[3] I. Shames, *Engineering Mechanics*, 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 1967.

[4] D. Orin and W. Schrader, "Efficient Jacobian Determination for Robot Manipulators," in *Robotics Research: The First International Symposium*, M. Brady and R.P. Paul, Editors, MIT Press, Cambridge, MA, 1984.

[5] B. Gorla and M. Renaud, *Robots Manipulateurs*, Cepadues-Editions, Toulouse, 1984.

[6] B. Noble, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, NJ, 1969.

[7] J.K. Salisbury and J. Craig, "Articulated Hands: Kinematic and Force Control Issues," *International Journal of Robotics Research*, Vol. 1, No. 1, Spring 1982.

[8] C. Wampler, "Wrist Singularities: Theory and Practice," in *The Robotics Review 2*, O. Khatib, J. Craig, and T. Lozano-Perez, Editors, MIT Press, Cambridge, MA, 1992.

[9] D.E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses," *IEEE Transactions on Man-Machine Systems*, 1969.

EXERCISES

5.1 [10] Repeat Example 5.3, but using the Jacobian written in frame {0}. Are the results the same as those of Example 5.3?

5.2 [25] Find the Jacobian of the manipulator with three degrees of freedom from Exercise 3 of Chapter 3. Write it in terms of a frame {4} located at the tip of the hand and having the same orientation as frame {3}.

5.3 [35] Find the Jacobian of the manipulator with three degrees of freedom from Exercise 3 of Chapter 3. Write it in terms of a frame {4} located at the tip of the hand and having the same orientation as frame {3}. Derive the Jacobian in three different ways: velocity propagation from base to tip, static force propagation from tip to base, and by direct differentiation of the kinematic equations.

5.4 [8] Prove that singularities in the force domain exist at the same configurations as singularities in the position domain.

5.5 [39] Calculate the Jacobian of the PUMA 560 in frame {6}.

5.6 [47] Is it true that any mechanism with three revolute joints and nonzero link lengths must have a locus of singular points interior to its workspace?

5.7 [7] Sketch a figure of a mechanism with three degrees of freedom whose linear velocity Jacobian is the 3×3 identity matrix over all configurations of the manipulator. Describe the kinematics in a sentence or two.

5.8 [18] General mechanisms sometimes have certain configurations, called "isotropic points," where the columns of the Jacobian become orthogonal and of equal magnitude [7]. For the two-link manipulator of Example 5.3, find out if any isotropic points exist. Hint: Is there a requirement on l_1 and l_2?

5.9 [50] Find the conditions necessary for isotropic points to exist in a general manipulator with six degrees of freedom. (See Exercise 5.8.)

5.10 [7] For the two-link manipulator of Example 5.2, give the transformation that would map joint torques into a 2×1 force vector, ${}^3F$, at the hand.

5.11 [14] Given

$$
{}^A_BT = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 0.0 \\ 0.000 & 0.000 & 1.000 & 5.0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

if the velocity vector at the origin of $\{A\}$ is

$$^{A}\nu = \begin{bmatrix} 0.0 \\ 2.0 \\ -3.0 \\ 1.414 \\ 1.414 \\ 0.0 \end{bmatrix},$$

find the 6×1 velocity vector with reference point the origin of $\{B\}$.

5.12 [15] For the three-link manipulator of Exercise 3.3, give a set of joint angles for which the manipulator is at a workspace-boundary singularity and another set of angles for which the manipulator is at a workspace-interior singularity.

5.13 [9] A certain two-link manipulator has the following Jacobian:

$$^{0}J(\Theta) = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}.$$

Ignoring gravity, what are the joint torques required in order that the manipulator will apply a static force vector $^{0}F = 10\hat{X}_0$?

5.14 [18] If the link parameter a_3 of the PUMA 560 were zero, a workspace-boundary singularity would occur when $\theta_3 = -90.0°$. Give an expression for the value of θ_3 where the singularity occurs, and show that, if a_3 were zero, the result would be $\theta_3 = -90.0°$. *Hint*: In this configuration, a straight line passes through joint axes 2 and 3 and the point where axes 4, 5, and 6 intersect.

5.15 [24] Give the 3×3 Jacobian that calculates linear velocity of the tool tip from the three joint rates for the manipulator of Example 3.4 in Chapter 3. Give the Jacobian in frame $\{0\}$.

5.16 [20] A $3R$ manipulator has kinematics that correspond exactly to the set of Z–Y–Z Euler angles (i.e., the forward kinematics are given by (2.72) with $\alpha = \theta_1$, $\beta = \theta_2$, and $\gamma = \theta_3$). Give the Jacobian relating joint velocities to the angular velocity of the final link.

