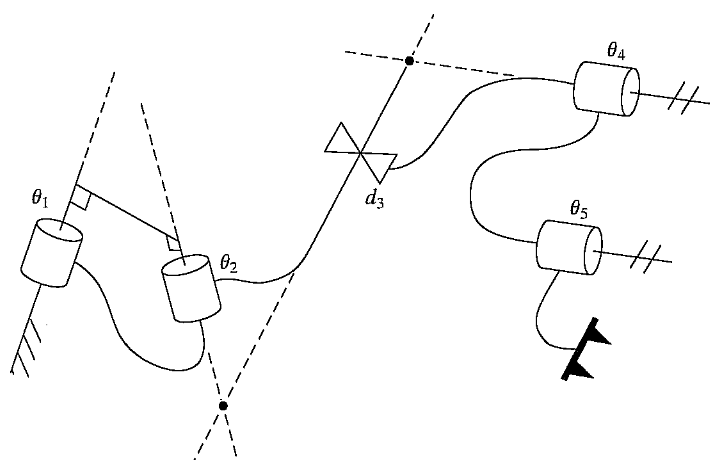

FIGURE 3.34: Schematic of a $2RP2R$ manipulator (Exercise 3.13).

3.18 [15] Show the attachment of link frames on the three-link robot shown in Fig. 3.38.
3.19 [15] Show the attachment of link frames on the three-link robot shown in Fig. 3.39.
3.20 [15] Show the attachment of link frames on the three-link robot shown in Fig. 3.40.
3.21 [15] Show the attachment of link frames on the three-link robot shown in Fig. 3.41.
3.22 [18] Show the attachment of link frames on the $P3R$ robot shown in Fig. 3.42. Given your frame assignments, what are the signs of d_2, d_3, and a_2?

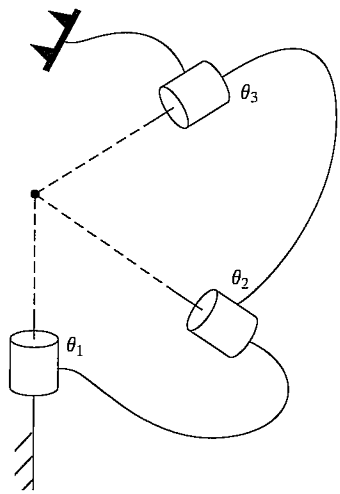

FIGURE 3.35: Schematic of a $3R$ manipulator (Exercise 3.15).

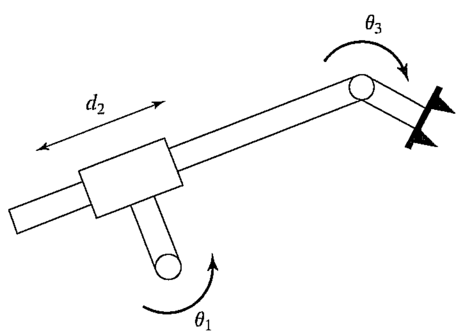

FIGURE 3.36: RPR planar robot (Exercise 3.16).

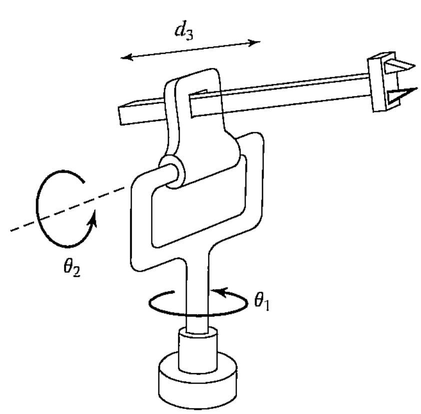

FIGURE 3.37: Three-link RRP manipulator (Exercise 3.17).

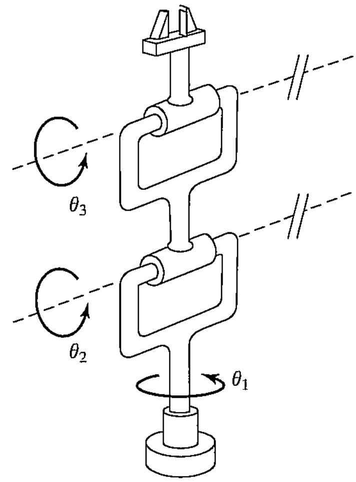

FIGURE 3.38: Three-link RRR manipulator (Exercise 3.18).

