[7] J. Bobrow, S. Dubowsky, and J. Gibson, "On the Optimal Control of Robotic Manipulators with Actuator Constraints," *Proceedings of the American Control Conference*, June 1983.

[8] K. Shin and N. McKay, "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints," *IEEE Transactions on Automatic Control*, June 1985.

[9] T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach," AI Memo 605, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1980.

[10] T. Lozano-Perez, "A Simple Motion Planning Algorithm for General Robot Manipulators," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 3, June 1987.

[11] R. Brooks, "Solving the Find-Path Problem by Good Representation of Free Space," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13:190–197, 1983.

[12] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, Vol. 5, No. 1, Spring 1986.

[13] R.P. Paul, "Robot Manipulators: Mathematics, Programming, and Control," MIT Press, Cambridge, MA, 1981.

[14] R. Castain and R.P. Paul, "An Online Dynamic Trajectory Generator," *The International Journal of Robotics Research*, Vol. 3, 1984.

[15] C.S. Lin and P.R. Chang, "Joint Trajectory of Mechanical Manipulators for Cartesian Path Approximation," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, 1983.

[16] C.S. Lin, P.R. Chang, and J.Y.S. Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots," *IEEE Transactions on Automatic Control*, Vol. AC-28, 1983.

[17] L. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, 12(4): 566–580, 1996.

[18] J. Barraquand, L. Kavraki, J.C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan, "A Random Sampling Scheme for Path Planning," *International Journal of Robotics Research*, 16(6): 759–774, 1997.

EXERCISES

7.1 [8] How many individual cubics are computed when a six-jointed robot moves along a cubic spline path through two via points and stops at a goal point? How many coefficients are stored to describe these cubics?

7.2 [13] A single-link robot with a rotary joint is motionless at $\theta = -5°$. It is desired to move the joint in a smooth manner to $\theta = 80°$ in 4 seconds. Find the coefficients of a cubic which accomplishes this motion and brings the arm to rest at the goal. Plot the position, velocity, and acceleration of the joint as a function of time.

7.3 [14] A single-link robot with a rotary joint is motionless at $\theta = -5°$. It is desired to move the joint in a smooth manner to $\theta = 80°$ in 4 seconds and stop smoothly. Compute the corresponding parameters of a linear trajectory with parabolic blends. Plot the position, velocity, and acceleration of the joint as a function of time.

7.4 [30] Write a path-planning software routine that implements (7.24) through (7.28) in a general way for paths described by an arbitrary number of path points. For example, this routine could be used to solve Example 7.4.

7.5 [18] Sketch graphs of position, velocity, and acceleration for the two-segment continuous-acceleration spline given in Example 7.2. Sketch them for a joint for which $\theta_0 = 5.0°, \theta_v = 15.0°, \theta_g = 40.0°$, and each segment lasts 1.0 second.

7.6 [18] Sketch graphs of position, velocity, and acceleration for a two-segment spline where each segment is a cubic, using the coefficients as given in (7.11). Sketch them for a joint where $\theta_0 = 5.0°$ for the initial point, $\theta_v = 15.0°$ is a via point, and $\theta_g = 40.0°$ is the goal point. Assume that each segment has a duration of 1.0 second and that the velocity at the via point is to be 17.5 degrees/second.

7.7 [20] Calculate $\dot{\theta}_{12}, \dot{\theta}_{23}, t_1, t_2$, and t_3 for a two-segment linear spline with parabolic blends. (Use (7.24) through (7.28).) For this joint, $\theta_1 = 5.0°, \theta_2 = 15.0°, \theta_3 = 40.0°$. Assume that $t_{d12} = t_{d23} = 1.0$ second and that the default acceleration to use during blends is 80 degrees/second^2. Sketch plots of position, velocity, and acceleration of θ.

7.8 [18] Sketch graphs of position, velocity, and acceleration for the two-segment continuous-acceleration spline given in Example 7.2. Sketch them for a joint for which $\theta_0 = 5.0°, \theta_v = 15.0°, \theta_g = -10.0°$, and each segment lasts 2.0 seconds.

7.9 [18] Sketch graphs of position, velocity, and acceleration for a two-segment spline where each segment is a cubic, using the coefficients as given in (7.11). Sketch them for a joint where $\theta_0 = 5.0°$ for the initial point, $\theta_v = 15.0°$ is a via point, and $\theta_g = -10.0°$ is the goal point. Assume that each segment has a duration of 2.0 seconds and that the velocity at the via point is to be 0.0 degrees/second.

7.10 [20] Calculate $\dot{\theta}_{12}, \dot{\theta}_{23}, t_1, t_2$, and t_3 for a two-segment linear spline with parabolic blends. (Use (7.24) through (7.28).) For this joint, $\theta_1 = 5.0°, \theta_2 = 15.0°, \theta_3 = -10.0°$. Assume that $t_{d12} = t_{d23} = 2.0$ seconds and that the default acceleration to use during blends is 60 degrees/second^2. Sketch plots of position, velocity, and acceleration of θ.

7.11 [6] Give the 6×1 Cartesian position and orientation representation ${}^S\chi_G$ that is equivalent to ${}^S_GT$ where ${}^S_GR = ROT(\hat{Z}, 30°)$ and ${}^SP_{GORG} = [10.0\ \ 20.0\ \ 30.0]^T$.

7.12 [6] Give the ${}^S_GT$ that is equivalent to the 6×1 Cartesian position and orientation representation ${}^S\chi_G = [5.0\ \ -20.0\ \ 10.0\ \ 45.0\ \ 0.0\ \ 0.0]^T$.

7.13 [30] Write a program that uses the dynamic equations from Section 6.7 (the two-link planar manipulator) to compute the time history of torques needed to move the arm along the trajectory of Exercise 7.8. What are the maximum torques required and where do they occur along the trajectory?

