9

Robotics, Dynamics, and Kinematics

After all the electronic sensing, signal processing, and computing have been put into effect, most applications must result in some mechanical movement. We might be required to look at the theory of coordinating the axes of a robot to put the workpiece in the correct position or more simply to choose a motor and gearbox to move a load at a safe top speed.

9.1 GEARS, MOTORS, AND MECHANISMS

Electricity is powerful stuff. It is quite easy to relate electrical power to mechanical power in metric units, although pounds and feet will require a lot of conversion factors.

Consider the following:

$$\begin{aligned} 1\text{ kilogram force} &= 9.81\text{ newtons} \\ &\quad (\text{so a force of one newton is about the weight of an apple}) \\ 1\text{ joule} &= 1\text{ newton-meter} \\ 1\text{ watt} &= 1\text{ joule per second} \end{aligned}$$

So a one-watt motor, if it were 100% efficient, could lift a one-kilogram mass at a rate of 10 centimeters per second. To lift a 75-kilogram passenger in an elevator at one meter per second will require

Essentials of Mechatronics, by John Billingsley

$$75 \times 9.81 \times 1\,\mathrm{W}$$

Now the motor may be only 50% efficient, so provision must be made for 1.5 kilowatts per passenger, plus a lot more for accelerating the cage.

In selecting a motor for a mechatronic task, it is important to allow for sufficient power. But it is also important not to provide excessive power, force, or speed.

Some time in the 1950s, the autopilot of a passenger aircraft decided that the aircraft should plunge vertically. Not surprisingly, the pilot disagreed, but could not disengage the autopilot. The resulting tug-of-war came to an end when the geartrain of the autopilot broke. Ever since, autopilots have been designed with a *shear link*, a sort of mechanical fuse, so that the possible disaster can ultimately be blamed on pilot error.

9.1.1 Calculating Motor Performance

A typical small motor might have a top speed of some 6000–12,000 revolutions per minute, that is, 100–200 revolutions per second. How can we convert this rotary motion into a linear motion of, say, 1 meter per second?

A pulley to match this speed would have to have an effective circumference of between 5 and 10 millimeters—much smaller than practical. With a reduction gear of ratio 30:1, however, the pulley could be between 50 and 100 mm in diameter (remember that π is involved).

There are a number of parameters that will define the motor: the resistance, the stall torque, the no-load speed, the moment of inertia, the rated voltage, and the rated power. We should also consider the starting torque.

When the motor rotates, it generates a back-emf—indeed, any good motor can be used as a generator. There is an important coefficient that we will call k_V, where, if we neglect the starting torque

$$k_V = V_{\text{rated}}\,60 / (2\pi\,\text{rpm}_{\text{no load}})$$

The generated voltage will be

$$V_{\text{gen}} = k_V\,\omega$$

where ω is the angular velocity of the rotation. You will note that k_V has been calculated to make the generated voltage equal to the rated voltage at the no-load speed.

A good permanent-magnet DC motor will have a small starting torque and corresponding small starting voltage. If allowed to run freely, it will take little current since it will run at such a speed that the generated back-emf almost equals the supply voltage. If a load is applied to the motor, it will slow down, the back-emf will drop, and the current will increase accordingly until the drive torque is equal to the load torque. That leads us to another important parameter, k_T, such that

$$\text{Torque} = k_T i$$

where i is the current in the motor. We can calculate k_T from the resistance R and the rated stall torque by

$$k_T = \text{stall torque } R/V_{\text{rated}}$$

Under load and at steady speed, the output power is the product of the torque and the angular velocity, so it is given by

$$\begin{aligned} &k_T i\omega \\ &\quad = k_T \omega (V - k_V \omega)/R \end{aligned}$$

When the motor runs free, the output power is zero; when the motor is stalled, the output power is also zero. Maximum mechanical power is obtained at half the no-load speed, when the back-emf V_{gen} will be $V/2$.

At any speed, the power dissipated in the motor as heat is $i^2 R$, while the power taken from the supply is iV. But

$$V = V_{\text{gen}} + iR$$

so the mechanical power output, equal to supply power minus dissipation, is

$$\begin{aligned} &= V_{\text{gen}}\, i \\ &= k_V \omega\, i \end{aligned}$$

But above we saw that this power was

$$k_T \omega\, i$$

so

$$k_T = k_V$$

So we can simply call these two parameters, which are actually the same parameter, k. The voltage generated per radian per second is equal to the meter-newtons of torque per ampere of current.

We also see that at half no-load speed, the output mechanical power is equal to the dissipated heat.

9.1.2 The Effect of an Inertial Load

Now we can set up a differential equation for the motor, when driven with no load from voltage V:

$$
\begin{aligned}
I\frac{d\omega}{dt} &= ki \\
&= k\frac{V-k\omega}{R}
\end{aligned}
$$

The motor will accelerate up to its steady speed with time constant $I\,R/k^2$.

When we add an inertial load of mass M, it will increase the effective moment of inertia to $I + M\,r^2$, where r is the "effective pulley," the distance moved by the mass for each radian of motor rotation. This takes any gearbox into consideration.

