is moderately expensive computationally, but the other solutions are found very quickly by summing and differencing angles, subtracting π, and so on.

BIBLIOGRAPHY

[1] B. Roth, J. Rastegar, and V. Scheinman, "On the Design of Computer Controlled Manipulators," *On the Theory and Practice of Robots and Manipulators*, Vol. 1, First CISM-IFToMM Symposium, September 1973, pp. 93–113.

[2] B. Roth, "Performance Evaluation of Manipulators from a Kinematic Viewpoint," *Performance Evaluation of Manipulators*, National Bureau of Standards, special publication, 1975.

[3] D. Pieper and B. Roth, "The Kinematics of Manipulators Under Computer Control," *Proceedings of the Second International Congress on Theory of Machines and Mechanisms*, Vol. 2, Zakopane, Poland, 1969, pp. 159–169.

[4] D. Pieper, "The Kinematics of Manipulators Under Computer Control," Unpublished Ph.D. Thesis, Stanford University, 1968.

[5] R.P. Paul, B. Shimano, and G. Mayer, "Kinematic Control Equations for Simple Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 6, 1981.

[6] L. Tsai and A. Morgan, "Solving the Kinematics of the Most General Six- and Five-degree-of-freedom Manipulators by Continuation Methods," Paper 84-DET-20, ASME Mechanisms Conference, Boston, October 7–10, 1984.

[7] C.S.G. Lee and M. Ziegler, "Geometric Approach in Solving Inverse Kinematics of PUMA Robots," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-20, No. 6, November 1984.

[8] W. Beyer, *CRC Standard Mathematical Tables*, 25th edition, CRC Press, Inc., Boca Raton, FL, 1980.

[9] R. Burington, *Handbook of Mathematical Tables and Formulas*, 5th edition, McGraw-Hill, New York, 1973.

[10] J. Hollerbach, "A Survey of Kinematic Calibration," in *The Robotics Review*, O. Khatib, J. Craig, and T. Lozano-Perez, Editors, MIT Press, Cambridge, MA, 1989.

[11] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 108, 1986.

[12] D. Baker and C. Wampler, "On the Inverse Kinematics of Redundant Manipulators," *International Journal of Robotics Research*, Vol. 7, No. 2, 1988.

[13] L.W. Tsai, *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*, Wiley, New York, 1999.

EXERCISES

4.1 [15] Sketch the fingertip workspace of the three-link manipulator of Chapter 3, Exercise 3.3 for the case $l_1 = 15.0$, $l_2 = 10.0$, and $l_3 = 3.0$.

4.2 [26] Derive the inverse kinematics of the three-link manipulator of Chapter 3, Exercise 3.3.

4.3 [12] Sketch the fingertip workspace of the 3-DOF manipulator of Chapter 3, Example 3.4.

4.4 [24] Derive the inverse kinematics of the 3-DOF manipulator of Chapter 3, Example 3.4.

4.5 [38] Write a Pascal (or C) subroutine that computes all possible solutions for the PUMA 560 manipulator that lie within the following joint limits:

$$-170.0 < \theta_1 < 170.0,$$
$$-225.0 < \theta_2 < 45.0,$$
$$-250.0 < \theta_3 < 75.0,$$
$$-135.0 < \theta_4 < 135.0,$$
$$-100.0 < \theta_5 < 100.0,$$
$$-180.0 < \theta_6 < 180.0.$$

Use the equations derived in Section 4.7 with these numerical values (in inches):

$$a_2 = 17.0,$$
$$a_3 = 0.8,$$
$$d_3 = 4.9,$$
$$d_4 = 17.0.$$

4.6 [15] Describe a simple algorithm for choosing the nearest solution from a set of possible solutions.

4.7 [10] Make a list of factors that might affect the repeatability of a manipulator. Make a second list of additional factors that affect the accuracy of a manipulator.

4.8 [12] Given a desired position and orientation of the hand of a three-link planar rotary-jointed manipulator, there are two possible solutions. If we add one more rotational joint (in such a way that the arm is still planar), how many solutions are there?