7.14 [32] Write a program that uses the dynamic equations from Section 6.7 (the two-link planar manipulator) to compute the time history of torques needed to move the arm along the trajectory of Exercise 7.8. Make separate plots of the joint torques required due to inertia, velocity terms, and gravity.

7.15 [22] Do Example 7.2 when $t_{f1} \neq t_{f2}$.

7.16 [25] We wish to move a single joint from θ_0 to θ_f starting from rest, ending at rest, in time t_f. The values of θ_0 and θ_f are given, but we wish to compute t_f so that $\|\dot{\theta}(t)\| < \dot{\theta}_{max}$ and $\|\ddot{\theta}(t)\| < \ddot{\theta}_{max}$ for all t, where $\dot{\theta}_{max}$ and $\ddot{\theta}_{max}$ are given positive constants. Use a single cubic segment, and give an expression for t_f and for the cubic's coefficients.

7.17 [10] A single cubic trajectory is given by

$$\theta(t) = 10 + 90t^2 - 60t^3$$

and is used over the time interval from $t = 0$ to $t = 1$. What are the starting and final positions, velocities, and accelerations?

7.18 [12] A single cubic trajectory is given by

$$\theta(t) = 10 + 90t^2 - 60t^3$$

and is used over the time interval from $t = 0$ to $t = 2$. What are the starting and final positions, velocities, and accelerations?

7.19 [13] A single cubic trajectory is given by

$$\theta(t) = 10 + 5t + 70t^2 - 45t^3$$

and is used over the time interval from $t = 0$ to $t = 1$. What are the starting and final positions, velocities, and accelerations?

7.20 [15] A single cubic trajectory is given by

$$\theta(t) = 10 + 5t + 70t^2 - 45t^3$$

and is used over the time interval from $t = 0$ to $t = 2$. What are the starting and final positions, velocities, and accelerations?

PROGRAMMING EXERCISE (PART 7)

1. Write a joint-space, cubic-splined path-planning system. One routine that your system should include is

```
Procedure CUBCOEF (VAR th0, thf, thdot0, thdotf: real; VAR cc:
vec4);
```

where

$$\text{th0} = \text{initial position of } \theta \text{ at beginning of segment,}$$
$$\text{thf} = \text{final position of } \theta \text{ at segment end,}$$
$$\text{thdot0} = \text{initial velocity of segment,}$$
$$\text{thdotf} = \text{final velocity of segment.}$$

These four quantities are inputs, and "cc", an array of the four cubic coefficients, is the output.

Your program should accept up to (at least) five via-point specifications—in the form of tool frame, $\{T\}$, relative to station frame, $\{S\}$—in the usual user form: (x, y, ϕ). To keep life simple, all segments will have the same duration. Your system should solve for the coefficients of the cubics, using some reasonable heuristic for assigning joint velocities at the via points. *Hint*: See option 2 in Section 7.3.

2. Write a path-generator system that calculates a trajectory in joint space based on sets of cubic coefficients for each segment. It must be able to generate the multisegment path you planned in Problem 1. A duration for the segments will be specified by the user. It should produce position, velocity, and acceleration information at the path-update rate, which will also be specified by the user.

3. The manipulator is the same three-link used previously. The definitions of the $\{T\}$ and $\{S\}$ frames are the same as before:

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

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

Using a duration of 3.0 seconds per segment, plan and execute the path that starts with the manipulator at position

$$[x_1\ y_1\ \phi_1] = [0.758\ \ 0.173\ \ 0.0],$$

moves through the via points

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

and

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

and ends at the goal point (in this case, same as initial point)

$$[x_4\ y_4\ \phi_4] = [0.758\ \ 0.173\ \ 0.0].$$

Use a path-update rate of 40 Hz, but print the position only every 0.2 seconds. Print the positions out in terms of Cartesian user form. You don't have to print out velocities or accelerations, though you might be interested in doing so.

MATLAB EXERCISE 7

The goal of this exercise is to implement polynomial joint-space trajectory-generation equations for a single joint. (Multiple joints would require n applications of the result.) Write a MATLAB program to implement the joint-space trajectory generation for the three cases that follow. Report your results for the specific assignments given; for each case, give the polynomial functions for the joint angle, angular velocity, angular acceleration, and angular jerk (the time derivative of acceleration). For each case, plot the results. (Arrange the plots vertically with angle, velocity, acceleration, and then jerk, all with the same time scale—check out the *subplot* MATLAB function to accomplish this.) Don't just plot out results—give some discussion; do your results make sense? Here are the three cases:

a) *Third-order polynomial.* Force the angular velocity to be zero at the start and at the finish. Given $\theta_s = 120^\circ$ (start), $\theta_f = 60^\circ$ (finish), and $t_f = 1$ sec.

b) *Fifth-order polynomial.* Force the angular velocity and acceleration to be zero at the start and at the finish. Given $\theta_s = 120^\circ, \theta_f = 60^\circ$, and $t_f = 1$ sec. Compare your results (functions and plots) with this same example, but using a single third-order polynomial, as in problem (a).

c) *Two third-order polynomials with via point.* Force the angular velocity to be zero at the start and at the finish. Don't force the angular velocity to be zero at the via point—you must match velocity and acceleration at this point for the two polynomials meeting at that point in time. Demonstrate that this condition is satisfied. Given $\theta_s = 60^\circ$ (start), $\theta_v = 120^\circ$ (via), $\theta_f = 30^\circ$ (finish), and $t_1 = t_2 = 1$ sec (relative time steps—i.e., $t_f = 2$ sec).

d) Check the results of (a) and (b) by means of the Corke MATLAB Robotics Toolbox. Try function *jtraj*().