Chapter 4

Kinematic Systems

Kinematics is a subfield of dynamics in which the geometrical aspects of motion are considered—apart from issues of mass and acceleration. The kinematic influence of the skeleton on the movement of an animal is fundamental and inescapable. Far from being a curse, however, exploiting the kinematic aptitudes of a biomechanical system leads to higher performance: it can produce an economy and grace of movement; enhances sensory and motor acuity; and guides the search for skillfull motor strategies. The same kinematic principles can be used to guide robot programming and to exploit the aptitudes of robot devices.

This chapter introduces tools for analyzing movement in kinematic systems. All machines introduce "sweet spots" where attractive kinematic properties can be exploited. We begin with a discussion of mechanisms [233] after which the kinematic performance of articulated machines is examined in some detail, including forward and inverse mappings between body configurations and Cartesian task descriptions. The governing kinematic equations for hand-eye coordination are presented and the chapter concludes by reviewing measures of *kinematic conditioning*—descriptions of the ability of mechanisms to project velocity and force efficiently into Cartesian space.

4.1 Terminology

Engineers have been studying principles of machine design for a very long time. At the most basic level, animals and robots are mechanisms whose behavior is governed by the same principles. A *mechanism* is a collection of bearings, gears, cams, links, and belts that convert motions from one form to another and transmit power. It is an assemblage of individual rigid bodies known as **links** that are connected in pairs through kinematic constraints called **joints**. A joint permits a specific relative motion between adjacent links—**prismatic** joints permit translations and **revolute** joints support a relative rotation.

Any parameter (length or angle) of a mechanism that specifies the position of an independent joint is called a **configuration variable**. A **kinematic chain** is an assemblage of interconnected links whose kinematic properties depend

on how links and joints are combined. The posture of devices with multiple configuration variables is a coordinate in the **configuration space**. The minimum number of such variables necessary to fully define the configuration of a mechanism is called the **degree of freedom (DOF)** of the system.

A mechanism is formed when one of the links is held fixed and the others move relative to the fixed link. The fixed link is referred to as the **ground link** or **ground frame**. Most mechanisms considered in the machine design literature consist of **closed chains**, that is, kinematic chains with every link connected through joints to two adjacent links. However, many robotic devices are **open chains** wherein one or more so-called **unitary links** are connected to only one joint.

4.2 Spatial Tasks

Tasks specify (directly or indirectly) the spatial configuration of limbs for locomotion and/or manipulation, the positioning of sensory devices, and the relative geometry of external objects in the environment. To meet task specifications, control inputs are used to modify the spatial configuration of the mechanisms comprising the robot.

In general, several frames of reference may be necessary to fully specify a robot control task. Kinematic relations such

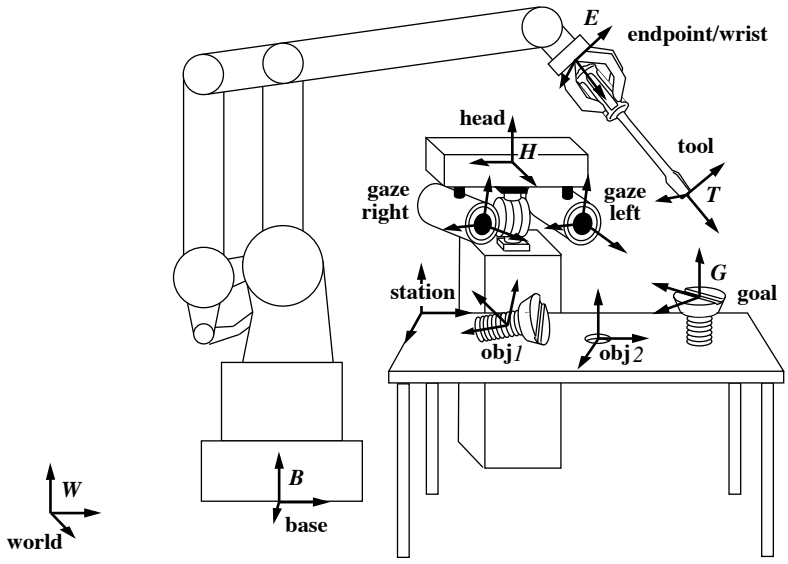


Figure 4.1 *Coordinate frames commonly used to specify robot tasks.*

as those illustrated in Figure 4.1 are naturally arranged in tree structures rooted in a fixed **world** frame. The distribution of objects are often described in terms of their position and orientation in the **world** frame or task frames fixed in the world to avoid accumulating uncertainty. For example, in Figure 4.1, the **station** coordinate system is such a frame. It describes aspects of the task related to the **world** through fixed spatial offsets. From the **station** frame, several other

features of the task (**obj1**, **obj2**, **goal**) may also be specified as fixed offsets or determined using sensor systems.

The sensory and motor resources of the robot can also be located in the **world** frame using spatial transformations. For example, the transform from **world** to **base** locates a coordinate frame attached to the robot. It could be a fixed position relative to the **world** frame as in Figure 4.1, or it could be time-varying and controllable as in the case of a mobile manipulator. In the latter case, it is established by integrating displacements[1] in the mobile base in a process called *odometry* or by relation to other visual or tactile features in the room whose positions are known.

From the **base** frame, two important frames establish the spatial distribution of sensory and motor resources that can be applied to the task. The first locates the **wrist** frame and the second identifies the pose of the sensor **head** relative to the **base**. These frames reflect both fixed offsets in the body and the posture of controllable degrees of freedom measured using sensors in every joint. Together with knowledge of the hand posture, they constitute multiple *forward kinematic* relations in the robot that depend of configuration variables and, therefore, support reasoning about spatial errors and control. The **tool** frame in Figure 4.1 identifies the working end of the grasped screwdriver. It is estimated by considering visual and tactile feedback in the context, perhaps, of prior information about the geometry of screwdrivers.

Given a complete specification of the kinematic tree described above, control tasks can be expressed in terms of the coordinate frames in Figure 4.1. In principle, then, movements in the actuated degrees of freedom can be computed in order to bring the **tool** and **goal** frames together and turn the screw. However, spatial estimates are not all equally precise. The geometry of the arm, fingers and head depends on the dimensions of the mechanisms and measurements of joint angles, which are relatively precise. In contrast, fingertip contact locations, visual feedback, and estimates of the grasped pose of the screwdriver can be significantly less precise.

EXAMPLE: Kinematic Description of "Roger-the-Crab"

"Roger-the-Crab" was developed by Paul Churchland in 1988 to demonstrate how the Cartesian space around a simple organism can be encoded in neural projections between sensory and motor maps [56]. The original Roger was a single planar, two degree of freedom arm and a stereo vision system. Figure 4.2 defines the kinematic parameters and intermediate coordinate frames of a more elaborate mobile, bi-manual version of Roger. The robot lives in the x-y plane and has a nine dimensional configuration space consisting of two "pan" degrees of freedom in the eyes (θ_1, θ_2)*, two degrees of freedom for the left arm* (θ_3, θ_4) *and the right arm* (θ_5, θ_6)*, and three degrees of freedom in the mobile base* $(x, y, \theta)_0$*. Each revolute joint (except for the wheels) rotates about the world frame* $\hat{\mathbf{z}}$ *axis. Roger's wheel and arm joints rotate continuously and his eyes have a range of motion limited to* $-\pi/2 \leq \theta_1, \theta_2 \leq \pi/2$.

Roger consists of several kinematic chains that establish spatial quantities marked by intermediate coordinate frames. Four important coordinate frames are rooted at fixed locations relative to the **base** *frame. Frames* $\boldsymbol{V1}$ *and* $\boldsymbol{V2}$ *establish the positions of Roger's eyes relative to the* **base***, and frames* $\boldsymbol{A1}$ *and* $\boldsymbol{A2}$ *establish the positions of the shoulders of Roger's arms relative to the* **base***. In Roger, these frames are rigid translations from frame* $\boldsymbol{B}$ *in the* $\hat{\mathbf{y}}_B$ *direction and are used to describe and control the geometry of sensors and effectors deployed by the robot.*

[1] Displacements are commonly determined by measuring wheel rotations, counting strides, or by using inertial measurement units (IMUs).

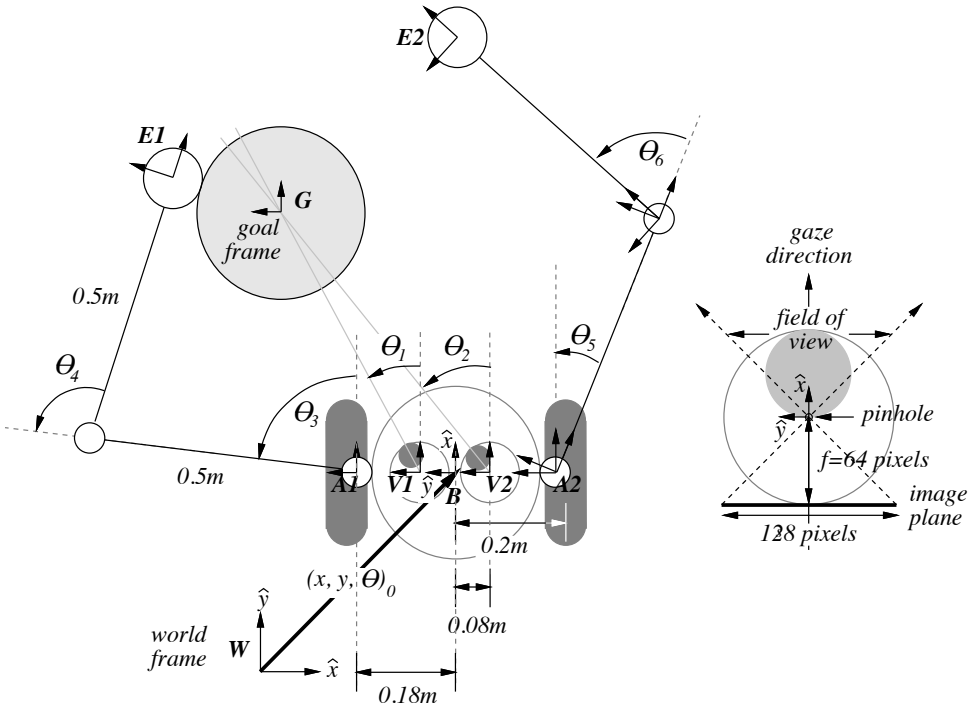


Figure 4.2 *The kinematic definitions and intermediate coordinate frames used to define Roger.*

Information about the **goal** *frame is derived from visual and tactile sensors. Cameras at frames* $\boldsymbol{V1}$ *and* $\boldsymbol{V2}$ *can pan independently using actuated joints* θ_1 *and* θ_2*. The transform from frames* $\boldsymbol{V1}$ *and* $\boldsymbol{V2}$ *to frames aligned with the gaze (right inset) is a pure rotation parameterized by the configuration variables. The eyes are pinhole RGB cameras (Section 4.5.1) with a focal length of* 64 *pixels*[2] *that produce a one dimensional image* 128 *pixels wide. Each eye, therefore, has a field of view of* ± 45 *degrees directed into the world by motor actions that pan the eyes and/or the body. Signals from the camera pair can be used to estimate the goal location by triangulation.*

Tactile sensors are located at **endpoint** *frames* $\boldsymbol{E1}$ *and* $\boldsymbol{E2}$*. The transform from* **arm** *frames to the* **endpoint** *frames depends on the geometry of links and actuated joints (*θ_3*,* θ_4*,* θ_5*,* θ_6*) in the arms. Tactile observations take the form of force vectors in the x-y plane written in the end effector coordinate frame. Like vision, signals from the tactile sensor together with knowledge of the configuration of the robot can be used to locate the Cartesian location of the* **goal** *frame* $\boldsymbol{G}$ *via forward kinematic relations (Section 4.4.1).*

The task of positioning the endpoint frames and directing the gaze of the eyes is completely controllable using commands to motors that actuate each of the degrees of freedom in these kinematic chains. Such completely controllable kinematic systems are **holonomic**.[3] *The pose of the mobile base is established by defining the location and orientation of the* **base** *coordinate frame using odometry to estimate* $(x, y, \theta)_0$ *in* **world** *coordinates. Navigation tasks for the wheeled base are expressed in this three dimensional pose space as well. However, unlike the other kinematic systems comprising Roger,*

[2] A pixel (a picture element) is a single receptive field on the image plane. Pixels tile the image plane and, therefore, define a unit of length.
[3] Holo- (in Greek ὅλος) means "whole" and -nomic (from νόμος) means "law."

the mobile base is **nonholonomic***—the wheels do not permit translational velocities along* $\hat{\boldsymbol{y}}_B$*. In* **world** *coordinates, from any given pose* $\boldsymbol{q} = [x\ y\ \theta]^T$*, the wheeled system can generate rotational velocities* $\dot{\hat{\boldsymbol{q}}}_\theta = [0\ 0\ 1]^T$ *and translational velocities along the current heading* $\dot{\hat{\boldsymbol{q}}}_{xy} = [cos(\theta)\ sin(\theta)\ 0]^T$*. Therefore, the controllable velocity from any given pose is a two dimensional subset of the pose velocity space and is expressed as a linear combination of a rotational velocity and a pose-dependent translational velocity*

$$\dot{\boldsymbol{q}} = v_\theta \dot{\hat{\boldsymbol{q}}}_\theta + v_{xy} \dot{\hat{\boldsymbol{q}}}_{xy},$$

where v_θ *and* v_{xy} *are scalar magnitudes for rotational and translational velocities, respectively. The difference between the number of degrees of freedom in the task and those in the control is stated succinctly in the form of a single nonholonomic kinematic constraint that restricts translational velocities*

$$f(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{l}^T \dot{\boldsymbol{q}} = [sin(\theta)\ \text{-}cos(\theta)\ 0] \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0,$$

where, $\boldsymbol{l}$ *is the vector in the x-y plane that is orthogonal to the current vehicle heading (the lateral direction). Now, the navigation task becomes more challenging—the absence of a "whole kinematic law" introduces interesting control and path planning issues. Other examples exist of important nonholonomic systems as, for instance, when fingertips roll over the surface of a grasped object during manipulation.*

□

To construct control tasks, we require a means of transforming (sets of) Cartesian quantities expressed in one coordinate frame into another frame appropriate for control. In the next section, we introduce one such representation called the homogeneous transform.