4.9 [26] Figure 4.13 shows a two-link planar arm with rotary joints. For this arm, the second link is half as long as the first—that is, $l_1 = 2l_2$. The joint range limits in

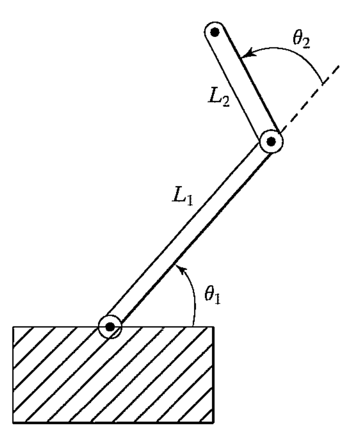

FIGURE 4.13: Two-link planar manipulator.

degrees are

$$0 < \theta_1 < 180,$$
$$-90 < \theta_2 < 180.$$

Sketch the approximate reachable workspace (an area) of the tip of link 2.

4.10 [23] Give an expression for the subspace of the manipulator of Chapter 3, Example 3.4.

4.11 [24] A 2-DOF positioning table is used to orient parts for arc-welding. The forward kinematics that locate the bed of the table (link 2) with respect to the base (link 0) are

$$ {}^0_2T = \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & l_2s_1 + l_1 \\ s_2 & c_2 & 0 & 0 \\ -s_1c_2 & s_1s_2 & c_1 & l_2c_1 + h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Given any unit direction fixed in the frame of the bed (link 2), ${}^2\hat{V}$, give the inverse-kinematic solution for θ_1, θ_2 such that this vector is aligned with ${}^0\hat{Z}$ (i.e., upward). Are there multiple solutions? Is there a singular condition for which a unique solution cannot be obtained?

4.12 [22] In Fig. 4.14, two $3R$ mechanisms are pictured. In both cases, the three axes intersect at a point (and, over all configurations, this point remains fixed in space). The mechanism in Fig. 4.14(a) has link twists (α_i) of magnitude 90 degrees. The mechanism in Fig. 4.14(b) has one twist of ϕ in magnitude and the other of $180 - \phi$ in magnitude.

The mechanism in Fig. 4.14(a) can be seen to be in correspondence with Z–Y–Z Euler angles, and therefore we know that it suffices to orient link 3 (with arrow in figure) arbitrarily with respect to the fixed link 0. Because ϕ is not equal to 90 degrees, it turns out that the other mechanism cannot orient link 3 arbitrarily.

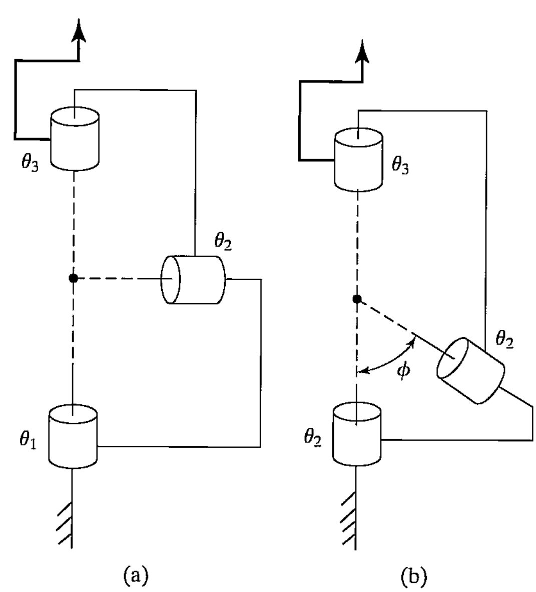

FIGURE 4.14: Two $3R$ mechanisms (Exercise 4.12).

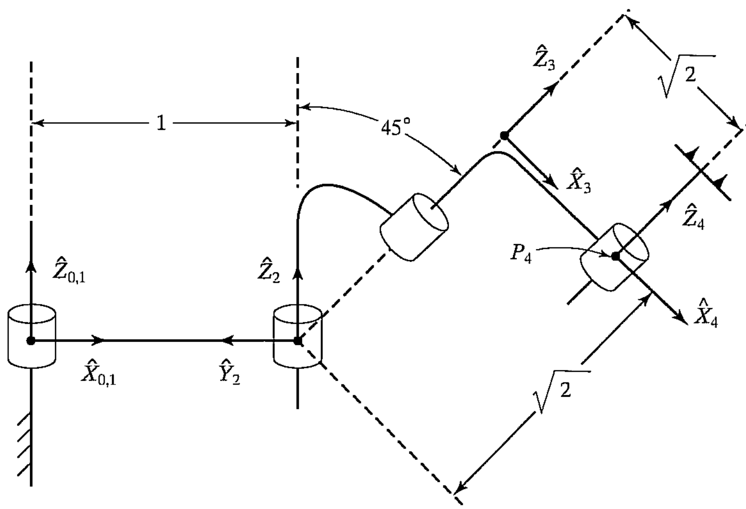

FIGURE 4.15: A $4R$ manipulator shown in the position $\Theta = [0, 90°, -90°, 0]^T$ (Exercise 4.16).