The maximum acceleration from rest is

$$
\begin{aligned}
&r\frac{d\omega}{dt} \\
&= r\frac{kV/R}{I+Mr^2}
\end{aligned}
$$

which will be greatest if the motor, gearbox, pulley, and mass are "matched" so that

$$Mr^2 = I$$

Of course, maximizing the acceleration may not be the most important objective. There may be a standing force on the mass, for example, if the mass moves vertically or if it is part of a machine with a cutting force. If the motor must withstand a disturbance torque at rest, the power taken from the supply will correspond to that torque acting at the motor's top speed. And all that power will be dissipated as heat in the motor.

It may, therefore, be desirable to increase the gear ratio, thereby decreasing the effective pulley, to obtain a compromise between standing torque and peak acceleration. If the gear ratio is doubled, for example, the standing torque is halved while the peak acceleration is reduced from its optimum by only 20%.

The gear ratio can be multiplied by 3.7 before the peak acceleration is halved, although this will reduce the top speed by the same factor of 3.7. Good design is always a matter of compromise.

9.1.3 Mechanisms

When we wish to convert the rotation of a motor to the motion of a load, a pulley is merely one very simple example of a mechanism to use. The choice will often have very little to do with dynamics.

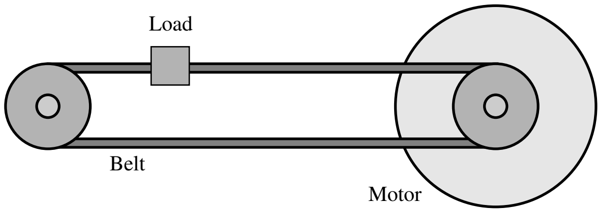

Figure 9.1 *Pulley and belt.*

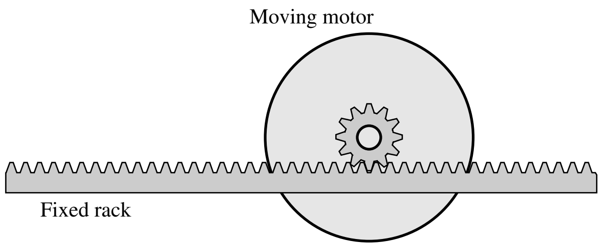

Figure 9.2 *Rack and pinion.*

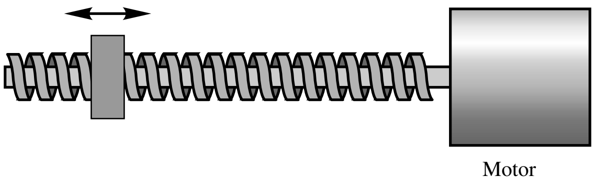

Figure 9.3 *Lead screw.*

A pulley-and-belt system (Fig. 9.1) has the advantage of simplicity, but has other drawbacks. In its simple form there is the risk of slip, so that there is an error between motion at the motor and motion of the load. This can be avoided with a "toothed belt"—although there is still the issue of stretching of the belt.

A more robust mechanism might appear to be the rack-and-pinion system (Fig. 9.2). A gear on the motor or its gearbox now runs on a rack, or linear gear, running the length of the travel required. This has some penalties of cost, but a greater drawback is that the motor now travels with the mass as part of the load.

Machine tools favor the lead screw (Fig. 9.3), a rod with square-cut threads running the length of the slideway. On one hand, mechanical efficiency is poor; on the other hand, it is insensitive to disturbing forces. It is also likely to suffer from "backlash."

In any system with "teeth," particularly a gearbox, the problem of backlash requires attention. As the motor rotates, the load is pushed along. When the motor stops and reverses, it must rotate a little way before the "other side" of the tooth engages to push the load the other way. There are several remedies.

An "antibacklash" gearbox can be installed. This is, in effect, two gearboxes working in parallel. A spring ensures that one gear pushes the output shaft hard up against the other gear. If enough torque is applied, the backlash is still there.

The same sort of effect occurs if the axis is vertical, so that the gearbox "holds the load up" and contact is always made on the same face of the gear.

In a rack-and-pinion system, the pinion can be sprung against the rack.

In a machine tool, care is usually taken to approach a setting from the same direction, as when a lathe traverse is moved to take a deeper cut.

This is a good point to mention a significant aspect of control theory.

> If we attempt to close a control loop around a backlash element, we will have problems. On reaching the target, the controller is likely to oscillate in a limit cycle as it attempts to nudge the load on either side of zero error. We can include a velocity term measured at the motor, but this might merely convert the dithering to a slower twitch.
>
> Alternatively, we can concentrate on controlling the motor position alone. The control problem will be much simpler, but now we might have an error in the load position equal to the backlash. Elasticity in a drive belt can pose a similar dilemma.

Of course, the load might not be constrained to move in a straight line. The whole appeal of the revolute robot is that arms rotate about pivots at the joints, where any straight lines are the result of cunning coordination of axes working in unison. Other devices rely on mechanisms such as the four-bar linkage that can result in rotation about a "virtual pivot."

Gearbox design is an art in itself. As well as conventional gears, there are worm drives, harmonic drives, sun and planet mechanisms, and many more. When the relationship between motor speed and load speed is to be nonlinear, there are solutions that include elliptic gears. Yet, however complicated the mechanism, we can apply the principle of virtual work.

The product of load force multiplied by the distance that it moves must, ignoring friction losses, be equal to the product of motor torque multiplied by the angle through which it rotates.