4.3 Homogeneous Transforms

A *group* is a set of elements that is closed under a binary operation—that is, applying the operator to pairs of elements in the set produces a result that is also an element of the set. In addition, to constitute a group the operator must be associative, i.e. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, the group must include an identity element I, such that $a \cdot I = I \cdot a = a$, and every element a in the set must have an inverse a^{-1}, such that $a \cdot a^{-1} = I$, that is also in the set. The classical example of a group is the set of integers and the addition operator.

Rigid body motions describe Euclidean groups. In particular, a three dimensional translation group T consists of elements that are displacements $\boldsymbol{t} \in \mathbb{R}^3$ ($[x\ y\ z]^T$, for example). The translation group is closed under addition. Moreover, the elements of the *Special Orthogonal* group in three dimensions $SO(3)$ designates three orthogonal rotations (roll, pitch, yaw, for example). The rotation group is closed under multiplication and compositions of elements preserve Euclidean distances. Groups T and $SO(3)$ are independent subgroups of the *Special Euclidean* group $SE(3)$, elements of which describe how rigid bodies move in a six dimensional Cartesian space. As a consequence, to completely define the pose of a rigid body, one needs to specify a minimum of six independent variables that specify the translation and rotation of the body.

Translations

Figure 4.3 shows two coordinate frames, $\boldsymbol{A}$ and $\boldsymbol{B}$, that are related through a pure translation. A position vector $\boldsymbol{r}_B$ written in frame $\boldsymbol{B}$ is expressed in coordinate frame $\boldsymbol{A}$ using the linear transformation

$$\boldsymbol{r}_A = \boldsymbol{r}_B + {}_A\boldsymbol{t}_B,$$

where ${}_A\boldsymbol{t}_B$ represents the translation from frame $\boldsymbol{A}$ to frame $\boldsymbol{B}$ written in frame $\boldsymbol{A}$ coordinates. During a pure translation, the relative orientation of the two frames remains fixed and $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ coordinate axes remain parallel.

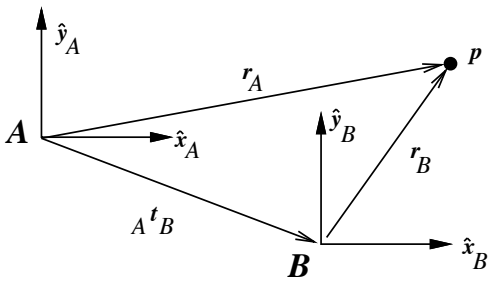


Figure 4.3 *Two coordinate frames related through a pure translation.*

Rotations

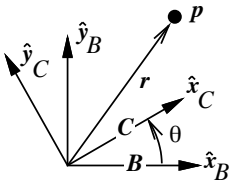


Figure 4.4 *Two coordinate frames related through a pure rotation.*

There are several alternatives for representing rotations [301], including exponential coordinates [214], Euler angles, roll-pitch-yaw notations, quaternions, and direction cosines. Of these choices, the direction cosines are the least compact—requiring nine numbers to specify three rotation variables—but, as we will see, direction cosines contribute to a simple, invertible, and homogeneous representation for $SE(3)$.

To derive the direction cosine matrix, consider frames $\boldsymbol{B}$ and $\boldsymbol{C}$ in Figure 4.4 that are related through a pure rotation. Position vector $\boldsymbol{r}$ written in frame $\boldsymbol{C}$ coordinates can be expressed in frame $\boldsymbol{B}$ coordinates by employing the 3×3 direction cosine matrix ${}_B\mathbf{R}_C$, such that

$$\boldsymbol{r}_B = {}_B\mathbf{R}_C\, \boldsymbol{r}_C, \text{ and}$$

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}_B = \begin{bmatrix} \hat{\boldsymbol{x}}_B \cdot \hat{\boldsymbol{x}}_C & \hat{\boldsymbol{x}}_B \cdot \hat{\boldsymbol{y}}_C & \hat{\boldsymbol{x}}_B \cdot \hat{\boldsymbol{z}}_C \\ \hat{\boldsymbol{y}}_B \cdot \hat{\boldsymbol{x}}_C & \hat{\boldsymbol{y}}_B \cdot \hat{\boldsymbol{y}}_C & \hat{\boldsymbol{y}}_B \cdot \hat{\boldsymbol{z}}_C \\ \hat{\boldsymbol{z}}_B \cdot \hat{\boldsymbol{x}}_C & \hat{\boldsymbol{z}}_B \cdot \hat{\boldsymbol{y}}_C & \hat{\boldsymbol{z}}_B \cdot \hat{\boldsymbol{z}}_C \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}_C, \tag{4.1}$$

where $\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}}$ represent the orthonormal basis for a coordinate frame.

The "dot" operator in Equation 4.1 is the dot (or scalar) product (Appendix A.1). Columnwise, the direction cosine matrix maps the orthonormal (orthogonal and unit length) basis vectors in frame $\boldsymbol{C}$ into a new orthonormal basis written in frame $\boldsymbol{B}$. The first column, for example, projects basis vector $\hat{\boldsymbol{x}}_C$ onto the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ axes of frame $\boldsymbol{B}$. The matrix multiply in Equation 4.1 sums the independent projections of the x-, y-, and z-components of $\boldsymbol{r}_C$ onto the basis vectors for frame $\boldsymbol{B}$.

The rotation matrix ${}_B\mathbf{R}_C$ can be derived geometrically by writing basis vectors $(\hat{\boldsymbol{x}},\ \hat{\boldsymbol{y}},\ \hat{\boldsymbol{z}})_C$ in frame $\boldsymbol{B}$ coordinates and then entering the result into the appropriate columns of ${}_B\mathbf{R}_C$. Consider the rotation illustrated in Figure 4.5 and let $\hat{\boldsymbol{x}}_C^B$

be the projection of $\hat{\boldsymbol{x}}_C$ onto coordinate frame $\boldsymbol{B}$.

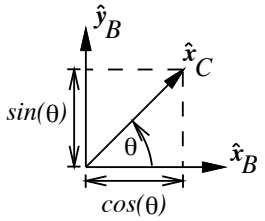


Figure 4.5 *Projecting $\hat{\boldsymbol{x}}_C$ onto the x-y plane of frame $\boldsymbol{B}$.*

$$\hat{\boldsymbol{x}}_C^B = \begin{bmatrix} cos(\theta) \\ sin(\theta) \\ 0 \end{bmatrix}$$

Repeating this geometric construction for $\hat{\boldsymbol{y}}_C^B$ and $\hat{\boldsymbol{z}}_C^B$ yields

$${}_B\mathbf{R}_C = \begin{bmatrix} \hat{\boldsymbol{x}}_C^B & \hat{\boldsymbol{y}}_C^B & \hat{\boldsymbol{z}}_C^B \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.2}$$

The columns of ${}_B\mathbf{R}_C$ represent the basis vectors of frame $\boldsymbol{C}$ projected onto the basis vectors of frame $\boldsymbol{B}$. Conversely, the rows of ${}_B\mathbf{R}_C$ represent the projection of the basis vectors for frame $\boldsymbol{B}$ onto the basis vectors of frame $\boldsymbol{C}$. This suggests, and it is clear by inspection of Equation 4.1, that the inverse of the direction cosine matrix is its transpose, ${}_B\mathbf{R}_C^{-1} = {}_B\mathbf{R}_C^T = {}_C\mathbf{R}_B$.

Finally, note that the rotation illustrated in Figure 4.4 is about the $\hat{\boldsymbol{z}}$ axis and, therefore, preserves the direction of the $\hat{\boldsymbol{z}}$ axis. The positive sense of the rotation indicates the rotation from frame $\boldsymbol{B}$ to frame $\boldsymbol{C}$.

$${}_B\mathbf{R}_C = rot(\hat{\boldsymbol{z}}, \theta)$$

Rotations about $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are constructed in the same way. For the sake of completeness, the direction cosine matrix for pure rotations about the $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{y}}$, and $\hat{\boldsymbol{z}}$ axes are summarized here,

$$rot(\hat{\boldsymbol{x}}, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix} \quad rot(\hat{\boldsymbol{y}}, \theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \quad rot(\hat{\boldsymbol{z}}, \theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

where $c\theta$ and $s\theta$ are shorthand notation for $cos(\theta)$ and $sin(\theta)$, respectively.

The **homogeneous transform** incorporates translation and rotation in a single linear transformation. It applies a translation from frame $\boldsymbol{A}$ to frame $\boldsymbol{B}$ followed by a rotation from frame $\boldsymbol{B}$ to frame $\boldsymbol{C}$ as illustrated in Figure 4.6. The order of application of the transforms is important—spatial transforms such as these are not commutative.

The homogeneous transform ${}_A\mathbf{T}_C$ and the homogeneous position vector are defined:

$${}_A\mathbf{T}_C = \left[\begin{array}{c|c} {}_B\mathbf{R}_C & {}_A\boldsymbol{t}_B \\ \hline 0\ 0\ 0 & 1 \end{array}\right] \qquad \boldsymbol{r}_C = \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix}_C.$$

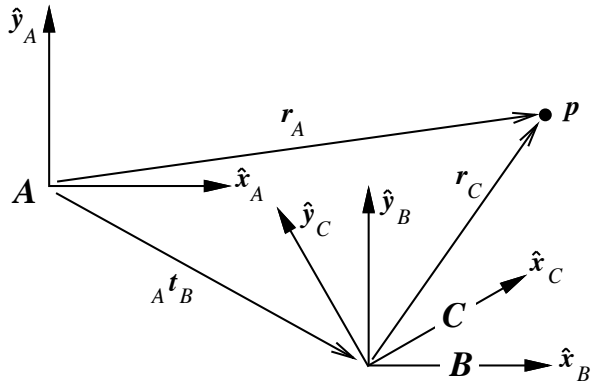


Figure 4.6 *Two coordinate frames related through a rotation and a translation.*

The fourth row in each of these definitions is there to provide the bookkeeping required to sum the affects of the translation and rotation correctly. It is left to the reader to verify that the product $\boldsymbol{r}_A = {}_A\mathbf{T}_C\, \boldsymbol{r}_C$ yields the intended homogeneous position vector for the construction illustrated in Figure 4.6.

Inverting the Homogeneous Transform

The inverse of the homogeneous transform is simple to derive. Here, we simply present the solution and leave it to the reader to verify.

$$ {}_A\mathbf{T}_B = \left[\begin{array}{ccc|c} \hat{\boldsymbol{x}}_B^A & \hat{\boldsymbol{y}}_B^A & \hat{\boldsymbol{z}}_B^A & \boldsymbol{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad {}_B\mathbf{T}_A = [{}_A\mathbf{T}_B]^{-1} = \left[\begin{array}{c|c} (\hat{\boldsymbol{x}}_B^A)^T & -(\hat{\boldsymbol{x}}_B^A)^T\boldsymbol{t} \\ (\hat{\boldsymbol{y}}_B^A)^T & -(\hat{\boldsymbol{y}}_B^A)^T\boldsymbol{t} \\ (\hat{\boldsymbol{z}}_B^A)^T & -(\hat{\boldsymbol{z}}_B^A)^T\boldsymbol{t} \\ \hline 0\ 0\ 0 & 1 \end{array}\right] $$

The homogeneous transform is commonly used in robotics and computer graphics and is one way of representing the kinematic transformations between the coordinate frames indicated in task descriptions like Figures 4.1 and 4.2. It provides a closed-form method for describing spatial transformations associated with articulated mechanisms and is easily composed and inverted. Moreover, the homogeneous transform is the basis for the standardized *Denavit-Hartenberg* notation that makes it possible to represent a complete kinematic specification for a robot using tables of standardized D-H parameters [234].