Describe the set of orientations that are *unattainable* with the second mechanism. Note that we assume that all joints can turn 360 degrees (i.e. no limits) and we assume that the links may pass through each other if need be (i.e., workspace not limited by self-collisions).

4.13 [13] Name two reasons for which closed-form analytic kinematic solutions are preferred over iterative solutions.

4.14 [14] There exist 6-DOF robots for which the kinematics are NOT closed-form solvable. Does there exist any 3-DOF robot for which the (position) kinematics are NOT closed-form solvable?

4.15 [38] Write a subroutine that solves quartic equations in closed form. (See [8, 9].)

4.16 [25] A $4R$ manipulator is shown schematically in Fig. 4.15. The nonzero link parameters are $a_1 = 1$, $\alpha_2 = 45°$, $d_3 = \sqrt{2}$, and $a_3 = \sqrt{2}$, and the mechanism is pictured in the configuration corresponding to $\Theta = [0, 90°, -90°, 0]^T$. Each joint has $\pm 180°$ as limits. Find all values of θ_3 such that

$$^{0}P_{4ORG} = [1.1, 1.5, 1.707]^T.$$

4.17 [25] A $4R$ manipulator is shown schematically in Fig. 4.16. The nonzero link parameters are $\alpha_1 = -90°$, $d_2 = 1$, $\alpha_2 = 45°$, $d_3 = 1$, and $a_3 = 1$, and the mechanism is pictured in the configuration corresponding to $\Theta = [0, 0, 90°, 0]^T$. Each joint has $\pm 180°$ as limits. Find all values of θ_3 such that

$$^{0}P_{4ORG} = [0.0, 1.0, 1.414]^T.$$

4.18 [15] Consider the RRP manipulator shown in Fig. 3.37. How many solutions do the (position) kinematic equations possess?

4.19 [15] Consider the RRR manipulator shown in Fig. 3.38. How many solutions do the (position) kinematic equations possess?

4.20 [15] Consider the RPP manipulator shown in Fig. 3.39. How many solutions do the (position) kinematic equations possess?

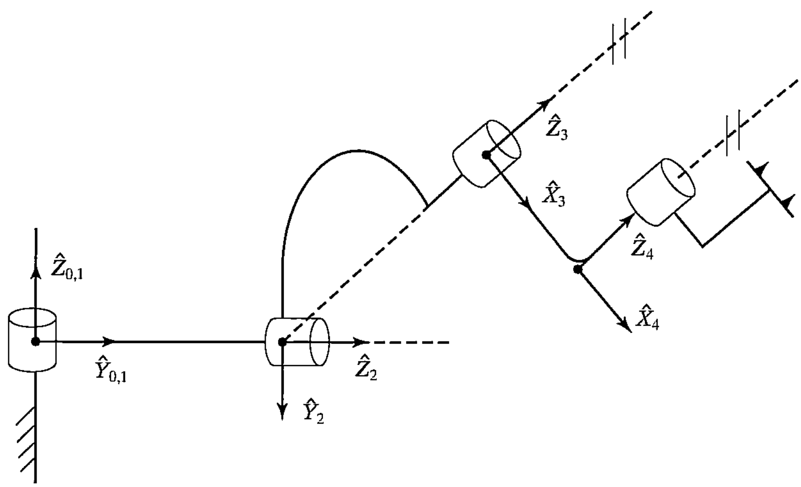

FIGURE 4.16: A $4R$ manipulator shown in the position $\Theta = [0, 0, 90°, 0]^T$ (Exercise 4.17).

4.21 [15] Consider the *PRR* manipulator shown in Fig. 3.40. How many solutions do the (position) kinematic equations possess?

4.22 [15] Consider the *PPP* manipulator shown in Fig. 3.41. How many solutions do the (position) kinematic equations possess?

4.23 [38] The following kinematic equations arise in a certain problem:

$$\sin\xi = a\sin\theta + b,$$
$$\sin\phi = c\cos\theta + d,$$
$$\psi = \xi + \phi.$$

Given a, b, c, d, and ψ, show that, in the general case, there are four solutions for θ. Give a special condition under which there are just two solutions for θ.