9.2 THREE-DIMENSIONAL MOTION

A point P in space is defined by a three-dimensional vector, but the method employed to represent it is not unique. The most obvious form is Cartesian (Fig. 9.4a), in which, the three coordinates are found by resolving the vector

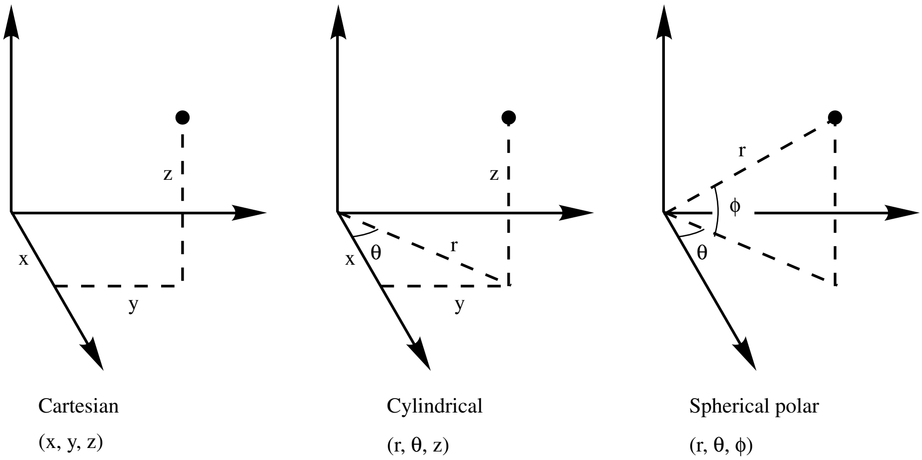

Figure 9.4 *(a) Cartesian (x,y,z), (b) cylindrical (r,θ,z), and (c) spherical polar (r,θ,ϕ) coordinates.*

from the origin in the directions of three orthogonal vectors through that origin. There are also spherical polar (Fig. 9.4c) coordinates, equivalent to defining the latitude, longitude, and distance of the point from the origin, also cylindrical polar (Fig. 9.4b), in which the point is represented by radius, direction, and height.

Not only is the location of the origin a matter of choice, we can orient the orthogonal vectors of Cartesian coordinates with 3 more degrees of freedom.

For now, however, let us take it that the origin is fixed and that we have three unit vectors $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ defining the x, y, and z directions.

As we saw in Chapter 7, our point P can be represented as $(x,y,z)'$, meaning

$$x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}$$

When the point moves, x, y, and z will vary as functions of time. Now we can take the derivatives of the vector components to calculate the velocity and acceleration. It is worth making a few remarks about these.

As it moves, P will follow a curve in space (see Fig. 9.5). The velocity vector will be a tangent to this curve at P. The acceleration can be broken into two perpendicular components. One of these is in the same direction as the velocity, representing a change in speed, while the other is perpendicular to the path, aligned through the instantaneous center of rotation, the center of curvature of the path at that point.

This may seem too simple—and it is. When we start to analyze the motion of a robot, we must deal with six dimensions, not just three. We are concerned with solid bodies, not mere points in space. We have three dimensions of

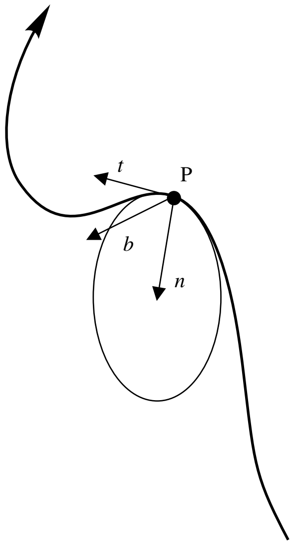

Figure 9.5 *Center of curvature.*

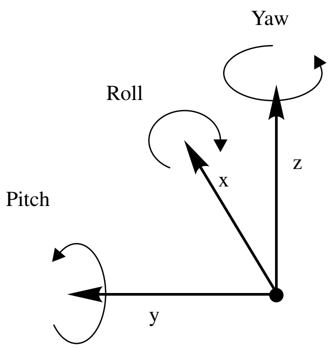

Figure 9.6 *Schematic representation of 6 degrees of freedom.*

freedom in the location of one particular point of the object, but then we can perform three rotations to orient the object in space. We might think of these rotations as movement about the pitch, roll, and yaw axes of an aircraft (see Fig. 9.6).

Instead of the vector coordinates of just one of its points, we have to think of the position and orientation of the object as being defined by the transformation that maps each of its points to the new position that it takes up. Let us first consider the transformation of rotation.

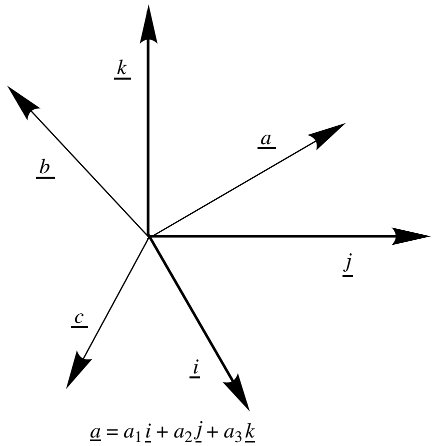

Figure 9.7 *Unit vectors.*

9.2.1 Rotations

We can set up a coordinate system of three orthogonal axes in the object. To start with, these will coincide with our "reference system" axes $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ that stay fixed. But as we rotate the object about the origin, its axes will move to be three other orthogonal vectors through the origin.