5.17 [31] Imagine that, for a general 6-DOF robot, we have available $^{0}\hat{Z}_i$ and $^{0}P_{i\,org}$ for all i—that is, we know the values for the unit Z vectors of each link frame in terms of the base frame and we know the locations of the origins of all link frames in terms of the base frame. Let us also say that we are interested in the velocity of the tool point (fixed relative to link n) and that we know $^{0}P_{tool}$ also. Now, for a revolute joint, the velocity of the tool tip due to the velocity of joint i is given by

$$^{0}v_i = \dot{\theta}_i \, {}^{0}\hat{Z}_i \times ({}^{0}P_{tool} - {}^{0}P_{i\,org}) \tag{5.110}$$

and the angular velocity of link n due to the velocity of this joint is given by

$$^{0}\omega_i = \dot{\theta}_i \, {}^{0}\hat{Z}_i. \tag{5.111}$$

The total linear and angular velocity of the tool is given by the sum of the $^{0}v_i$ and $^{0}\omega_i$ respectively. Give equations analogous to (5.110) and (5.111) for the case of joint i prismatic, and write the 6×6 Jacobian matrix of an arbitrary 6-DOF manipulator in terms of the $\hat{Z}_i$, $P_{i\,org}$, and P_{tool}.

5.18 [18] The kinematics of a $3R$ robot are given by

$$^{0}_{3}T = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & l_1 c_1 + l_2 c_1 c_2 \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & l_1 s_1 + l_2 s_1 c_2 \\ s_{23} & c_{23} & 0 & l_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Find ${}^0J(\Theta)$, which, when multiplied by the joint velocity vector, gives the linear velocity of the origin of frame {3} relative to frame {0}.

5.19 [15] The position of the origin of link 2 for an *RP* manipulator is given by

$$ {}^0P_{2ORG} = \begin{bmatrix} a_1c_1 - d_2s_1 \\ a_1s_1 + d_2c_1 \\ 0 \end{bmatrix}. $$

Give the 2×2 Jacobian that relates the two joint rates to the linear velocity of the origin of frame {2}. Give a value of Θ where the device is at a singularity.

5.20 [20] Explain what might be meant by the statement: "An n-DOF manipulator at a singularity can be treated as a redundant manipulator in a space of dimensionality $n-1$."

PROGRAMMING EXERCISE (PART 5)

1. Two frames, {A} and {B}, are not moving relative to one another—that is, ${}^A_BT$ is constant. In the planar case, we define the velocity of frame {A} as

$$ {}^A\nu_A = \begin{bmatrix} {}^A\dot{x}_A \\ {}^A\dot{y}_A \\ {}^A\dot{\theta}_A \end{bmatrix}. $$

Write a routine that, given ${}^A_BT$ and ${}^A\nu_A$, computes ${}^B\nu_B$. *Hint:* This is the planar analog of (5.100). Use a procedure heading something like (or equivalent C):

```
Procedure Veltrans (VAR brela: frame; VAR vrela, vrelb: vec3);
```

where "vrela" is the velocity relative to frame {A}, or ${}^A\nu_A$, and "vrelb" is the output of the routine (the velocity relative to frame {B}), or ${}^B\nu_B$.

2. Determine the 3×3 Jacobian of the three-link planar manipulator (from Example 3.3). In order to derive the Jacobian, you should use velocity-propagation analysis (as in Example 5.2) or static-force analysis (as in Example 5.6). Hand in your work showing how you derived the Jacobian.

Write a routine to compute the Jacobian in frame {3}—that is, ${}^3J(\Theta)$—as a function of the joint angles. Note that frame {3} is the standard link frame with origin on the axis of joint 3. Use a procedure heading something like (or equivalent C):

```
Procedure Jacobian (VAR theta: vec3; Var Jac: mat33);
```

The manipulator data are $l_2 = l_2 = 0.5$ meters.

3. A tool frame and a station frame are defined as follows by the user for a certain task (units are meters and degrees):

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

$$ {}^B_ST = [x\ y\ \theta] = [0.0\ 0.0\ 0.0]. $$

At a certain instant, the tool tip is at the position

$$ {}^S_TT = [x\ y\ \theta] = [0.6\ \ -0.3\ 45.0]. $$

At the same instant, the joint rates (in deg/sec) are measured to be

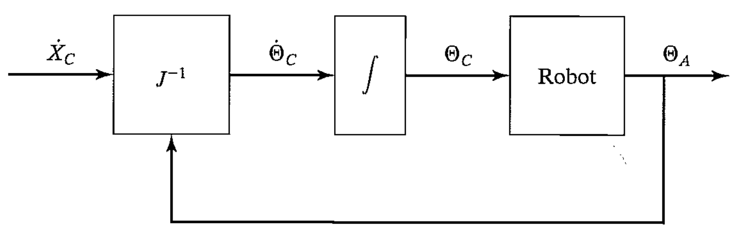
$$\dot{\Theta} = [\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3] = [20.0\ \ -10.0\ 12.0].$$

Calculate the linear and angular velocity of the tool tip relative to its own frame, that is, ${}^{T}\nu_{T}$. If there is more than one possible answer, calculate all possible answers.

MATLAB EXERCISE 5

This exercise focuses on the Jacobian matrix and determinant, simulated resolved-rate control, and inverse statics for the planar 3-DOF, 3R robot. (See Figures 3.6 and 3.7; the DH parameters are given in Figure 3.8.)