4.4 Manipulator Kinematics

Forward and inverse kinematic relations describe reciprocal mappings between an n-dimensional configuration space ($\boldsymbol{q} \in C^n$) and Cartesian space ($\boldsymbol{r} \in SE(3)$). Forward kinematic transformations map configuration space to Cartesian space, $\boldsymbol{q} \mapsto \boldsymbol{r}$, and inverse kinematic transformations map them back again, $\boldsymbol{r} \mapsto \boldsymbol{q}$. The character of these mapping functions reveals a great deal about the aptitudes and limitations of robotic mechanisms.

4.4.1 Forward Kinematics

Closure and associativity properties of $SE(3)$ imply that homogeneous transforms can be used to compose other homogeneous transforms. This is of great consequence when describing tasks, especially when this description incorporates the configuration variables of a robot.

Consider the forward kinematics of the two degree of freedom robot arm used as the manipulator in the Roger-the-Crab simulator (Figure 4.2). This manipulator geometry consists of a series of two revolute joints with parallel axes of rotation so that the manipulator remains in the plane normal to this axis of rotation. For this reason, it is often called the planar 2R manipulator.

EXAMPLE: Forward Kinematics of the Planar 2R Manipulator

Figure 4.7 illustrates Roger's arm. It includes several intermediate coordinate frames distributed along the length of the arm. Frame 0 *is the* **arm** *frame that locates the shoulder joint with respect to the* **body** *frame in Figure 4.2. Frames* 1, 2, *and* 3 *represent cumulative displacements from frame* 0. *Frame* 1 *is rotated by* θ_1 *so that* $\hat{\mathbf{x}}_1$ *remains directed along link* 1 *as the shoulder joint moves.*

$$ {}_0\mathbf{T}_1 = rot(\hat{\mathbf{z}}_0, \theta_1) = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Relative to frame 1, *a translation along* $\hat{\mathbf{x}}_1$ *for a distance* l_1 *followed by a rotation of* θ_2 *about* $\hat{\mathbf{z}}_1$ *results in frame* 2, *with* $\hat{\mathbf{x}}_2$ *directed along link* 2. *Finally, relative to frame* 2, *a translation along* $\hat{\mathbf{x}}_2$ *of distance* l_2 *locates the endpoint frame* 3.

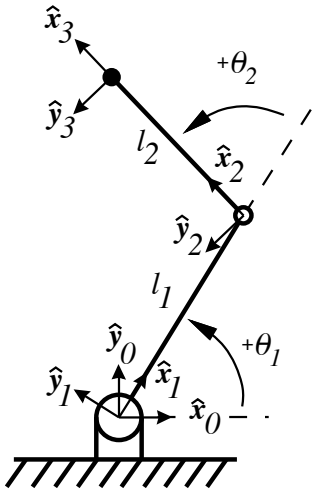


Figure 4.7 *The planar 2R manipulator.*

$$ {}_1\mathbf{T}_2 = trans(\hat{\mathbf{x}}_1, l_1)\, rot(\hat{\mathbf{z}}_1, \theta_2) = \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}_2\mathbf{T}_3 = trans(\hat{\mathbf{x}}_2, l_2) \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The net transformation from frame 0 *to frame* 3 *is the product of this sequence of homogeneous transformations.*

$$ \begin{aligned} {}_0\mathbf{T}_3 = {}_0\mathbf{T}_1\, {}_1\mathbf{T}_2\, {}_2\mathbf{T}_3 &= \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & l_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.4} $$

where: $c_i = cos(\theta_i)$; $s_i = sin(\theta_i)$; $c_{12} = cos(\theta_1 + \theta_2)$ *and identities* $sin(\alpha \pm \beta) = sin\alpha\, cos\beta \pm cos\alpha\, sin\beta$ *and* $cos(\alpha \pm \beta) = cos\alpha\, cos\beta \mp sin\alpha\, sin\beta$ *are used to simplify the result.*

□

The forward kinematic relation in the form of the homogeneous transform ${}_0\mathbf{T}_3$ provides insight into the kinematic character of the planar 2R manipulator. For example, inspecting the rotational part of the transform, we determine that the orientation of the end effector is $(\theta_1 + \theta_2)$. Moreover, the position of the end effector is determined in the fourth

column of the transform,

$$\begin{aligned} x &= l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) \\ y &= l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) \\ z &= 0. \end{aligned} \tag{4.5}$$

These equations can be evaluated over all reachable joint angles to yield the **reachable workspace** for the manipulator. Figure 4.8 illustrates the result for the planar 2R manipulator assuming that each joint is free to rotate continuously (light grey). Also illustrated in darker grey is the reachable workspace for the human shoulder/elbow analog of the 2R manipulator with realistic joint range limits ($-\pi/4 \leq \theta_1 \leq 5\pi/4$, $0 \leq \theta_2 \leq \pi$). Three kinematic variations of Roger's arm are sketched in the figure. The leftmost panel shows Roger's arm where $l_1 = l_2$. Under these conditions, Roger can reach the entire interior (including the perimeter) of the disk with radius $l_1 + l_2$ centered on the shoulder joint. Absent joint range limitations, each position in the interior of the reachable workspace can be achieved using two configurations of the arm—one with the elbow flexed and one with the elbow (hyper)extended. The adult human arm conforms fairly closely to the $l_1 = l_2$ case.

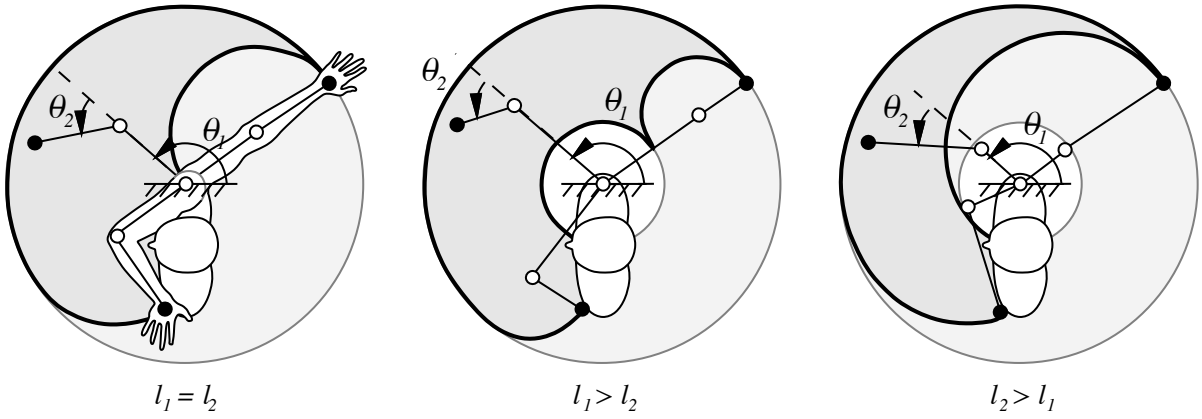


Figure 4.8 *The reachable workspace for the two degree of freedom, planar robot.*

The **dexterous workspace** is the subset of the reachable workspace where all orientations of the end effector can be achieved. This subset is non-empty only when $l_1 = l_2$ (the left panel). It includes only a single Cartesian location (the origin), corresponding to a one dimensional subset of configurations where $\theta_2 = \pm\pi$ and $-\pi \leq \theta_1 \leq \pi$. Given this elbow posture, the endpoint orientation can assume any value in the plane by controlling the position of shoulder in the range $-\pi$ to π.

Realistic joint range limits interact with relative link lengths to significantly influence the reachable workspace and, as the center and right panels of Figure 4.8 illustrate, there are significant advantages to keeping l_1 and l_2 approximately equal in length. While the workspace for the $l_1 > l_2$ and $l_2 > l_1$ cases without joint range limits look similar, with joint limits the case when $l_1 > l_2$ has significant advantages.

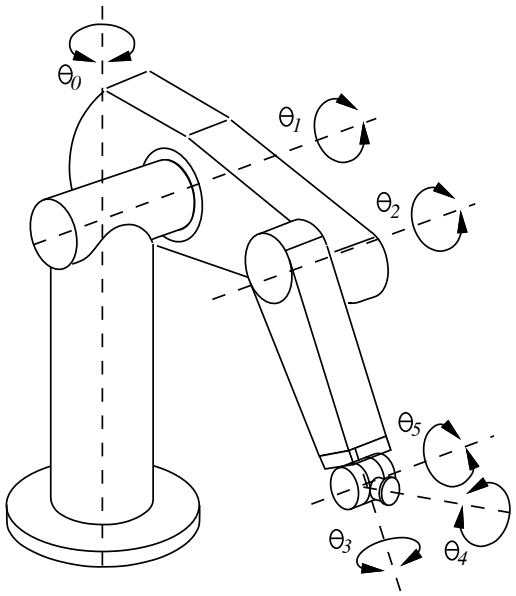


Figure 4.9 *The Unimate PUMA 560.*

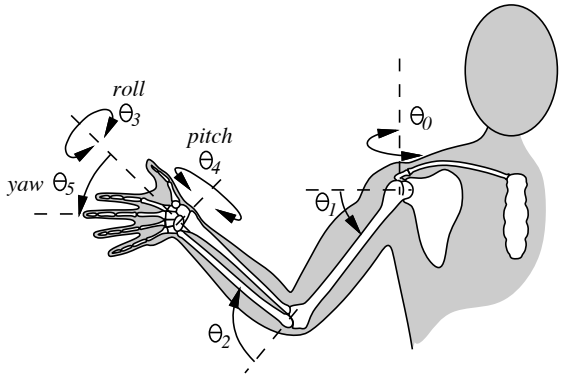


Figure 4.10 *The human shoulder, arm, and wrist. Three intersecting revolute axes in the wrist create an approximately spherical wrist joint important for brachiation in human ancestors.*

4.4.2 Inverse Kinematics

Tasks are efficiently expressed in terms of Cartesian goals $\boldsymbol{r} \in SE(3)$ that are transformed into configuration variables $\boldsymbol{q} \in C^n$ for use in motor controllers. The mapping required is the inverse of the kinematic mapping discussed in Section 4.4.1. However, forward kinematic functions are nonlinear—many Cartesian goals are unreachable, and others can be realized by many possible joint angle configurations. As result, inverting forward kinematic mappings is nontrivial.

Complete inverse kinematic solutions yield all joint angle configurations for any reachable Cartesian goal position. In general, complete inverse kinematic solutions are very difficult (or impossible) to compute in closed-form, however, they do exist for some important special cases. For example, Pieper (ca. 1968) formulated a general inverse kinematic solution for six degree of freedom manipulators consisting of three revolute or prismatic joints followed by three consecutive joints with rotational axes that intersect at a point [243]. The first three degrees of freedom generate positions in $\mathbb{R}^3$ and the spherical wrist generates $SO(3)$. Several robot designs incorporate this kinematic simplification, a notable example of which was the Unimate PUMA 560 (Figure 4.9), the first commercially available arm. The human arm/wrist geometry illustrated in Figure 4.10 embodies these kinematic properties as well. Three proximal degrees of freedom in the shoulder and elbow position the wrist in $\mathbb{R}^3$. Three orthogonal rotational axes (humeral rotation, wrist flexion/extension, and wrist adduction/abduction) intersect at a point in the wrist near the base of the palm and, thus, generate $SO(3)$. Therefore, the ability to position and orient the hand are decoupled to a large degree.

Most closed-form solutions exploit structure in the kinematic design and employ algebraic or geometric methods [302]. Of these approaches, geometric methods attempt to identify opportunities to decompose the inverse kinematic problem

for a particular manipulator into several lower-dimensional problems. The properties underlying Pieper's conditions are an example of such an approach wherein the problem is decomposed into inverse position and inverse orientation components. Instead of a comprehensive treatment of these techniques, we will pose a particularly simple example in the context of inverse kinematic solutions applicable to Roger's two degree of freedom planar arm that first appeared in [72].

EXAMPLE: Geometric Inverse Kinematic Solution for the Planar 2R Manipulator

A kinematic definition of Roger's planar manipulator is presented in Figures 4.2 and 4.7. In this example, we construct a complete inverse kinematic solution for the arm. Recall that by "complete" in this case, we mean that every point on the interior of the reachable workspace should yield two unique solutions.

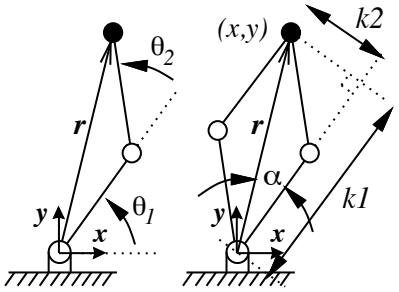


Figure 4.11 *A geometric construction for simplifying the inverse kinematic solution.*

Figure 4.11 defines the geometric constructions used to solve the inverse kinematics. Once again, we use the shorthand $c_i = cos(\theta_i)$, $s_i = sin(\theta_i)$ and identities

$$sin(\alpha \pm \beta) = sin\alpha\ cos\beta \pm cos\alpha\ sin\beta, \text{ and}$$

$$cos(\alpha \pm \beta) = cos\alpha\ cos\beta \mp sin\alpha\ sin\beta$$

to simplify the result.