Let us consider three such unit vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, passing through the origin of our coordinate system and orthogonal to each other (Fig. 9.7).

A point expressed in terms of these vectors as coordinates $(x,y,z)'$ will be

$$\boldsymbol{a}x + \boldsymbol{b}y + \boldsymbol{c}z$$

This can be expanded as

$$[\boldsymbol{i} \quad \boldsymbol{j} \quad \boldsymbol{k}]\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

to give the coordinates of the same point in terms of the reference system.

We transform the coordinates to the reference axes by multiplying $(x,y,z)'$ by this matrix A. So, let us look at some of the properties of A.

Since they are unit vectors, $\boldsymbol{a} \cdot \boldsymbol{a} = 1$, $\boldsymbol{b} \cdot \boldsymbol{b} = 1$ and $\boldsymbol{c} \cdot \boldsymbol{c} = 1$. Also, since the vectors are orthogonal, the scalar product of any two different vectors is zero, for example, $\boldsymbol{a} \cdot \boldsymbol{b} = 0$.

Let us consider the product of A with its transpose:

$$A'A = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix}$$

Remember the "scalar products" way to look at matrix multiplication. We see that

$$A'A = \begin{bmatrix} \boldsymbol{a} \cdot \boldsymbol{a} & \boldsymbol{a} \cdot \boldsymbol{b} & \boldsymbol{a} \cdot \boldsymbol{c} \\ \boldsymbol{b} \cdot \boldsymbol{a} & \boldsymbol{b} \cdot \boldsymbol{b} & \boldsymbol{b} \cdot \boldsymbol{c} \\ \boldsymbol{c} \cdot \boldsymbol{a} & \boldsymbol{c} \cdot \boldsymbol{b} & \boldsymbol{c} \cdot \boldsymbol{c} \end{bmatrix}$$

But from what we know of these scalar products

$$A'A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So

$$A'A = I$$

or

$$A' = A^{-1}$$

The rotation transformation matrix is extremely easy to invert!

There is a further property that we have to preserve; the axes must make up a "righthanded" set. The conventional set of axes will be $\boldsymbol{i}$ and $\boldsymbol{j}$, as we draw x and y on a horizontal sheet of graph paper, and $\boldsymbol{k}$ vertically upward in the z direction.

Because it reverses the x coordinate, the matrix

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

would map a lefthanded glove into a righthanded glove, something no rotation could do. Yet it satisfies the property of having three mutually orthogonal unit vectors as its rows and its columns. What is wrong?

A property of a rotation is that there is an axis about which the rotation takes place. Now, if a vector $\boldsymbol{\xi}$ is aligned with this axis, it is not changed by being transformed by A; in other words

$$A\boldsymbol{\xi} = \boldsymbol{\xi}$$

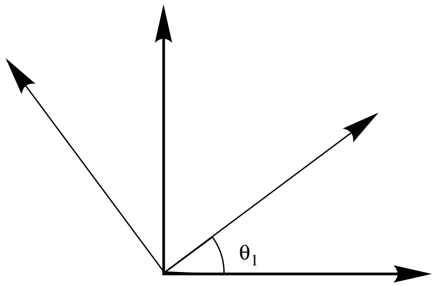

Figure 9.8 *Rotation about z axis.*

So $\boldsymbol{\xi}$ is an eigenvector of A, with eigenvalue 1. All the eigenvalues of a rotation must be 1, so the determinant of A must be 1. The determinant of the glove-bending matrix is –1, so it cannot represent a rotation.

We should look at some examples of rotation matrices. If we rotate the x–y plane by an angle θ_1 about the z axis (Fig. 9.8), we get new coordinates:

$$(x\cos\theta_1 - y\sin\theta_1, x\sin\theta_1 + y\cos\theta_1, z)$$

The z component stays the same.

In matrix terms, the transformation is

$$\begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A rotation θ_2 about the y axis would be represented by

$$\begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix}$$

Note that a positive rotation is "clockwise looking out along the axis," so this tips the x axis downward.

If we multiply the matrices together to get the result of applying both transformations, we will start to build up a string of sines and cosines that will be lengthy to write and muddling to read. Therefore, we use considerable abbreviation and write $\cos\theta_1$ as c_1, $\sin\theta_1$ as s_1, and so on. If we apply these in order, the transformed coordinates will be

$$\begin{bmatrix} c_2 & 0 & s_2 \\ 0 & 1 & 0 \\ -s_2 & 0 & c_2 \end{bmatrix}\begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Note that the transformation that is applied first is closest to the vector; in other words, the matrices are ordered right to left. Note, too, that the order is important and must not be changed. Here the result is

$$\begin{bmatrix} c_1c_2 & -s_1c_2 & s_2 \\ s_1 & c_1 & 0 \\ -s_2c_1 & s_1s_2 & c_2 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Check that the columns are unit vectors that are mutually orthogonal.

See rotations in action at www.essmech.com/9/2/1.htm

So far we have been considering transformations that leave the origin fixed, but we must also be able to move the coordinates anywhere in three dimensions.

9.2.2 Translations

To move an object a vector distance, we simply add that vector to every one of its points.