The resolved-rate control method [9] is based on the manipulator velocity equation ${}^{k}\dot{X} = {}^{k}J\dot{\Theta}$, where ${}^{k}J$ is the Jacobian matrix, $\dot{\Theta}$ is the vector of relative joint rates, ${}^{k}\dot{X}$ is the vector of commanded Cartesian velocities (both translational and rotational), and k is the frame of expression for the Jacobian matrix and Cartesian velocities. This figure shows a block diagram for simulating the resolved-rate control algorithm:



Resolved-Rate-Algorithm Block Diagram

As is seen in the figure, the resolved-rate algorithm calculates the required commanded joint rates $\dot{\Theta}_C$ to provide the commanded Cartesian velocities $\dot{X}_C$; this diagram must be calculated at every simulated time step. The Jacobian matrix changes with configuration Θ_A. For simulation purposes, assume that the commanded joint angles Θ_C are always identical to the actual joint angles achieved, Θ_A (a result rarely true in the real world). For the planar 3-DOF, 3R robot assigned, the velocity equations ${}^{k}\dot{X} = {}^{k}J\dot{\Theta}$ for $k = 0$ are

$${}^{0}\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \omega_z \end{Bmatrix} = {}^{0}\begin{bmatrix} -L_1s_1 - L_2s_{12} - L_3s_{123} & -L_2s_{12} - L_3s_{123} & -L_3s_{123} \\ L_1c_1 + L_2c_{12} + L_3c_{123} & L_2c_{12} + L_3c_{123} & L_3c_{123} \\ 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix},$$

where $s_{123} = \sin(\theta_1 + \theta_2 + \theta_3)$, $c_{123} = \cos(\theta_1 + \theta_2 + \theta_3)$, and so on. Note that ${}^{0}\dot{X}$ gives the Cartesian velocities of the origin of the hand frame (at the center of the grippers in Figure 3.6) with respect to the origin of the base frame {0}, expressed in {0} coordinates.

Now, most industrial robots cannot command $\dot{\Theta}_C$ directly, so we must first integrate these commanded relative joint rates to commanded joint angles Θ_C, which can be commanded to the robot at every time step. In practice, the simplest possible integration scheme works well, assuming a small control time step Δt: $\Theta_{new} = \Theta_{old} + \dot{\Theta}\Delta t$. In your MATLAB resolved-rate simulation, assume that the commanded Θ_{new} can be achieved perfectly by the virtual robot. (Chapters 6 and 9 present dynamics and control material for which we do not have to make this simplifying assumption.) Be sure to update the

Jacobian matrix with the new configuration Θ_{new} before completing the resolved-rate calculations for the next time step.

Develop a MATLAB program to calculate the Jacobian matrix and to simulate resolved-rate control for the planar 3R robot. Given the robot lengths $L_1 = 4$, $L_2 = 3$, and $L_3 = 2$ (m); the initial joint angles $\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T = \{10^\circ \ 20^\circ \ 30^\circ\}^T$, and the constant commanded Cartesian rates ${}^0\{\dot{X}\} = \{\dot{x} \ \dot{y} \ \omega_z\}^T = \{0.2 \ -0.3 \ -0.2\}^T$ (m/s, m/s, rad/s), simulate for exactly 5 sec, using time steps of exactly $dt = 0.1$ sec. In the same program loop, calculate the inverse-statics problem—that is, calculate the joint torques $\mathrm{T} = \{\tau_1 \ \tau_2 \ \tau_3\}^T$ (Nm), given the constant commanded Cartesian wrench ${}^0\{W\} = \{f_x \ f_y \ m_z\}^T = \{1 \ 2 \ 3\}^T$ (N, N, Nm). Also, in the same loop, animate the robot to the screen during each time step, so that you can watch the simulated motion to verify that it is correct.

a) For the specific numbers assigned, present five plots (each set on a separate graph, please):

1. the three active joint rates $\dot{\Theta} = \{\dot{\theta}_1 \ \dot{\theta}_2 \ \dot{\theta}_3\}^T$ vs. time;
2. the three active joint angles $\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T$ vs. time;
3. the three Cartesian components of ${}^0_H T$, $X = \{x \ y \ \phi\}^T$ (rad is fine for ϕ so that it will fit) vs. time;
4. the Jacobian matrix determinant $|J|$ vs. time—comment on nearness to singularities during the simulated resolved-rate motion;
5. the three active joint torques $\mathrm{T} = \{\tau_1 \ \tau_2 \ \tau_3\}^T$ vs. time.

Carefully label (by hand is fine!) each component on each plot; also, label the axes with names and units.

b) Check your Jacobian matrix results for the initial and final joint-angle sets by means of the Corke MATLAB Robotics Toolbox. Try function *jacob0()*. **Caution**: The toolbox Jacobian functions are for motion of {3} with respect to {0}, not for $\{H\}$ with respect to {0} as in the problem assignment. The preceding function gives the Jacobian result in {0} coordinates; *jacobn()* would give results in {3} coordinates.