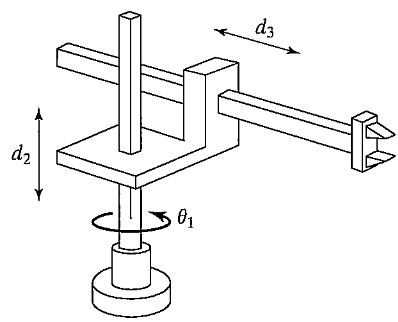

FIGURE 3.39: Three-link RPP manipulator (Exercise 3.19).

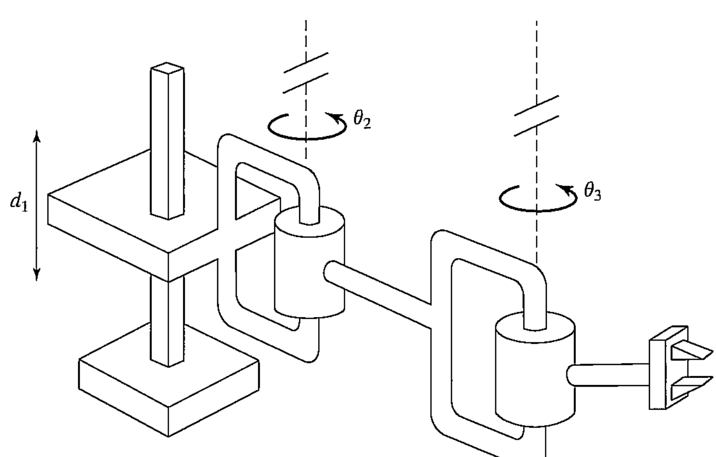

FIGURE 3.40: Three-link PRR manipulator (Exercise 3.20).

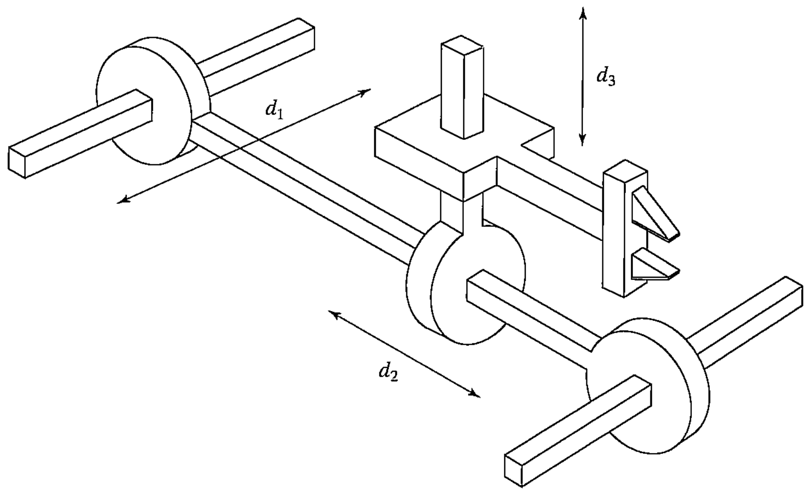

FIGURE 3.41: Three-link PPP manipulator (Exercise 3.21).

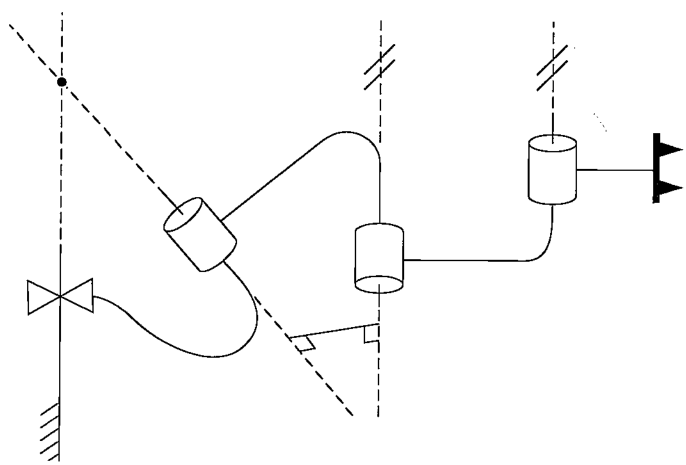

FIGURE 3.42: Schematic of a $P3R$ manipulator (Exercise 3.22).