For example, a point $(x,y,z)'$ can be moved a vector distance $(1,2,3)'$ to arrive at $(x + 1, y + 2, z + 3)'$—it's not really difficult! To find the new vector, we simply add the displacement to it.

The problem is that we now have two different processes for dealing with the two types of movement: rotation and translation. One involves multiplying the point coordinates by a 3×3 matrix, while the other involves adding constants to each component. Can we find some way of gluing them together into a single operation? If we can, we can start to deal with combinations of transformations, such as "screwing" where the object is rotated at the same time as it is moved along the rotation axis.

We have to appease the mathematicians! Rotation is a transformation given by a simple multiplication of a vector by a matrix, but the ability to add a constant to the result requires an *affine* transformation.

However, there is a way around the problem. Suppose that instead of writing our vector as $(x,y,z)'$ we write it as $(x,y,z,1)'$.

What is the 1 for? It gives something for a matrix to grab onto to add a translation d to the vector! But now the vector has four components, and the matrix is 4×4.

We can "partition" a matrix to see its various parts in action, so if we write $T\,\boldsymbol{x}$ for the product of our point with a transformation matrix, now 4×4, we can break it down as follows.

$$\begin{bmatrix} A & \boldsymbol{d} \\ 0\,0\,0 & 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}=\begin{bmatrix} A\boldsymbol{x}+\boldsymbol{d} \\ 1 \end{bmatrix}$$

Thus, at the expense of changing our matrices to 4×4, where the bottom row is always $(0,0,0,1)$, we can apply any combination of rotations and translations, just by multiplying the T matrices together.

This transformation is called the Denavit–Hartenberg (or D–H) matrix.

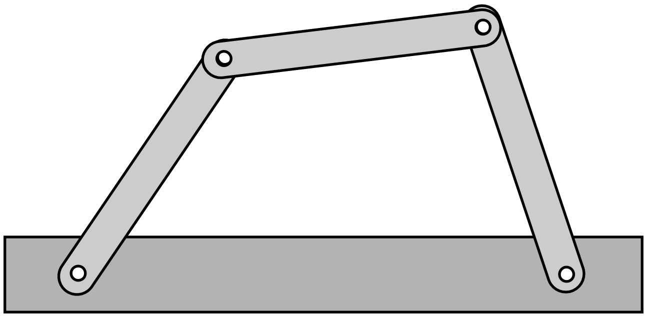

Figure 9.9 Four-bar linkage (sometimes called three-bar).

9.3 KINEMATIC CHAINS

The most usual form for a robot is a chain of links with actuated joints between them. These joints can be *revolute*, a sort of powered hinge, or *prismatic*, with one member sliding past another. We will refer to both types as *axes*. Although some kinematic chains can be "closed," such as the four-bar linkage of Figure 9.9, most robots are "open" where only one end of the chain is fixed.

When we consider the toolpiece of a robot, its location in space has been transformed by the motion of every axis in turn that moves it. Before we can address the task of deciding on joint angles or displacements to put the tool where we want it, we have to derive an expression for its location and orientation in terms of the joint axis variables.

9.3.1 Chains of Axes

When we have just one movable axis, there is a single transformation and all is straightforward. When we have a robot such as the *Unimation Puma*, with 6 degrees of freedom, we have to be systematic in the way that we analyze it.

Let us start with $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ as the usual x,y,z axes fixed in the mounting of the robot and call them *frame 0*. We need to know the transformation that will convert the coordinates of anything held in the gripper into coordinates with respect to the reference frame 0 in the robot's base.

We can define a succession of frames as we make our way along the robot to the gripper. Each of these frames will have a local x, y, and z direction related by some transformation to the next frame. Some transformations will relate to the variable angles that make up the axes; others will simply take us from one end to the other of a link such as the "forearm."

We can choose the frames so that the transformations between them are extremely simple, involving either a rotation about one of the axes or a translation along one of the axes.

Let us see this in action (Fig. 9.10).

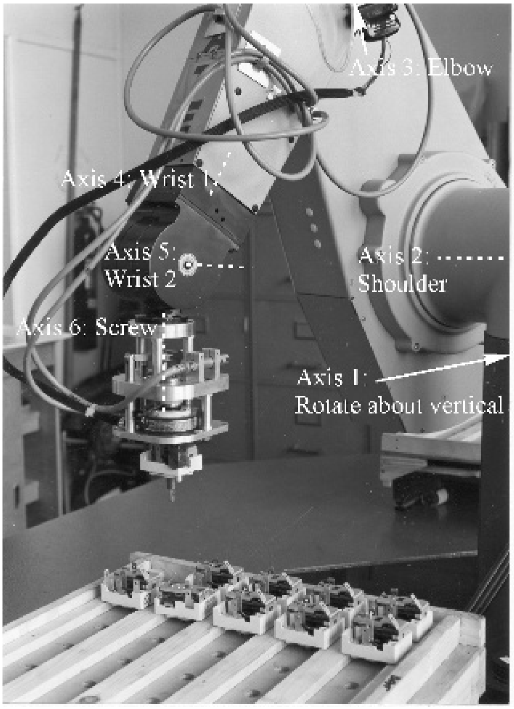

Figure 9.10 *Axes of a Unimation Puma.*

The joints of the Puma can be thought of as mimicking the human body. The first joint is a "waist joint" that rotates the whole of the rest of the robot about a vertical axis.