To eliminate θ_1, we note that $r^2 = x^2 + y^2$ and from the forward kinematics, $x = l_1c_1 + l_2c_{12}$ and $y = l_1s_1 + l_2s_{12}$. Therefore,

$$\begin{aligned} r^2 = x^2 + y^2 &= l_1^2c_1^2 + 2l_1l_2c_1c_{12} + l_2^2c_{12}^2 + l_1^2s_1^2 + 2l_1l_2s_1s_{12} + l_2^2s_{12}^2, \quad or \\ r^2 &= l_1^2 + 2l_1l_2c_2 + l_2^2. \end{aligned}$$

This geometric construction reveals that r does not depend on θ_1, rearranging terms to solve for c_2,

$$c_2 = \frac{r^2 - l_1^2 - l_2^2}{2l_1l_2}. \tag{4.6}$$

*The quotient on the right side of Equation 4.6 determines the geometric feasibility of target **r** for the 2R mechanism with link lengths l_1 and l_2. For an inverse kinematic solution to exist, Equation 4.6 must produce a value for c_2 in the interval $[-1, +1]$.*

*If **r** is reachable, there will be two (possibly repeated) manipulator configurations for $-\pi < \theta_2 \leq +\pi$. The only exception to this rule is when the goal is at the origin and $l_1 = l_2$. In this case, $\theta_2 = \pm\pi$ and there are an infinite number of θ_1 configurations that can be used to reach the goal with any endpoint orientation in the range from $-\pi$ to π.*

To solve for both θ_2 solutions, we note that $s_2^2 + c_2^2 = 1$, so that $s_2^{+/-} = +/- (1 - c_2^2)^{1/2}$, and

$$\theta_2^{+/-} = tan^{-1} \frac{s_2^{+/-}}{c_2}. \tag{4.7}$$

The θ_1 that correspond to these θ_2 solutions are determined by introducing auxiliary variables k_1 and k_2 (Figure 4.11),

$$k_1 = rc_\alpha = l_1 + l_2 c_2, \text{ and} \tag{4.8}$$
$$k_2^{+/-} = rs_\alpha = l_2 s_2^{+/-}, \tag{4.9}$$

so that,

$$\alpha^{+/-} = tan^{-1} \frac{k_2^{+/-}}{k_1}. \tag{4.10}$$

Finally,

$$x = k_1 c_1 - k_2 s_1 = (rc_\alpha)c_1 - (rs_\alpha)s_1 = rcos(\alpha + \theta_1)$$
$$y = k_1 s_1 + k_2 c_1 = (rc_\alpha)s_1 + (rs_\alpha)c_1 = rsin(\alpha + \theta_1)$$

and

$$tan(\alpha + \theta_1) = \frac{rsin(\alpha + \theta_1)}{rcos(\alpha + \theta_1)} = \frac{y}{x}$$

so that,

$$\theta_1^{+/-} = tan^{-1} \frac{y}{x} - \alpha^{+/-}. \tag{4.11}$$

These kinematic relations are summarized in the algorithm presented in Figure 4.12 [72].

GIVEN (x,y) endpoint position goal:

$r^2 = x^2 + y^2$
$c_2 = (r^2 - l_1^2 - l_2^2)/(2l_1 l_2)$
if $(-1 \leq c_2 \leq +1)$
  $s_2^{+/-} = +/- (1 - c_2^2)^{1/2}$
  $\theta_2^{+/-} = tan^{-1} \left(s_2^{+/-} / c_2 \right)$
  $k_1 = l_1 + l_2 c_2$
  $k_2^{+/-} = l_2 s_2^{+/-}$
  $\alpha^{+/-} = tan^{-1} \left(k_2^{+/-} / k_1 \right)$
  $\theta_1^{+/-} = tan^{-1} (y/x) - \alpha^{+/-}$
else "out of reach"

Figure 4.12 *The complete inverse kinematic mapping for the 2R, planar manipulator.*

□

Forward and inverse kinematic transformations contribute to a *proprioceptive* sense of space. Proprioception refers to the sense of the position and movement of one's body—including the orientation and viewpoint of externally directed

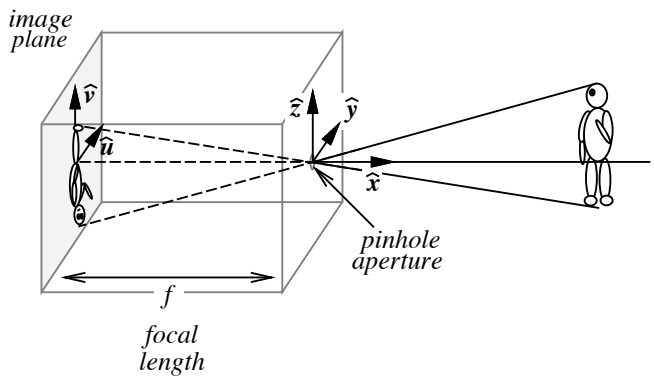


Figure 4.13 *The pinhole camera geometry.*

sensors. Combined with other sensor modalities, like tactile sensing and vision, proprioceptive information can contribute to the acquisition of *exteroceptive* information regarding the space outside the robot's body. One of the most important sources of this kind of information is derived from stereoscopic vision.

4.5 Kinematics of Stereo Reconstruction

Apart from issues of computer vision (Chapter **??**) and signal processing (Chapter 8), stereo vision is largely a kinematic issue. In Section 4.4, forward kinematic functions were used to map the configuration variables of a robot manipulator into Cartesian coordinates. This mapping function can be used to map tactile events on the surface of the manipulator to the Cartesian places where contacts occur. Forward kinematic relations can also be used to localize other sensory events. For example, oculomotor configuration variables corresponding to binocular visual events can be mapped to the Cartesian location of a light source. In this section, we will introduce a simple model of the imaging geometry and the kinematic equations underlying stereoscopic reconstruction.

4.5.1 Pinhole Camera - Projective Geometry

Figure 4.13 illustrates a pinhole camera, constructed by making a single small hole in an opaque enclosure. The "pinhole" is positioned at the origin of the camera's coordinate frame in the diagram and provides the only pathway for electromagnetic energy to enter the enclosure. Rays of light travel in straight line paths through the opening to strike the image plane at $x = -f$, where f is called the focal length. The process creates an image of environmental light sources. Such a *projective geometry* preserves the topology of the light sources from the *perspective* of the camera (the $+\hat{\boldsymbol{x}}$ axis in Figure 4.13).

Pinhole cameras can be found in the writing of philosophers, astronomers, and scientists going back thousands of years in references from China (Mo-Ti 5^{th} century BC), Ancient Greece (Aristotle ca. 350 BC), and Persia (Alhazen of Basra,

10^{th} century). It was described in the notebooks of Leonardo Da Vinci in the 15^{th} century and the 16^{th} century Dutch artist Johannes Vermeer, one of the early realists, used the "camera obscura" to create his remarkably accurate interior still lifes. With the advent of optical lenses and good quality light sensitive materials early in the 19^{th} century, the camera obscura evolved into the modern camera.

The *perspective projection* is defined by the geometric construction illustrated in Figure 4.13. By similar triangles, world coordinate (x, y, z) projects to image plane coordinate, $u = -fy/x$ and $v = -fz/x$. The positive directions for $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$ are defined by the positive directions of $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$, respectively and all valid world coordinates have positive $\hat{\boldsymbol{x}}$ components. Consequently, the image of these sources is inverted on the focal plane. Moreover, scale on the image plane (the ratio of the image plane distance to the Cartesian distance) is inversely proportional to range x. The resulting depth dependent projective distortion explains why train tracks appear to come together on the horizon at a position called the *vanishing* point. Projective scale distortion diminishes as a subject with finite depth approaches $x = \infty$. Under these circumstances, the projection is approximately *orthographic*. In practical applications, perspective distortion is generally insignificant if the depth of the object is small relative to its range. Under these circumstances, the subject is said to have *shallow structure* and the projection can be assumed to be essentially orthographic.

4.5.2 Binocular Localization: Forward Kinematics

The perspective projection equations map light rays reflected from three dimensional objects onto a two dimensional image plane. Information about depth in the scene is lost in this projection and cannot be recovered without additional information. A stereo pair of cameras with two different perspectives on the scene can reconstruct some of the three dimensional structure lost in the projection.

EXAMPLE: Stereo Localization in the Plane

Consider Roger's planar binocular imaging geometry illustrated in Figure 4.14. A pair of cameras separated by $2d$ *in the* $\hat{\boldsymbol{y}}$ *direction are illustrated as opaque enclosures at the bottom of the figure with pinhole apertures aimed along parallel gaze directions,* $\hat{\boldsymbol{x}}_L$ *and* $\hat{\boldsymbol{x}}_R$*. The reference frame midway between the pinholes is called the stereo (or cycloptic) coordinate frame. The goal is to recover the* (x, y) *coordinate of point* $\boldsymbol{p}$ *in the stereo frame. This is a special case where the two cameras have fixed gazes mutually parallel to the cycloptic* $\hat{\boldsymbol{x}}$ *axis. In this case depth is encoded exclusively in the "disparity" between the image coordinates of point* $\boldsymbol{p}$ *on the left and right image planes. Under these conditions:*

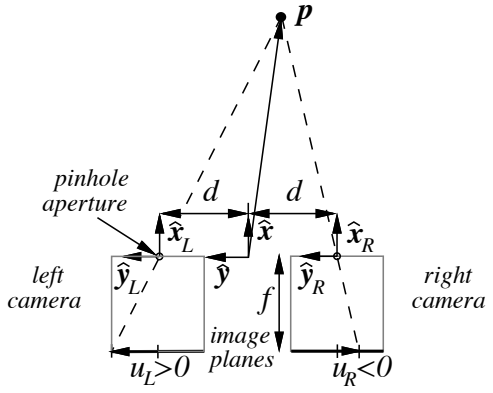


Figure 4.14 *Top view of a 2D stereo geometry where depth is encoded exclusively in stereo disparity.*

$$u_L = \frac{-f(y-d)}{x}, \qquad u_R = \frac{-f(y+d)}{x},$$

so that

$$xu_L = -f(y-d), \qquad xu_R = -f(y+d).$$

The difference of these functions yields $x(u_L - u_R) = 2df$ *and therefore,*

$$x = \frac{2df}{(u_L - u_R)}\ . \tag{4.12}$$

The range estimate is inversely proportional to the stereo disparity, $(u_L - u_R)$*, which is positive for any feature* $\mathbf{p}$ *in front of the camera pair. The numerator of Equation 4.12 is a positive constant. As result, the range goes to positive infinity as the disparity goes to zero and regions with equal disparity are equally distant.*

It is relatively straightforward to generalize this result to cases where the cameras can pan independently (i.e. that can "verge" toward feature $\mathbf{p}$*). Figure 4.15 shows the new geometry for this situation and defines variables* θ *and* ϕ *for the left camera. We assume that the verge degrees of freedom are located at the pinhole apertures and rotate about axes parallel to the stereo frame* $\hat{\mathbf{z}}$ *axis. The new variable,* γ*, is the sum of the verge angle,* θ*, and the angular offset of the*

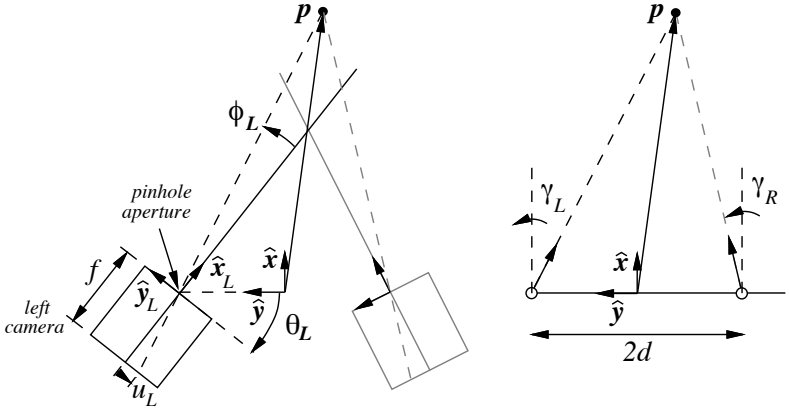


Figure 4.15 *The binocular imaging geometry with independently panning cameras. The figure on the right summarizes the stereo geometry in terms of parameter* $\gamma = \theta + \phi$*, the sum of the eye configuration,* θ*, and the angular offset from the fovea,* ϕ*.*

feature $\mathbf{p}$ *from the image center. The illustration on the right of Figure 4.15 identifies the governing geometry of this stereo system. It is relatively straight forward to solve for the spatial coordinates of feature* $\mathbf{p}$ *under these conditions.*

$$\begin{aligned} x &= 2d\frac{cos(\gamma_R)cos(\gamma_L)}{sin(\gamma_R - \gamma_L)} \\ y &= d + 2d\frac{cos(\gamma_R)sin(\gamma_L)}{sin(\gamma_R - \gamma_L)} \end{aligned} \tag{4.13}$$

It is left as an exercise for the reader to verify these triangulation equations for this geometry. □

4.6 Hand-Eye Kinematic Transformations

In addition to contact interactions with the world, kinematic structures are a means of delivering sensors to places where useful information is accessible. Supported by joint angle feedback, degrees of freedom in the arms, hands, and head can be used to orient tactile and visual sensors to the world geometry. Visual signals are subject to line-of-sight constraints and tactile signals are subject to reachability constraints, but together these independent sensors provide richer and more complete information than is possible with either one alone.