PROGRAMMING EXERCISE (PART 3)

1. Write a subroutine to compute the kinematics of the planar $3R$ robot in Example 3.3—that is, a routine with the joint angles' values as input, and a frame (the wrist frame relative to the base frame) as output. Use the procedure heading (or equivalent in C)

```
Procedure KIN(VAR theta: vec3; VAR wrelb: frame);
```

where "wrelb" is the wrist frame relative to the base frame, ${}^B_W T$. The type "frame" consists of a 2×2 rotation matrix and a 2×1 position vector. If desired, you may represent the frame with a 3×3 homogeneous transform in which the third row is [0 0 1]. (The manipulator data are $l_1 = l_2 = 0.5$ meters.)

2. Write a routine that calculates where the tool is, relative to the station frame. The input to the routine is a vector of joint angles:

```
Procedure WHERE(VAR theta: vec3; VAR trels: frame);
```

Obviously, WHERE must make use of descriptions of the tool frame and the robot base frame in order to compute the location of the tool relative to the station frame. The values of ${}^{W}_{T}T$ and ${}^{S}_{B}T$ should be stored in global memory (or, as a second choice, you may pass them as arguments in WHERE).

3. A tool frame and a station frame for a certain task are defined as follows by the user:

$$ {}^{W}_{T}T = [x\ y\ \theta] = [0.1\ 0.2\ 30.0], $$

$$ {}^{B}_{S}T = [x\ y\ \theta] = [-0.1\ 0.3\ 0.0]. $$

Calculate the position and orientation of the tool relative to the station frame for the following three configurations (in units of degrees) of the arm:

$$ [\theta_1\ \theta_2\ \theta_3] = [0.0\ 90.0\ \ -90.0], $$

$$ [\theta_1\ \theta_2\ \theta_3] = [-23.6\ \ -30.3\ 48.0], $$

$$ [\theta_1\ \theta_2\ \theta_3] = [130.0\ 40.0\ 12.0]. $$

MATLAB EXERCISE 3

This exercise focuses on DH parameters and on the forward-pose (position and orientation) kinematics transformation for the planar 3-DOF, 3R robot (of Figures 3.6 and 3.7). The following fixed-length parameters are given: $L_1 = 4$, $L_2 = 3$, and $L_3 = 2$ (m).

a) Derive the DH parameters. You can check your results against Figure 3.8.

b) Derive the neighboring homogeneous transformation matrices ${}^{i-1}_{i}T$, $i = 1, 2, 3$. These are functions of the joint-angle variables θ_i, $i = 1, 2, 3$. Also, derive the constant ${}^{3}_{H}T$ by inspection: The origin of $\{H\}$ is in the center of the gripper fingers, and the orientation of $\{H\}$ is always the same as the orientation of $\{3\}$.

c) Use Symbolic MATLAB to derive the forward-pose kinematics solution ${}^{0}_{3}T$ and ${}^{0}_{H}T$ symbolically (as a function of θ_i). Abbreviate your answer, using $s_i = \sin(\theta_i)$, $c_i = \cos(\theta_i)$, and so on. Also, there is a $(\theta_1 + \theta_2 + \theta_3)$ simplification, by using sum-of-angle formulas, that is due to the parallel Z_i axes. Calculate the forward-pose kinematics results (both ${}^{0}_{3}T$ and ${}^{0}_{H}T$) via MATLAB for the following input cases:

i) $\Theta = \{\theta_1\ \ \theta_2\ \ \theta_3\}^T = \{0\ \ 0\ \ 0\}^T$.

ii) $\Theta = \{10^\circ\ \ 20^\circ\ \ 30^\circ\}^T$.

iii) $\Theta = \{90^\circ\ \ 90^\circ\ \ 90^\circ\}^T$.

For all three cases, check your results by sketching the manipulator configuration and deriving the forward-pose kinematics transformation by inspection. (Think of the definition of ${}^{0}_{H}T$ in terms of a rotation matrix and a position vector.) Include frames $\{H\}$, $\{3\}$, and $\{0\}$ in your sketches.

d) Check all your results by means of the Corke MATLAB Robotics Toolbox. Try functions *link*(), *robot*(), and *fkine()*.