Then, mounted a little to one side, is a simplified "shoulder joint." This allows the upper arm to rotate about a horizontal axis extending from the "shoulder."

Next we have a simplified "elbow joint," also allowing rotation about a horizontal axis parallel to that of the shoulder.

Then we have three wrist joints to which it is difficult to assign names. The first allows rotation about the line of the forearm, as you would use when turning a door handle. The second is a hinge perpendicular to this, such as you might use when petting a dog. The third is a twist, rather like a screwdriver held between fingers and thumb.

We need to define a chain of frames all the way from frame 0 to the gripper. For our first "journey," let us simply climb up the shaft of the robot to the height of the shoulder, where we will put frame 1. The transformation ${}^0_1T$ will convert frame 1 coordinates to frame 0 coordinates, and so will be

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where h is the height of the shoulder from the base. This transformation will simply add h to the z coordinate.

Now we will use the waist joint to rotate the line of the shoulder. This is a rotation about the z axis through an angle θ_1, and we will use our shorthand notation. Frame 2 will now be at shoulder height with the y axis along the line of the shoulder pivot:

$${}^1_2T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, since the upper arm is offset from the shoulder, for frame 3 we should step in the y direction to the line of the upper arm, distance a:

$${}^2_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Now the shoulder axis rotates the upper arm θ_2 about the y axis of frame 3, so we align frame 4 with that limb. But should we align it with x or z? It seems logical to measure the arm's angles up and down from "straight out," so we choose x.

$${}^3_4T = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now we must "move down the upper arm" to the elbow, by a distance l, say. This is in the x direction of frame 4, so

$${}^4_5T = \begin{bmatrix} 1 & 0 & 0 & l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In the Unimation Puma, the forearm is offset slightly from the upper arm, but to avoid adding an extra frame, we can take account of this in the value of a, above.

So now let us bend the elbow through θ_3 and line up frame 6 with the forearm. Once again, the pivot is the y axis and zero deflection is taken as "elbow straight":

$$ {}^5_6T = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Frame 7 is lined up with the forearm, but has moved down to the wrist, distance m:

$$ {}^6_7T = \begin{bmatrix} 1 & 0 & 0 & m \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} $$

Frame 8 follows the first wrist rotation θ_4 about the local x axis.

Frame 9 "waves farewell" θ_5 about the local y axis.

Frame 10 "twists the screwdriver" θ_6 about the local x axis.

Finally, frame 11 "reaches" the tip of the "screwdriver."

As an exercise, write down the corresponding transformations.

So, just what do we do with all these matrices? Each matrix transforms the coordinates to the next-lower frame of reference; the final transformation ${}^0_1T$ brings us to the reference frame 0. But remember that the matrices are stacked up right to left, with the first to be applied closest to the vector that it multiplies, which in this case is the coordinate of a point with respect to the gripper axes. So the product ends up as

$$ {}^{0}_{11}T = {}^0_1T \quad {}^1_2T \quad {}^2_3T \quad {}^3_4T \quad {}^4_5T \quad {}^5_6T \quad {}^6_7T \quad {}^7_8T \quad {}^8_9T \quad {}^9_{10}T \quad {}^{10}_{11}T $$

We have more matrices to multiply than there are axes, but they are all elementary rotations about an axis or translation along an axis. A prismatic joint appears no different from translation along a limb. The only difference is that the distance parameter will be a variable.

Although the final matrix will be unique, there can be many ways to get there. Rotations about the y axis can be changed so that the "travel" along a limb is in the z direction, rather than x. But when the matrices are all multiplied together, they must give the same result.

There is another methodology that involves just one matrix for each actuated axis. The matrices are not primitives, as above, but are generally the product of three elementary moves.

9.3.2 D–H Parameters

The mechanism consists of a chain of links between one axis and the next. The Denavit–Hartenberg convention is based on making all rotations and prismatic actuations take place about the z axis of a frame:

- We have a set of axes at each joint. The z axes z_{n-1} and z_n at each end of link n are aligned with the axis of rotation or translation there.
- The x axis x_n at the "outer end" is chosen so that it is normal to both of these z axes.
- Now that we know x_n and z_n, we can define y_n to be perpendicular to these to make up a righthanded set of axes.
- If the z axes are not parallel, the transformation for that link must include a "twist" α about the x axis.
- The translation will consist not only of a displacement l in the x direction, but can also have a z component d to account for an offset between the points where the "previous" and the "next" normals intersect the z axis.

For a rotation θ about the first of these z axes, this results in a transformation matrix between these frames:

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & l\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & l\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The link transformation can thus be defined by a set of D–H parameters: the actuation angle θ, the link length l, the link offset d, and the twist α.

But with the slightest change in the convention, the "formula" for the transformation will be changed. It is my opinion that the approach of chaining a set of elementary transformations is safer and better.

9.3.3 Inverse Kinematics

Of course, calculating the kinematics of the robot is only half the story. We can now express the location and orientation of the gripper in terms of the axis movements, but what we really want is to find the axis values needed to put the gripper in some desired position. This calculation is referred to as *inverse kinematics*.