Figure 4.16 summarizes the kinematic mapping functions for the arm/eye subsystems of Roger-the-Crab developed in this chapter. It illustrates how they are related to each other and to the Cartesian task space.[4] The middle panel shows a regular Cartesian grid describing the space around the robot. The reachable subset of the world is green and can be observed using both visual and tactile sensors. The region of space that is not reachable by either arm is colored red—environmental stimuli in this region are accessible only by vision.

The bottom panels of Figure 4.16 remap the reachable subset of Cartesian space onto the configuration variables of Roger's arms. Joint angles in the arm can rotate continuously, so these maps are *toroidal*—the top/bottom and left/right edges of the configuration space wrap around at $\pm\pi$. Figure 4.16 shows how the reachable subset of Cartesian space is deformed when parameterized in terms of the configuration variables of Roger's arms.

The top panel in the figure shows the results of projecting Cartesian space onto the configuration parameters of Roger's stereo oculomotor system. The world beyond arm's length is red and maps to a narrow diagonal where the gaze of the two eyes is convergent, but very nearly parallel i.e. $\gamma_L \approx \gamma_R$. We observe a kind of "fish eye" stereo effect that emphasizes a region of space directly in front of the pair of eyes. Lateral and distant regions of space are compressed into a relatively small range of stereo configuration parameters.

The regular Cartesian grid lines in Figure 4.16 are deformed when they are projected into these configuration spaces. Both the visual and proprioceptive maps preserve the topology of the Cartesian space but as a result of this deformation, configuration space observers vary in precision and sensitivity depending on where they are "looking" and how they are configured.

The kinematic mapping functions derived in earlier sections are shown in Figure 4.16 as well. Function $f(x, y)$ is the pinhole camera projections (Section 4.5.1) for the stereo pair that map a coordinate in Cartesian space to visual parameters γ_L and γ_R. Its inverse, $f^{-1}()$, represents the stereo triangulation equations (Section 4.5.2). Functions $g(\theta_1, \theta_2)$ and $g^{-1}(x, y)$ are the forward and inverse kinematic transformations, respectively, for Roger's arms (Section 4.4). These mappings are written for the left arm, $g_L()$, and the right arm, $g_R()$. Compositions of these transformations can be used to map directly from oculomotor to arm configurations and vice versa.

Goals in one motor map are strongly associated with goals in the other maps—a visual stimulus, for example, is associated with arm configurations where spatially co-located tactile events are likely to also occur. All the blue

[4] Figure 4.16 was constructed by scanning a discretized configuration space for the limb in question and using forward kinematic functions to index into a Cartesian map. The color of the Cartesian map at that location is copied to the corresponding configuration space bin. Small asymmetries in the figure are artifacts of the discrete model and the projection process.

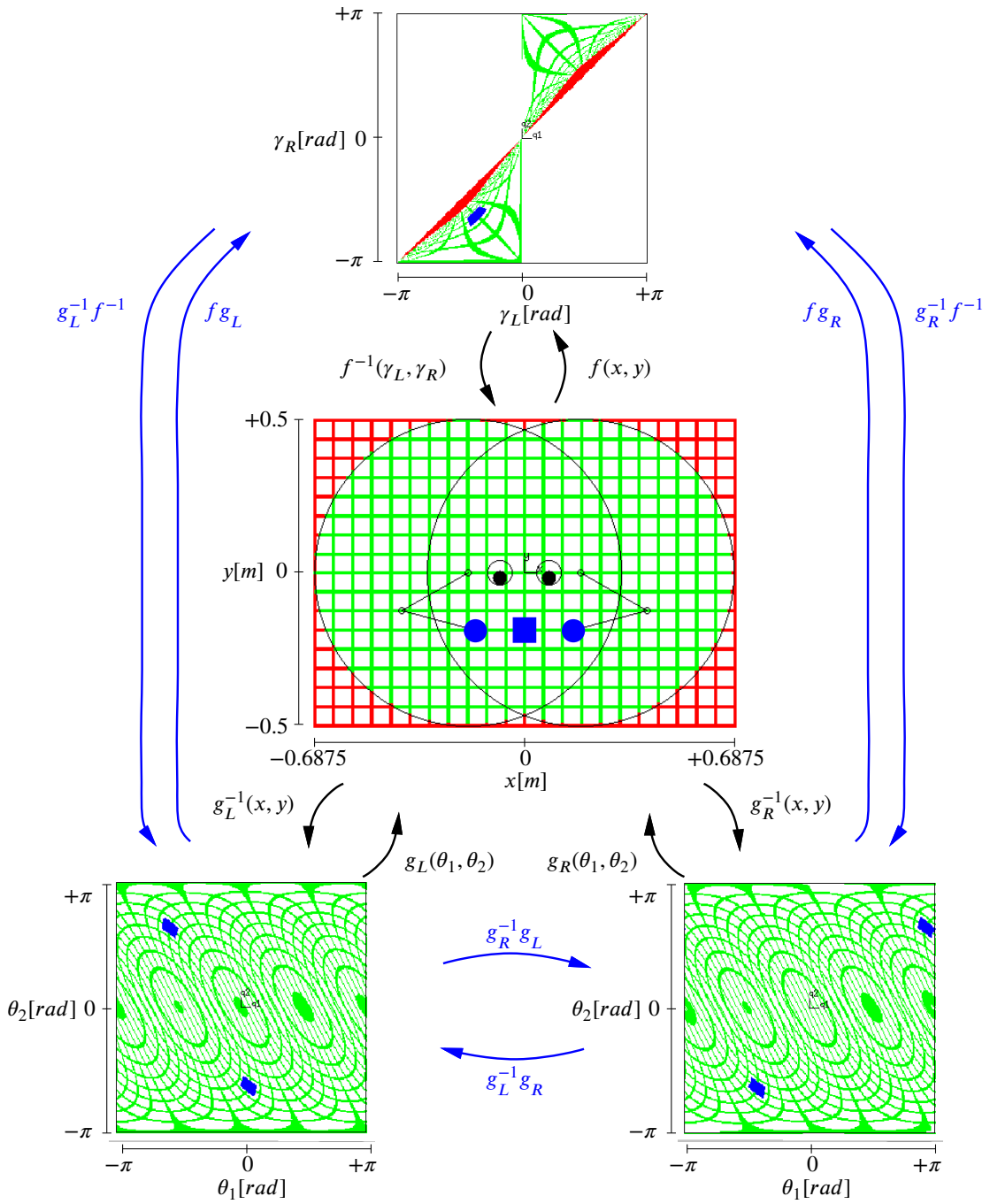


Figure 4.16 *Multi-sensor models of space and hand-eye coordination. This illustration uses a different coordinate system than other Roger examples and arms/hands have been removed from images for clarity.*

features in the configuration space maps refer to the single blue square in the Cartesian map. A unique observation in the oculomotor space is associated with up to two configurations of each arm. This is the basis for a multi-modal and fused interpretation of space.

Immediately after birth, human infants use a relatively mature binaural auditory sense of space to bootstrap reaching and visual orientation skills [59, 23, 24]. During the infant's first year (and beyond), cells from layers of tissue in the visual, auditory, and somatosensory cortices project via neural processes to associated cells in the motor cortex where co-located stimuli are expected. During this period, developmental reflexes orient the neck and hand so as to place the hand in the field of view. Among other things, this reflex generates training data and bootstraps the acquisition of hand-eye projections like this (Section 9.3.2). All of these precursor skills and cumulative perceptual abilities support more complex skills in a developmental progression—from spatial mapping, to objects, to grasping and manipulation, and to words (Section 9.4). The first complete sense of the space outside the infant is supported by correlations between sight, sound, and touch that capture mappings like those depicted in Figure 4.16.

4.7 Kinematic Conditioning

Techniques for evaluating the *kinematic condition* of a robot rely on methods for analyzing the ability of the mechanism to deliver velocities as well as positions. Tools from linear analysis provide useful insight, however, forward kinematic transformations are generally nonlinear. To make use of these tools, the forward kinematic equations must be linearized around an operating configuration using the *Jacobian* of the nonlinear kinematic equations.

4.7.1 Jacobian

Nonlinear forward kinematic functions $\boldsymbol{r}(\boldsymbol{q})$, like those in Equations 4.5 and 4.13, map coordinates in configuration space, $\boldsymbol{q}$, to Cartesian space. Assume that these functions are *analytic* functions, so that they can be written in the neighborhood of $\boldsymbol{q} = \boldsymbol{a}$ in the form of a Taylor series:

$$\boldsymbol{r}(\boldsymbol{a} + d\boldsymbol{q}) = \boldsymbol{r}(\boldsymbol{a}) + \frac{d\boldsymbol{q}}{1!}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} + \frac{d\boldsymbol{q}^2}{2!}\frac{\partial^2 \boldsymbol{r}}{\partial \boldsymbol{q}^2} + \ldots \tag{4.14}$$

The first-order linear approximation of the function $\boldsymbol{r}(\boldsymbol{q})$ near $\boldsymbol{q} = \boldsymbol{a}$ is extracted from the Taylor series by ignoring higher-order terms,

$$d\boldsymbol{r}|_{q=a} \approx \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}}\right|_{q=a} d\boldsymbol{q} = \mathbf{J}|_{q=a}\ d\boldsymbol{q}. \tag{4.15}$$

Equation 4.15 maps differential displacements in configuration space, $d\boldsymbol{q}$, into differential displacements in Cartesian space, $d\boldsymbol{r}$. Equivalently, it transforms input velocities to output velocities. The matrix of partial derivatives, $\partial \boldsymbol{r}/\partial \boldsymbol{q}$ is the *Jacobian*, $\mathbf{J}$, named after the mathematician Carl Gustav Jacobi. It represents the tangent hyperplane of the function $\boldsymbol{r}(\boldsymbol{q})$ in the neighborhood $d\boldsymbol{q}$ surrounding $\boldsymbol{q} = \boldsymbol{a}$. The Jacobian is an accurate local approximation of the original nonlinear system in the limit as $d\boldsymbol{q} \to \mathbf{0}$ and it can be analyzed using powerful tools applicable to linear systems.

4.7.2 The Manipulator Jacobian

The Cartesian velocity of the manipulator is derived by differentiating the forward kinematic mapping and, in general, depends on both the pose and the velocity of the joints in the mechanism. The analysis can be applied at any point on an articulated structure, however, without loss of generality, we consider the velocity of the endpoint of the kinematic chain and illustrate the idea using the planar 2R manipulator whose forward kinematic equations were derived in Section 4.4.1.

EXAMPLE: First Order Velocity Control for the Planar 2R Manipulator

The $x-y$ components of the forward kinematic function for the planar 2R manipulator (Equation 4.5) are repeated here:

$$\begin{aligned} r_x &= l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2), \\ r_y &= l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2). \end{aligned}$$

The first derivative of these equations is written

$$\begin{aligned} dr_x &= -l_1 sin(\theta_1) d\theta_1 - l_2 sin(\theta_1 + \theta_2) d\theta_1 - l_2 sin(\theta_1 + \theta_2) d\theta_2, and \\ dr_y &= l_1 cos(\theta_1) d\theta_1 + l_2 cos(\theta_1 + \theta_2) d\theta_1 + l_2 cos(\theta_1 + \theta_2) d\theta_2, \end{aligned} \tag{4.16}$$

which can be rewritten to emphasize the locally linear relationship:

$$d\boldsymbol{r} = \begin{bmatrix} dr_x \\ dr_y \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \end{bmatrix} = \mathbf{J} d\boldsymbol{\theta}. \tag{4.17}$$

The manipulator Jacobian $\mathbf{J}$ is a 2×2 matrix that defines the locally linear mapping from a 2-dimensional displacement/velocity in configuration space $[d\theta_1 \; d\theta_2]^T$ to a 2-dimensional displacement/velocity in Cartesian space $[dx \; dy]^T$ at the manipulator endpoint.

With no joint range limits, this manipulator can execute continuous rotations in both θ_1 and θ_2—that is, it can execute arbitrary velocities $[\dot{\theta}_1 \; \dot{\theta}_2]$ from any configuration $\boldsymbol{\theta}$ in configuration space. Suppose that the set of executable velocities forms the unit disk in configuration space centered on configuration $\boldsymbol{\theta}$ as illustrated on the left side of Figure 4.17. In general, the projection into Cartesian space represented by the Jacobian deforms this set as illustrated on the right side of the diagram.