4.24 [20] Given the description of link frame $\{i\}$ in terms of link frame $\{i-1\}$, find the four Denavit–Hartenberg parameters as functions of the elements of ${}^{i-1}_{i}T$.

PROGRAMMING EXERCISE (PART 4)

1. Write a subroutine to calculate the inverse kinematics for the three-link manipulator of Section 4.4. The routine should pass arguments in the form

```
Procedure INVKIN(VAR wrelb: frame; VAR current, near, far: vec3;
VAR sol: boolean);
```

where "wrelb," an input, is the wrist frame specified relative to the base frame; "current," an input, is the current position of the robot (given as a vector of joint angles); "near" is the nearest solution; "far" is the second solution; and "sol" is a flag that indicates whether solutions were found. (sol = FALSE if no solutions were found). The link lengths (meters) are

$$l_1 = l_2 = 0.5.$$

The joint ranges of motion are

$$-170^\circ \le \theta_i \le 170^\circ.$$

Test your routine by calling it back-to-back with KIN to demonstrate that they are indeed inverses of one another.

2. A tool is attached to link 3 of the manipulator. This tool is described by ${}^W_T T$, the tool frame relative to the wrist frame. Also, a user has described his work area, the station frame relative to the base of the robot, as ${}^B_S T$. Write the subroutine

```
Procedure SOLVE(VAR trels: frame; VAR current, near, far: vec3;
VAR sol: boolean);
```

where "trels" is the $\{T\}$ frame specified relative to the $\{S\}$ frame. Other parameters are exactly as in the `INVKIN` subroutine. The definitions of $\{T\}$ and $\{S\}$ should be globally defined variables or constants. SOLVE should use calls to `TMULT`, `TINVERT`, and `INVKIN`.

3. Write a main program that accepts a goal frame specified in terms of x, y, and ϕ. This goal specification is $\{T\}$ relative to $\{S\}$, which is the way the user wants to specify goals.

The robot is using the same tool in the same working area as in Programming Exercise (Part 2), so $\{T\}$ and $\{S\}$ are defined as

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

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

Calculate the joint angles for each of the following three goal frames:

$$[x_1\ y_1\ \phi_1] = [0.0\ 0.0\ -90.0],$$

$$[x_2\ y_2\ \phi_2] = [0.6\ -0.3\ 45.0],$$

$$[x_3\ y_3\ \phi_3] = [-0.4\ 0.3\ 120.0],$$

$$[x_4\ y_4\ \phi_4] = [0.8\ 1.4\ 30.0].$$

Assume that the robot will start with all angles equal to 0.0 and move to these three goals in sequence. The program should find the nearest solution with respect to the previous goal point. You should call `SOLVE` and `WHERE` back-to-back to make sure they are truly inverse functions.

MATLAB EXERCISE 4

This exercise focuses on the inverse-pose kinematics solution for the planar 3-DOF, 3R robot. (See Figures 3.6 and 3.7; the DH parameters are given in Figure 3.8.) The following fixed-length parameters are given: $L_1 = 4$, $L_2 = 3$, and $L_3 = 2$(m).

a) Analytically derive, by hand, the inverse-pose solution for this robot: Given ${}^0_H T$, calculate all possible multiple solutions for $\{\theta_1\ \ \theta_2\ \ \theta_3\}$. (Three methods are presented in the text—choose one of these.) Hint: To simplify the equations, first calculate ${}^0_3 T$ from ${}^0_H T$ and L_3.

b) Develop a MATLAB program to solve this planar 3R robot inverse-pose kinematics problem completely (i.e., to give all multiple solutions). Test your program, using the following input cases:

i) $^0_H T = \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

ii) $^0_H T = \begin{bmatrix} 0.5 & -0.866 & 0 & 7.5373 \\ 0.866 & 0.6 & 0 & 3.9266 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

iii) $^0_H T = \begin{bmatrix} 0 & 1 & 0 & -3 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

iv) $^0_H T = \begin{bmatrix} 0.866 & 0.5 & 0 & -3.1245 \\ -0.5 & 0.866 & 0 & 9.1674 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

For all cases, employ a circular check to validate your results: Plug each resulting set of joint angles (for each of the multiple solutions) back into the forward-pose kinematics MATLAB program to demonstrate that you get the originally commanded $^0_H T$.

c) Check all results by means of the Corke MATLAB Robotics Toolbox. Try function *ikine()*.