To find the required joint angles, we can calculate the transformation representing the desired position and then compare coefficients with the general transformation that is full of sines and cosines of those joint angles. That leaves us with some unpleasant simultaneous equations to solve. In fact, the result of aligning the three rotations of the wrist joint of the Unimation Puma through the same point reduces the algebra and trigonometry significantly. Nevertheless, the solutions are not unique.

For any given gripper position and attitude, there is an "elbow up" solution as well as an "elbow down" one. These are doubled again with "lefty" and "righty." By "turning its back" on the work, the robot can turn its single "right arm" into a left one.

Then, of course, not all positions have a solution. The desired position might be just out of reach of the outstretched arm.

Another problem is *singularity*. The robot normally has 6 degrees of freedom. But when two joints are in line, such as the wrist and "screwdriver twist," the degrees of freedom drop to 5. In the neighborhood of a singularity, one of the axes will have to move rapidly for the slightest change of the target position.

Think of the problem of trying to watch aircraft as they fly past straight overhead.

Of course, the robot might not have six axes, and we might not wish to move in all 6 degrees of freedom. For example, a "pick and place" robot might be concerned with placing components on a circuit board. The components are presented "flat," so we have no need to tilt them. We might, however, need to rotate them about a vertical axis to align them with the board, in which case we would need to move them to an accurate x–y position. We need a fourth axis to lift them above the board before we place them, but this might just travel between two stops.

Clearly, for a solution to make sense, there must be the same number of control axes as we wish to obtain degrees of freedom. But what if our robot has seven axes?

For various reasons, extra axes might be added, perhaps to allow the robot to "reach around corners." In this case a unique solution is impossible, not even a choice of one in four. To extract a solution, an extra condition has to be imposed, such as that one axis is held at an extreme or at zero deflection.

9.4 ROBOT DYNAMICS

From the kinematics, we have a chain of matrices that can be multiplied together to obtain the transformation matrix describing the motion of a robot. The right hand column defines the location of the origin of the gripper, while a 3×3 submatrix tells us the gripper's orientation. From this submatrix we can unravel the parameters in terms of pitch, roll, and yaw to obtain a vector with six components:

$$(x, y, z, \theta, \phi, \psi)'$$

Each of these coefficients will be a function of all six joint axes

$$x(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$
$$y(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$$

and so on.

Although solving to find functions for the axis values might not be easy, we can find the effect of a "twitch" in one of the axes by partial differentiation.

If we change just θ_1 by $\delta\theta_1$, the change in x will be

$$\delta x = \frac{\partial x}{\partial \theta_1}\delta\theta_1$$

If we change more of the joints, we will have

$$\delta x = \frac{\partial x}{\partial \theta_1}\delta\theta_1 + \frac{\partial x}{\partial \theta_2}\delta\theta_2 + \frac{\partial x}{\partial \theta_3}\delta\theta_3 + \frac{\partial x}{\partial \theta_4}\delta\theta_4 + \frac{\partial x}{\partial \theta_5}\delta\theta_5 + \frac{\partial x}{\partial \theta_6}\delta\theta_6$$

In fact, we can calculate all the partial derivatives to find the Jacobian, a matrix that has these partial derivatives as its coefficients.

Now, at any given position, these coefficients will just be numbers that we can calculate, so that we can find the effect of "nudging" the joints from

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\theta \\ \delta\phi \\ \delta\psi \end{bmatrix} = J \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \\ \delta\theta_3 \\ \delta\theta_4 \\ \delta\theta_5 \\ \delta\theta_6 \end{bmatrix}$$

If we are off target, we know the values we need to approach it, at least to a first approximation. We should therefore be able to calculate a set of axis corrections to bring us closer, simply by inverting the Jacobian and multiplying by the error vector.

Often this will work! But it is possible that the Jacobian is singular and has no finite inverse. That is what happens at a singularity.

All is not lost. A method of successive approximations can bring us closer to the target, or to the point in the "reachable" space that is closest to it. For each axis in turn, inspect the corresponding column of the Jacobian and decide whether a positive or a negative nudge will bring us closer to the target, or whether that axis should remain the same. Apply the nudges and measure the new error. When there is no sign of improvement, halve the nudge size.

The Jacobian also relates the gripper velocity to the velocities of the axes. If the objective is to move it along a path at maximum speed, one or more of the axes will be required to reach maximum velocity. As the gripper moves along the path, the identity of the limiting axis will probably change. Once again, the Jacobian and its inverse will be valuable tools in calculating the axis drive values.

9.5 SIMULATING A ROBOT

Many years ago my son, Richard, helped me develop a package to simulate and articulate robot mechanisms. A version has been converted into Visual Basic and is available on the Web at http://www.essmech.com/9/5.htm.

The initial task is to design robot "parts," sets of points in three dimensions joined by a selection of lines. The data format is a set of coordinate triples defining the points and a set of integer pairs defining the pairs of points to be joined by lines.

These parts are then "assembled" to construct the robot. The robot can take the form of a simple chain, such as a manipulator, or alternatively a robot with multiple attachments such as articulated legs. The restriction is that there are no closed chains.

So, how are the parts "attached"? Two points on a component of the assembly are defined as "primary" and "secondary." They will act as the hinge about which the new part will rotate. Two points on the new part are also defined as primary and secondary where the hinge will be attached. To align the new part, the two primary points are moved together, by a simple displacement, and the new part is rotated to bring the two vectors between primary and secondary points into line. The hinge is now complete.