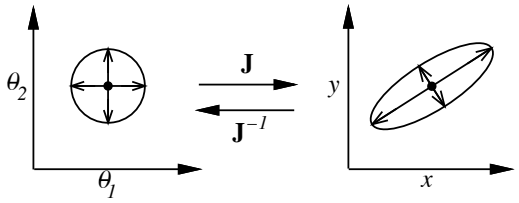


Figure 4.17 *The Jacobian of the 2R manipulator transforms sets of executable velocities in configuration space from posture (θ_1, θ_2) into sets of achievable velocities in Cartesian space from endpoint position (x, y).*

The determinant (Appendix A.2) of the Jacobian in Equation 4.17 is a scalar value with units $[m^2/rad^2]$ proportional

to the 'scale' of the transformation. The determinant for the 2R manipulator Jacobian is written

$$\begin{vmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{vmatrix} = -l_1 l_2 s_1 c_{12} - l_2^2 s_{12} c_{12} + l_1 l_2 c_1 s_{12} + l_2^2 c_{12} s_{12},$$
$$= l_1 l_2 (c_1 s_{12} - s_1 c_{12}) = l_1 l_2 s_2. \tag{4.18}$$

It describes the degree to which velocities in configuration space are amplified or attenuated as they are projected into velocities in the Cartesian plane. A determinant of zero indicates that the Jacobian is singular and has lost rank. In general, this means that an n-dimensional input maps (locally) to a lower-dimensional manifold of the output space.

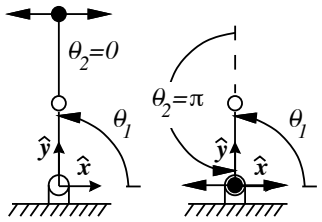


Figure 4.18 *Singularities in the 2R manipulator.*

manifold of the output space. From Equation 4.18, the 2R manipulator Jacobian is singular when $sin(\theta_2) = 0$, or when $\theta_2 = 0, \pi$ (Figure 4.18). These configurations correspond to endpoint positions on the outer and inner boundary of the reachable workspace. In these postures, the manipulator can not generate arbitrary velocities in the x-y plane. Figure 4.18 shows that the achievable endpoint velocities for these singular configurations are limited to a 1-dimensional subset of the Cartesian plane.

In non-singular configurations, however, the manipulator Jacobian is full rank and is, therefore, invertible. Suppose that Roger's arm is required to execute an endpoint velocity of $v = 1\ m/sec$ in the $\hat{\mathbf{x}}$ direction as illustrated in Figure 4.19. Inverting Equation 4.17 (Appendix A.2) yields:

$$\mathbf{J}^{-1} = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix}$$

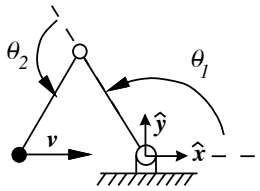


Figure 4.19 *The 2D, planar manipulator and a specific endpoint velocity command.*

so that,

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{l_1 l_2 s_2} \begin{bmatrix} l_2 c_{12} \\ -l_1 c_1 - l_2 c_{12} \end{bmatrix} \frac{rad}{sec}. \tag{4.19}$$

This closed-form relation defines the joint angle velocities that produce the target endpoint velocity for every pose (θ_1, θ_2) of the manipulator. As the arm approaches singular configurations like those in Figure 4.18, the inverse of the Jacobian becomes ill-conditioned because $1/l_1 l_2 s_2$ goes to infinity. As a consequence, Equation 4.19 predicts that the fully extended 2R manipulator requires infinite joint space velocities to withdraw the endpoint radially (along the $-\hat{\mathbf{y}}$ direction in Figure 4.18) toward the origin. The reader can easily verify that this is not the correct conclusion using their own arms. Omitting the higher-order terms in Equation 4.14 and relying solely on a first-order approximation can introduce significant inaccuracies and the predicted behavior near singularities can misleading in important ways.

□

Manipulability Ellipsoid

The scalar determinant is only a coarse measure of manipulator conditioning—the manipulator Jacobian also provides an opportunity to examine the spatial character of the velocity transform. If we assume that joint angle velocities in the manipulator are limited to the unit (hyper)sphere, $\dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}} \leq 1$, then we can derive the quadratic form (Appendix A.6) describing the corresponding envelope of Cartesian endpoint velocities.

$$\dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}} = (\mathbf{J}^{-1}\dot{\boldsymbol{r}})^T(\mathbf{J}^{-1}\dot{\boldsymbol{r}}) = \dot{\boldsymbol{r}}^T[(\mathbf{J}^{-1})^T\mathbf{J}^{-1}]\,\dot{\boldsymbol{r}} = \dot{\boldsymbol{r}}^T(\mathbf{J}\mathbf{J}^T)^{-1}\,\dot{\boldsymbol{r}} \leq 1 \tag{4.20}$$

The rightmost inequality in Equation 4.20 defines the quadratic form we are interested in. It describes the Cartesian *manipulability ellipsoid*—a set of Cartesian velocities generated by mapping the unit (hyper)sphere of joint space velocities through the manipulator Jacobian. Appendix A.6 shows how the ellipsoid is determined from the eigenvalues and eigenvectors of $\mathbf{J}\mathbf{J}^T$.

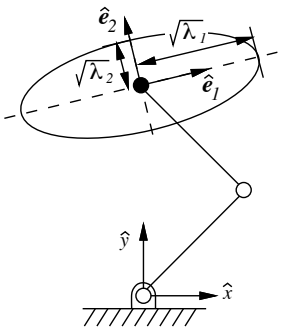


Figure 4.20 *The velocity ellipsoid in terms of the eigenvalues and eigenvectors of* $\mathbf{J}\mathbf{J}^T$ *for the planar 2R manipulator.*

The velocity ellipsoid is pictured for the 2R planar manipulator in Figure 4.20. The diagram illustrates the relative ability of the arm to generate endpoint velocities from this configuration in different directions. Unit vectors, $\hat{\boldsymbol{e}}_1$ and $\hat{\boldsymbol{e}}_2$, are the eigenvectors of $\mathbf{J}\mathbf{J}^T$. They define the principal axes of the transformation. The relative amplification of velocity in these directions is proportional to the square root of the corresponding eigenvalues λ_1 and λ_2 of $\mathbf{J}\mathbf{J}^T$. In this light, the directions in which the manipulator is best at transforming small joint angle velocities $\dot{\boldsymbol{\theta}}$ into large Cartesian velocities $\dot{\boldsymbol{r}}$ is identified by the eigenvector with the largest eigenvalue and the scale is proportional to the square root of that eigenvalue.

Conversely, the eigenvector with the smallest eigenvalue identifies the direction in Cartesian space where the precision of the mapping is optimized. Imagine an encoder that measures joint angles positions over the interval $[0, 2\pi)$ with 8 bits of precision. In this case, the encoder resolves $2\pi/2^8 = 2\pi/256$ radians per encoder "tick." Projecting this resolution through the Jacobian yields

$$\mathbf{J}\left[\frac{m}{rad}\right] \quad \frac{2\pi}{256}\left[\frac{rad}{tick}\right] = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix} \frac{2\pi}{256}\left[\frac{m}{tick}\right].$$

Thus, the eigenvector with the smallest eigenvalue identifies the direction in Cartesian space where the sensor resolves the smallest Cartesian displacements per encoder tick.

Force Ellipsoid

The work output produced at the endpoint of a mechanism that is in contact with the world is the product of endpoint force $\boldsymbol{f}$ and distance $d\boldsymbol{r}$ traveled. If we assume no losses, then the work input in configuration space must be equal to the work output in Cartesian space [72]

$$\boldsymbol{\tau}^T d\boldsymbol{\theta} = \boldsymbol{f}^T d\boldsymbol{r}.$$

However, since $d\boldsymbol{r} = \mathbf{J}d\boldsymbol{\theta}$, we can write

$$\boldsymbol{\tau}^T d\boldsymbol{\theta} = \boldsymbol{f}^T \left[\mathbf{J}d\boldsymbol{\theta}\right],$$

so that,

$$\boldsymbol{\tau}^T = \boldsymbol{f}^T\mathbf{J}, \quad \text{or} \quad \boldsymbol{\tau} = \mathbf{J}^T\boldsymbol{f}. \tag{4.21}$$

The manipulator Jacobian, therefore, describes the transformation from joint velocity to Cartesian velocity *and* the mapping from Cartesian endpoint forces to joint torques.

As a consequence, the velocity ellipsoid (Equation 4.20) has a direct analog in the quadratic form representing the *force ellipsoid*

$$\boldsymbol{\tau}^T\boldsymbol{\tau} = (\mathbf{J}^T\boldsymbol{f})^T(\mathbf{J}^T\boldsymbol{f}) = \boldsymbol{f}^T(\mathbf{J}\mathbf{J}^T)\boldsymbol{f} \leq 1. \tag{4.22}$$

Thus, unit hypersphere $\boldsymbol{\tau}^T\boldsymbol{\tau}$ maps through the manipulator Jacobian to the Cartesian force ellipsoid defined by the eigenvalues and eigenvectors of $(\mathbf{J}\mathbf{J}^T)^{-1}$. The eigenvectors of $\mathbf{J}\mathbf{J}^T$ and $(\mathbf{J}\mathbf{J}^T)^{-1}$ are identical and the eigenvalues of $\mathbf{J}\mathbf{J}^T$ (velocity amplifier) are reciprocals of the eigenvalues of $(\mathbf{J}\mathbf{J}^T)^{-1}$ (force amplifier). Therefore, once the eigenvectors and eigenvalues of $\mathbf{J}\mathbf{J}^T$ are established, both the velocity and force capabilities of the manipulator can be determined.

EXAMPLE: Velocity and Force Ellipsoids for Roger

Velocity and force conditioning ellipsoids for Roger's arms are illustrated in Figure 4.21. Close inspection verifies that the eigenvectors are the same and that eigenvalues are reciprocal. For velocity (left), a large eigenvalue of $\mathbf{J}\mathbf{J}^T$ amplifies relatively small $\dot{\boldsymbol{\theta}}$ inputs into large $\dot{\boldsymbol{r}}$ outputs along the corresponding eigenvector. This is evident in Figure 4.21 near $x = 0.5$ [m], where the left arm is nearly fully extended and joint velocities are mapped efficiently through the Jacobian to Cartesian velocities in the $\pm\hat{\mathbf{y}}$ direction. Conversely, the arm is quite good at detecting small Cartesian displacements of the endpoint in the $\hat{\mathbf{x}}$ direction as relatively large displacements of the joints in the arm.

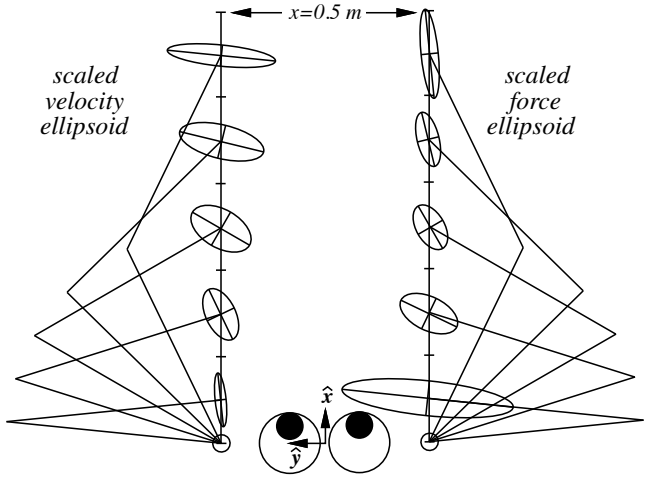


Figure 4.21 *The manipulator conditioning ellipsoids: velocity ellipsoids from $(\mathbf{J}\mathbf{J}^T)$ are on the left; force ellipsoids on the right.*

The nearly extended right arm demonstrates the same kind of kinematic efficiency in the force generating capacity of the arm mechanism. Unlike the capacity for Cartesian velocity, when the robot arm is extended, it is quite good at converting relatively small joint torques into relatively large endpoint forces. Loads in the $\hat{\mathbf{x}}$ direction on the extended arm are carried primarily in the links (skeleton) and not in the actuators (musculature). Conversely, in directions with small eigenvalues, small external loads produce relatively large joint torques. As result, these robot configurations are capable of detecting and precisely controlling relatively small interaction forces at the expense of strength.

□

The tools introduced in this section characterize (to first order) how kinematic devices produce Cartesian velocities and forces. The Jacobian is the key to this analysis—it can be viewed as a configuration dependent velocity amplifier that describes kinematic mappings. In the next section, the same analysis will be applied to stereo triangulation. In this analysis, however, the emphasis is not force and velocity, instead, we focus on how visual acuity depends on the kinematic configuration of the oculomotor system.