Each part or "limb" of the assembly now has a set of properties. First is the identifier of the shape that it takes—several parts can use the same shape. Second is the identity of the "parent" limb, the part to which it is attached, with a pair of integers to define the primary and secondary points of the parent that form the hinge. Another pair of integers will define the primary and secondary points of the part itself. Two variables describe the hinge angle and its datum value. Finally, a transformation matrix describes the absolute position and orientation of the part.

Each hinge is manipulated in turn. To change the hinge angle, a transformation is calculated that represents a rotation about the line of the hinge. This is applied to the part that "owns" the hinge and also to any other parts that are attached to that part.

The line of the hinge, defined by the primary and secondary points $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ of the parent, is found by multiplying the parent's shape coordinates by its transformation.

So, how can we find the transformation that represents rotation about this line? Let us first consider its 3×3 rotation matrix. Suppose that the direction of the hinge axis is given by the unit vector $\boldsymbol{c}$. What is the effect of rotation about this axis through the origin on a general vector $\boldsymbol{x}$?

We can break $\boldsymbol{x}$ down into a component in the direction of $\boldsymbol{c}$ and another orthogonal to it:

$$(\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c}$$

$$\boldsymbol{x} - (\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c}$$

When we rotate $\boldsymbol{x}$ about $\boldsymbol{c}$ through an angle θ, this perpendicular component will become

$$(\boldsymbol{x} - (\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c})\cos\theta$$

and there will be a second component the same size as

$$(\boldsymbol{x} - (\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c})\sin\theta$$

but in a direction perpendicular to both $\mathbf{c}$ and the orthogonal component of $\boldsymbol{x}$. But we can "turn" the orthogonal component to line up with this direction, simply by taking its cross-product with the unit vector $\boldsymbol{c}$ to get

$$\boldsymbol{c} \times (\boldsymbol{x} - (\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c})\sin\theta$$

But $\boldsymbol{c} \times \boldsymbol{c} = 0$, so this reduces to

$$\boldsymbol{c} \times \boldsymbol{x}\sin\theta$$

The three components can be combined to give the resulting vector

$$(\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c} + \{\boldsymbol{x} - (\boldsymbol{c} \cdot \boldsymbol{x})\boldsymbol{c}\}\cos\theta + \{\boldsymbol{c} \times \boldsymbol{x}\}\sin\theta$$

or

$$(\boldsymbol{c}\cdot\boldsymbol{x})\boldsymbol{c}\,(1-\cos\theta)+\boldsymbol{x}\cos\theta+\{\boldsymbol{c}\times\boldsymbol{x}\}\sin\theta$$

This is fine as a mathematical expression, but to be useful, we have to express it in matrix terms for computing. We can rearrange the first term as a matrix multiplied by $\boldsymbol{x}$. We can also express the cross-product $\boldsymbol{c}\times\boldsymbol{x}$ as the product of $\boldsymbol{x}$ with a matrix.

So, in matrix terms we have the result

$$\left(\begin{bmatrix} c_1c_1 & c_1c_2 & c_1c_3 \\ c_2c_1 & c_2c_2 & c_2c_3 \\ c_3c_1 & c_3c_2 & c_3c_3 \end{bmatrix}(1-\cos\theta)+\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\cos\theta+\begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix}\sin\theta\right)\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The braces give a "recipe" for a matrix that describes the rotary part of the transformation, R. But we need to take into account that the line $\boldsymbol{p}_0$–$\boldsymbol{p}_1$ probably does not pass through the origin, so the matrix becomes 4 × 4 with a translation component in the fourth column.

For the translation part, we first subtract the coordinates of the primary point $\boldsymbol{p}_0$ from $\boldsymbol{x}$, then multiply by R and add $\boldsymbol{p}_0$ again. So the fourth column of the transformation is given by

$$\boldsymbol{p}_0 - R\boldsymbol{p}_0$$

There is no need to record the bottom row of the transformation matrix, since this is always (0 0 0 1). Although needed for the perfection of a mathematician's algebra, the computer is perfectly capable of performing 3 × 4 matrix operations without it.

The same sort of transformation is needed to align the hinge when attaching a new part. In this case, the axis of rotation is the cross-product of the two vectors that join primary and secondary points of the component and of its parent. To calculate the angle of the rotation needed, we note that the scalar product of the two vectors divided by the product of their moduli gives us the cosine of the angle between them. The magnitude of the cross-product divided by the product of their moduli gives us the sine.

Robot joints are not always revolute. Some are prismatic, where one part slides linearly against another. This transformation is much simpler to calculate than the rotary one. It simply involves adding a proportion of vector $\boldsymbol{c}$ to every point:

$$\begin{bmatrix} 1 & 0 & 0 & kc_1 \\ 0 & 1 & 0 & kc_2 \\ 0 & 0 & 1 & kc_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now each part can be multiplied by its transformation to give its absolute position.

One way to project the coordinates for plotting on the screen is to ignore the y coordinate and simply plot the (x,z) coordinates. If desired, however, a perspective projection is simple. Plot $z/(y + r)$ against $x/(y + r)$, where r is the distance from which the robot is viewed.

Of course, the code on the Website is only the beginning. Once you have designed and tested your robot, you need to rewrite a large part of the code so that you can coordinate the simultaneous movement of the axes.