4.7.3 Stereo Localizability

In Section 4.5.2, stereo reconstruction is posed as a kinematics problem. Stereo triangulation equations are used to transform the configuration of a pair of eyes and the signal on the image plane (collectively, an oculomotor configuration) into an estimate of the Cartesian coordinate of the light source. Therefore, a differential error in the oculomotor configuration will map, via the oculomotor Jacobian, to a differential error in the Cartesian location of the light source.

Errors in the oculomotor configuration arise from errors in the mechanism, imprecision in joint angle sensors, optical aberrations in cameras, and finite resolution on the image plane. The precision of the stereo system can be analyzed using the techniques developed in the previous section for the manipulator, namely, by mapping a circular error bound on the input through the linearized triangulation equations to an anisotropic Cartesian ellipsoid.

EXAMPLE: Roger's Oculomotor Jacobian and Stereo Localizability

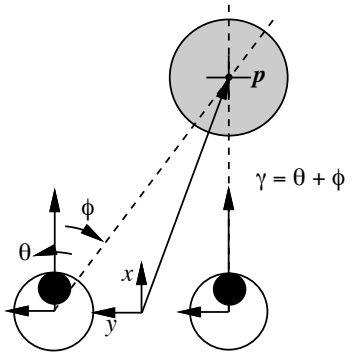


Figure 4.22 *Triangulation parameters for Roger's stereo system.*

Figure 4.22 reproduces the important geometry from the binocular system treated in Section 4.5 in the context of Roger's stereo visual system. The stereo triangulation results for this geometry are repeated here in Equation 4.23.

$$\begin{aligned} x &= 2d\frac{cos(\gamma_R)cos(\gamma_L)}{sin(\gamma_R-\gamma_L)} \\ y &= d+2d\frac{cos(\gamma_R)sin(\gamma_L)}{sin(\gamma_R-\gamma_L)} \end{aligned} \tag{4.23}$$

These equations localize point $\mathbf{p}$*. To estimate the sensitivity of this result to small errors in the imaging geometry, we compute the oculomotor Jacobian.*

The triangulation equations are differentiated to yield the linear form $d\mathbf{p} = \mathbf{J}d\boldsymbol{\gamma}$*. The partial derivative of Equa-*

tion 4.23 with respect to configuration variables, γ_L and γ_R, yields:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \frac{\partial x(\gamma_L,\gamma_R)}{\partial \gamma_L} & \frac{\partial x(\gamma_L,\gamma_R)}{\partial \gamma_R} \\ \frac{\partial y(\gamma_L,\gamma_R)}{\partial \gamma_L} & \frac{\partial y(\gamma_L,\gamma_R)}{\partial \gamma_R} \end{bmatrix} \begin{bmatrix} d\gamma_L \\ d\gamma_R \end{bmatrix},$$

$$= \frac{2d}{sin^2(\gamma_R - \gamma_L)} \begin{bmatrix} cos^2(\gamma_R) & -cos^2(\gamma_L) \\ sin(\gamma_R)cos(\gamma_R) & -sin(\gamma_L)cos(\gamma_L) \end{bmatrix} \begin{bmatrix} d\gamma_L \\ d\gamma_R \end{bmatrix}. \quad (4.24)$$

Equation 4.24 maps differential changes in a feature coordinate on the image plane, $d\boldsymbol{\gamma}^T = [d\gamma_L \; d\gamma_R]$, to differential changes in the estimated position of the source $d\mathbf{p}^T = [dx \; dy]$.

If the net error from all sources is bounded to a disk of radius k on the image plane, then

$$d\boldsymbol{\gamma}^T d\boldsymbol{\gamma} = d\mathbf{r}^T (\mathbf{J}\mathbf{J}^T)^{-1} d\mathbf{r} \leq k^2 \quad (4.25)$$

defines the stereo localizability ellipsoid. Roger's focal length is 64 pixels. If features on the image plane are accurate to 1 pixel, then the maximum angular error in $\boldsymbol{\gamma}$ is bounded by $k = tan^{-1}(1/64) = 0.01562379\ rad$. Therefore, the localizability ellipsoid has principal axes of length $k\sqrt{\lambda_1}$ and $k\sqrt{\lambda_2}$, where λ_1 and λ_2 are the eigenvalues of $\mathbf{J}\mathbf{J}^T$.

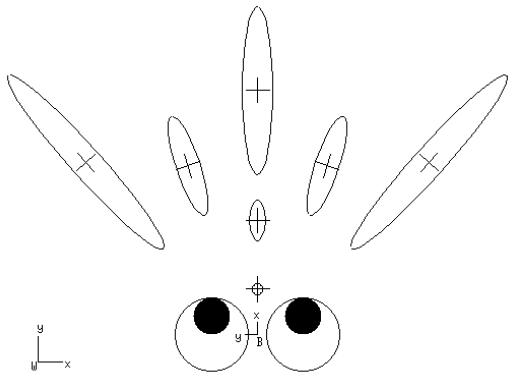


Figure 4.23 *Localizability ellipsoids for Roger's stereo geometry (scaled for visibility).*

Figure 4.23 illustrates localizability ellipsoids at several positions (marked by crosshairs) in Roger's field of view. The ellipsoids illustrate the shape and relative magnitude of the Cartesian error covariance. Generally, the lateral error is relatively small and the radial error can be significant depending on the position of the subject. The circular ellipsoid closest to the eyes is the position where stereo acuity is maximized.

□

4.8 Kinematic Redundancy

All the Jacobians introduced to this point have been square—the dimension of the input space has been equal to the dimension of the output space. Thus, the non-singular 2R manipulator in Section 4.7.2 generates output velocities in the two dimensional Cartesian plane. However, there is no requirement that input and output dimensions should be the same. In general, a forward velocity transform can be written in terms of the non-square Jacobian $\mathbf{J} \in \mathbb{R}^{m \times n}$ where $m \neq n$.

The velocity transformation $\dot{\boldsymbol{r}} = \mathbf{J}\dot{\boldsymbol{q}}$ with $\dot{\boldsymbol{q}} \in \mathbb{R}^n$ and $\dot{\boldsymbol{r}} \in \mathbb{R}^m$ is **redundant** if $\mathbf{J} \in \mathbb{R}^{m \times n}$ has more columns than rows ($n > m$) and $\mathbf{J}$ is full rank, i.e. the determinant of $\mathbf{J}\mathbf{J}^T \in \mathbb{R}^{m \times m}$ is non-zero. Under these conditions, the manipulator

has redundant degrees of freedom and an infinite number of inverse kinematic solutions exist. The set of such solutions is defined on a continuous $(n - m)$ dimensional manifold where $\dot{\boldsymbol{r}} = \mathbf{J}\dot{\boldsymbol{q}} = \mathbf{0}$ known as the *nullspace* of the redundant Jacobian. A sequence of movements in the nullspace trace out a path on the multi-dimensional *self-motion* manifold.

EXAMPLE: Self-Motion Manifold

The 3R planar manipulator is illustrated in Figure 4.24. With respect to endpoint positions in the $x - y$ plane, the 3R manipulator is redundant. The forward kinematic equations for this manipulator are:

$$\begin{aligned} x &= l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) + l_3 cos(\theta_1 + \theta_2 + \theta_3) \\ y &= l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) + l_3 sin(\theta_1 + \theta_2 + \theta_3) \end{aligned}$$

and the Jacobian is written

$$\mathbf{J} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix}.$$

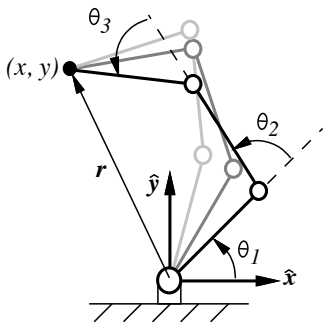


Figure 4.24 *Redundant solutions for the 3R manipulator at $x = -1$, $y = \sqrt{2}$.*

The Jacobian defines the velocity relation, $\dot{\mathbf{r}} = \mathbf{J}\dot{\boldsymbol{\theta}}$ and the self-motion manifold is determined by solving for the configuration space velocity, $\dot{\boldsymbol{\theta}}_{null}$, that causes no displacement in the endpoint, i.e. $\dot{\mathbf{r}} = 0$.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{4.26}$$

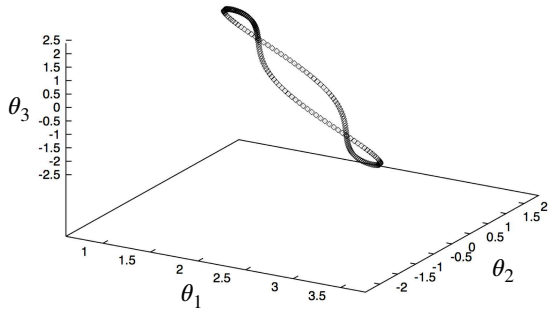


Figure 4.25 *Internal motions along the self-motion manifold of the 3R manipulator with $l_1 = l_2 = l_3 = 1$ and fixed endpoint position at $x = -1.0$, $y = \sqrt{2}$.*

Equation 4.26 represents a system of two equations in three unknowns that can be solved for internal motions by introducing one more constraint, namely, that $\dot{\theta}_1^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2 = 1$. Under these circumstances, one can verify that

$$\dot{\boldsymbol{\theta}}_{null} = \begin{bmatrix} l_2 l_3 sin(\theta_3) \\ -l_2 l_3 sin(\theta_3) - l_1 l_3 sin(\theta_2 + \theta_3) \\ l_1 l_2 sin(\theta_2) + l_1 l_3 sin(\theta_2 + \theta_3) \end{bmatrix}. \tag{4.27}$$

A trajectory on the self-motion manifold for a particular endpoint position is generated by following $\dot{\boldsymbol{\theta}}_{null}$ through a sequence of manipulator postures. One such trajectory is illustrated in Figure 4.25.

□

Internal motions introduce postural flexibility that supports multiple objective control designs—to avoid kinematic

singularities or meet force and velocity requirements, for example—to optimize subordinate tasks in the null space of the Cartesian endpoint velocity task.

The general solution for redundant systems (like Equation **??**) involves inverting non-square Jacobians $\mathbf{J} \in \mathbb{R}^{m \times n}$. When the Jacobian has more rows than columns ($m > n$), there are more constraint equations than unknowns and the system of equations is *over-constrained*. When there are fewer rows than columns ($m < n$) the system is under-constrained—there are fewer constraint equations than there are unknowns. Redundant Jacobians are examples of underconstrained systems and, consequently, there is no unique inverse. In the previous example concerning the planar 3R manipulator, there are many velocities in configuration space that map through $\mathbf{J}$ to the same velocity in Cartesian space.

In Appendix A.9, the pseudoinverse is introduced to address both under- and over-constrained systems using least squares optimal solutions. In the under-constrained (redundant) case, $\dot{\boldsymbol{r}} = \mathbf{J}\dot{\boldsymbol{q}}$ where $\dot{\boldsymbol{r}} \in \mathbb{R}^m$, $\dot{\boldsymbol{q}} \in \mathbb{R}^n$, and $m < n$ the result is the *right pseudoinverse*

$$\mathbf{J}^{\#} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1} \tag{4.28}$$

so that $\dot{\boldsymbol{q}} = \mathbf{J}^{\#}\dot{\boldsymbol{r}}$ yields a unique inverse kinematic solution that minimizes $\boldsymbol{q}^T\boldsymbol{q}$. Chapter 10 will return to the implications of redundancy in the control of complex robots by introducing tools that exploit redundancy in order to co-articulate multiple, simultaneous control objectives.

4.9 Exercises

1. **Inverting the Homogeneous Transform** - Given the general expression for the homogeneous transform and its inverse:

$$ {}_A\mathbf{T}_B = \left[\begin{array}{ccc|c} \hat{\boldsymbol{x}}_{3\times1} & \hat{\boldsymbol{y}}_{3\times1} & \hat{\boldsymbol{z}}_{3\times1} & \boldsymbol{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]_{4\times4} \qquad {}_B\mathbf{T}_A = [{}_A\mathbf{T}_B]^{-1} = \left[\begin{array}{ccc|c} \hat{\boldsymbol{x}}^T_{1\times3} & & & -\hat{\boldsymbol{x}}^T\boldsymbol{t} \\ \hat{\boldsymbol{y}}^T_{1\times3} & & & -\hat{\boldsymbol{y}}^T\boldsymbol{t} \\ \hat{\boldsymbol{z}}^T_{1\times3} & & & -\hat{\boldsymbol{z}}^T\boldsymbol{t} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]_{4\times4} $$

   Prove that ${}_B\mathbf{T}_A$ is the inverse of ${}_A\mathbf{T}_B$.

2. **The Homogeneous Structure of Operators in SE(3)** - Homogeneous transforms are not the most compact representation for describing spatial relationships, in fact, much of the information in a homogeneous transform is redundant. Given the following transform

$$ \mathbf{T} = \begin{bmatrix} ? & 0 & -1 & 0 \\ ? & 0 & 0 & 1 \\ ? & -1 & 0 & 2 \\ ? & 0 & 0 & 1 \end{bmatrix} $$

   use the structural properties of the homogeneous transform to solve for the unknown values.

3. **Spatial Algebra with the Homogeneous Transform** -

   (a) **Composing Homogeneous Transforms** - Consider three coordinate frames: W (world), A, and B, where;

$$ {}_W\mathbf{T}_A = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad {}_W\mathbf{T}_B = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

   i. Use the given transforms for ${}_W\mathbf{T}_A$ and ${}_W\mathbf{T}_B$ to calculate the transform from A to B, ${}_A\mathbf{T}_B$.

   ii. The position vector that locates point P with respect to coordinate frame B is given by

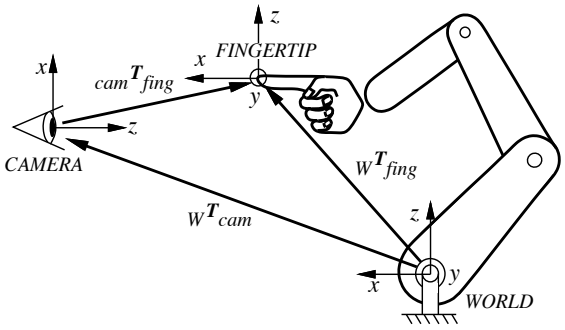
$$ \boldsymbol{r}^T_B = [\ -1,\ 1.5,\ 0,\ 1\ ]. $$

   Solve for the position vector that locates point p with respect to frame A, $\boldsymbol{r}_A$.

   (b) **Hand-Eye Coordination** - Given homogeneous transforms from world to camera coordinates and from world to fingertip coordinates:

$$ {}_W\mathbf{T}_{cam} = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 0 & -1 & 4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}_W\mathbf{T}_{fing} = \begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

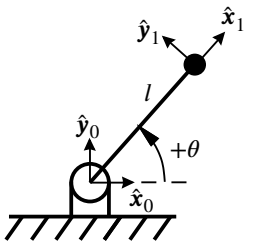


   i. Solve for ${}_{cam}\mathbf{T}_{fing}$

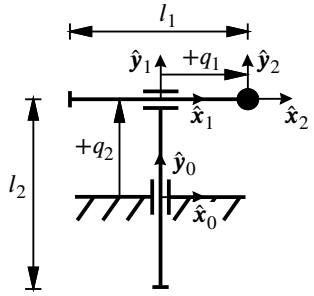
ii. Compute the position of the hand's fingertip in the camera's coordinate frame.

4. **Kinematics** - For each of the simple mechanisms below: compute the: **(i) forward kinematics** - compose the set of homogeneous transforms ${}_{i-1}T_i \;:\; i = 1, N$, through intermediate coordinate frames illustrated and compute the composite ${}_0T_N$ transform; **(ii) reachable workspace** - define the subset of Cartesian space that the endpoint (frame N) can reach; and **(iii) derive the Jacobian** describing how joint velocities transform into Cartesian velocities.

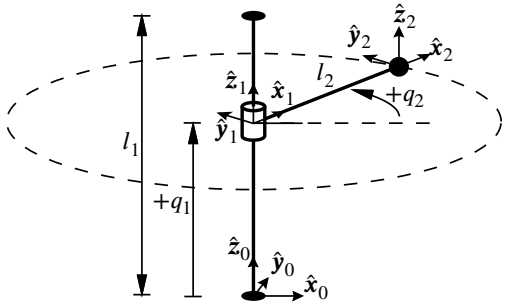
   (a) **1R Mechanism** comprised of a single revolute degree of freedom is illustrated below, where control input $q = \theta$ maps the mechanism to outputs (x, y).



   (b) **Planar Cartesian 2P Mechanism** composed of two consecutive prismatic degrees of freedom (implemented as linear sliding elements) with input degrees of freedom (q_1, q_2) and generates endpoint positions (x, y).



   (c) **PR Mechanism** - a prismatic-revolute sequence of degrees of freedom. The prismatic joint moves through range of motion $0 \leq q_1 \leq l_1$ and the revolute joint can rotate continuously $0 \leq q_2 < 2\pi$. The length of the link from the revolute joint to the endpoint is l_2.

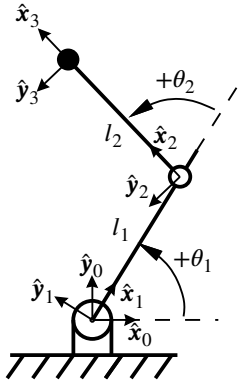


5. **Kinematic Analysis:** - The planar, 2R mechanism is used in several examples and serves as Roger's arm in the Roger-the-Crab simulator. It consists of a pair of adjacent revolute joints that rotate about the world frame $\hat{\mathbf{z}}$ axis.

   (a) Write the homogeneous transforms for ${}_0\mathbf{T}_1,\ {}_1\mathbf{T}_2,\ {}_2\mathbf{T}_3$ and compute the net transform ${}_0\mathbf{T}_3 = {}_0\mathbf{T}_1\,{}_1\mathbf{T}_2\,{}_2\mathbf{T}_3$. Simplify the result.

(a) The Jacobian for the 2 DOF manipulator, is derived in the text:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$



i. Compute the joint angle velocities required to execute an instantaneous velocity of 1 $[m/s]$ in the $\hat{\mathbf{x}}_0$-direction from any initial joint angle configuration, i.e

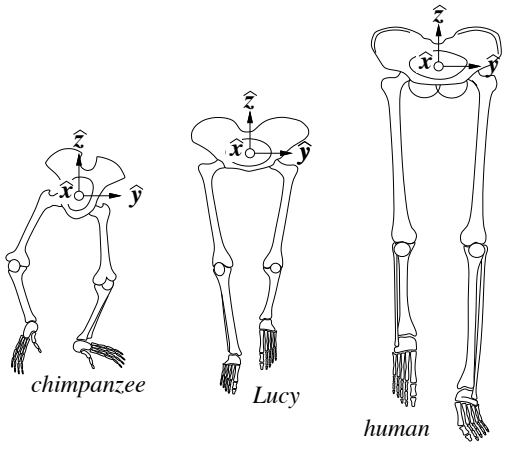
$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

ii. Compute an expression for the torques on the joints of the manipulator necessary to apply 1 $[N]$ endpoint force in the $-\hat{\mathbf{y}}_0$ direction from any reachable initial posture.

(a) Compute the eigenvalues and eigenvectors for $\mathbf{J}\mathbf{J}^T$ (the squared velocity transformation) at $\theta_1 = \pi/4$, $\theta_2 = \pi/2$, assuming $l_1 = l_2 = 1$. Draw the velocity ellipsoid at this configuration.

6. **Forensic Anthropology: Force & Velocity Ellipsoids** -

"Lucy" (Australopithecus afarensis) was discovered in 1978 in Ethiopia by Mary Leakey. Lucy is a rare data point concerning the emergence of contemporary humans in the fossil record. She is classified *Hominid*—an ape closely related to human beings—that lived around 3.75 million years ago. In terms of overall body and brain size, Lucy resembles a chimpanzee. However, her hip joint, knee, and pelvis indicate that she walked upright like a modern human. This suggests that human-like bipedalism may have preceded the development of the contemporary human hand and brain. Subsequently, the discovery of Ardipithecus ramidus ("Ardi"), dating back another 500,000 years before Lucy, supports this general hypothesis.



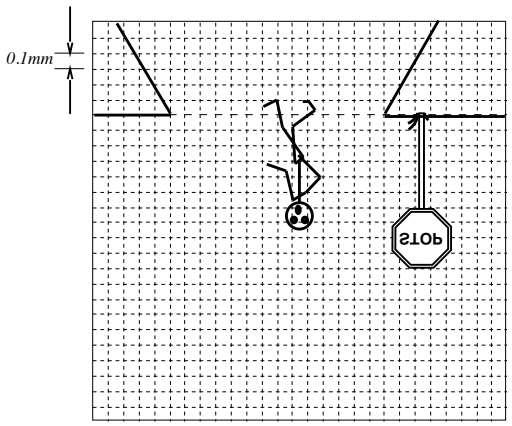
Although the skeleton of Lucy and the chimp are similar, the respective ranges of motion of hip and knee are quite different. The musculature, tendons, and ligaments of the chimp establish a relatively "bent-leg" and "duck-footed" posture that influences the quality of bipedal movements. This is clearly demonstrated in Walking with Lucy (https://youtu.be/xT8Np0gI1dI).

Based of the anatomical differences in the two animals, formulate representative configurations of the planar 2R manipulator to approximate a support leg mid-stride for the chimp and for Lucy in the sagittal ($\hat{\mathbf{x}}$-$\hat{\mathbf{z}}$) plane, plot their respective force and velocity ellipsoids.

Using this result, contrast the kinematic ability of Lucy's hip and knee with that of the chimpanzee and speculate about the kinds of habitat that could select for these skeletal variations.

7. **Pinhole Camera** -

You are speeding through downtown Amherst when the image below appears on your retina. It takes 35 meters to bring your car to a stop from this speed! Stop signs are 2.5 meters high and the optometrist told you that the focal length of your eye, f is exactly 0.015 m. Can you stop in time?



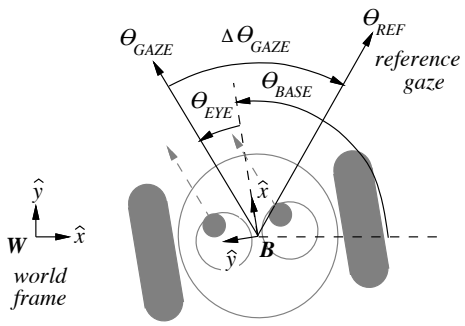
8. **Stereo Question - stereo acuity**
A high-velocity object must be tracked visually to be caught. Where should the agent position itself so that the target is not lost? Write a controller, for the well-placed agent, to track and "catch" the projectile—i.e. so that the trajectory of the agent on the ground intersects the trajectory of the projectile.

9. **Error Propagation** - from the retina to Cartesian space

   Compute the scale factor for stereo triangulation for Roger the crab, i.e. given focal length, field of view, and pixel width => +/- 1/2 pixel, $d\boldsymbol{\phi}$...

10. **Redundant Kinematic Systems**

   (a) **Redundant Gaze Control for Roger-the-Crab** - The figure illustrates the mobile base and stereo visual system of the simulated Roger-the-Crab robot. Consider the task in which the robot controls the one dimensional cycloptic gaze direction in the world frame



$\boldsymbol{r}^T = [\theta_{\text{GAZE}}]$ to coincide with a reference direction θ_{REF} using two input degrees of freedom $\boldsymbol{q}^T = [\theta_{\text{BASE}}\ \theta_{\text{EYE}}]$, where $\theta_{\text{GAZE}} = \theta_{\text{BASE}} + \theta_{\text{EYE}}$ is the governing forward kinematic relation.

The gaze direction can be controlled with any combination of values for the configuration variables that comply with the forward kinematics of the system. The sensitivity of the control to each of these configuration variables is estimated in the Jacobian

$$\mathbf{J} = \begin{bmatrix} \frac{d\theta_{\text{GAZE}}}{d\theta_{\text{BASE}}} & \frac{d\theta_{\text{GAZE}}}{d\theta_{\text{EYE}}} \end{bmatrix} = [\,1\ \ 1\,]$$

which shows that the gaze angle is equally sensitive to both configuration variables. This is a redundant system because the number of inputs $n = 2$ is greater than the number of outputs $m = 1$ and both columns of the Jacobian are non-zero.

$$\begin{aligned} \Delta\boldsymbol{r} &= \mathbf{J}\ \Delta\boldsymbol{q} \\ \Delta\theta_{\text{GAZE}} &= [\,1\ \ 1\,] \begin{bmatrix} \Delta\theta_{\text{BASE}} \\ \Delta\theta_{\text{EYE}} \end{bmatrix} \end{aligned} \qquad \begin{aligned} \Delta\boldsymbol{q} &= \mathbf{J}^{\#}\ \Delta\boldsymbol{r} \\ \begin{bmatrix} \Delta\theta_{\text{BASE}} \\ \Delta\theta_{\text{EYE}} \end{bmatrix} &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} \left([\,1\ \ 1\,] \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right)^{-1} \Delta\theta_{\text{GAZE}} \\ &= \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \Delta\theta_{\text{GAZE}} \end{aligned}$$

The right pseudoinverse distributes the gaze error equally over the two degrees of freedom in order to minimize $\Delta\boldsymbol{q}^T\Delta\boldsymbol{q}$. Other concerns can influence the distribution of displacements over the controllable

degrees of freedom. For example, energy expenditure and response time depend on how gaze errors are distributed to controllers that actuate the relatively massive body and the lightweight and responsive eyes. Weighted least squares solutions exist [148] that can be used to incorporate energetic and other performance issues directly into the choice of inverse solution.

(b) **Motion-Stabilized Hand Control**

11. **Invent Your Own Homework** - Make a homework problem out of your favorite content in Chapter 4. Write a question and a solution for it from the material in the reading. The problem should be different than the questions already here. Ideally, they should call for a short discussion and quantative analysis—it should not take more than 30 minutes to solve open book.