Lecture Notes (CS223A)

Introduction to Robotics

Oussama Khatib and Krasimir Kolarov

Stanford University

Winter 2000

Contents

Chapter 1

Spatial Description

1.1 Manipulator Structure

A manipulator is a mechanical structure consisting of rigid bodies, called *links*, connected together through *joints*. The part of the manipulator that interacts the most with the surrounding environment is the last body in the chain of the manipulator's structure, called the *end-effector*. Based on the task the manipulator is performing, the end-effector can be a variety of devices: a mechanical gripper, a tool, a vacuum-operated positioning device. The first part of the manipulator is typically fixed in the environment and is called the *base*.

The configuration of a manipulator is determined by the positions of its joints. To a given configuration of the manipulator correspond a unique configuration of end effector, which can be described by the the effector position and orientation. The model describing the relationships between the manipulator configuration and the end-effector configuration is called the *forward kinematics* of the manipulator. In this chapter, we will focus on the development of the forward kinematics of a manipulator.

Here we assume that each joint can only perform one degree-of-freedom (1 DOF) motion. We will distinguish between two types of joints: *prismatic* joints which provide linear motion between links and *revolute*

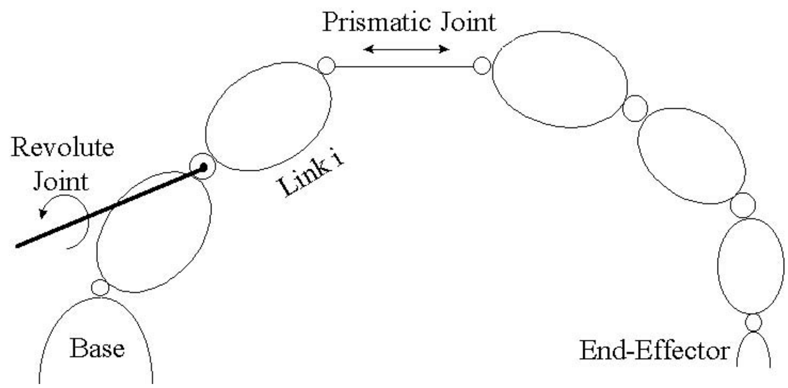

Figure 1.1: General Manipulator

joints which provide rotational motion between links, as illustrated in Figure 1.1. We denote by n the number of joints that the robot has.

Our first goal is to describe the configuration of the manipulator. In general, a manipulator has n joints and $n+1$ links (including the base and the end-effector). We want to describe the configuration of all these connected rigid bodies. To describe the configuration of the system, we need to select a set of parameters that allow to determine this configuration. Such parameters are called *configuration parameters*. There are many different possibilities for selecting these parameters. One approach is to attach a frame to the base of the manipulator and then to locate all the moving links by vectors with respect to that fixed frame, as illustrated in Figure 1.2. Since the configuration of a rigid body can be described by 3 vectors locating 3 different points of the body, this would result in 9-parameter representation for each of the links. Obviously, this is not an efficient approach for the description of the configuration of the manipulator, as it would require $9 \times n$ parameters for describing the configuration of the n moving link of this manipulator. In the next section, we will introduce a particular set of parameters, called generalized coordinates, which provides a minimal representation of the manipulator configuration.

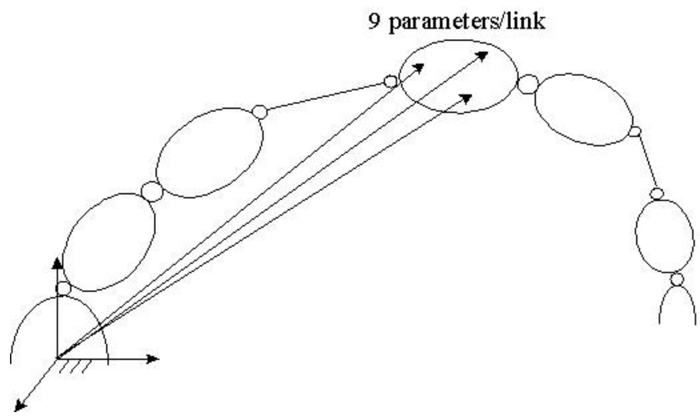

Figure 1.2: Generalized Coordinates

1.1.1 Generalized Coordinates

A set of generalized coordinates for a manipulator is a set of independent configuration parameters. The number of generalized coordinates defines the number of degrees of freedom of the system. In other words the number of DOF of the system is determined by the number of generalized coordinates needed to describe the system. For a system having $n + 1$ links and n joints we want to determine how many degrees of freedom the system has, or alternatively how many generalized coordinates will be needed to describe it.

We will start by disassembling the manipulator. This generates an $n+1$ rigid bodies, of which 1 is fixed to the ground and n are completely free to move. The task now is to describe their configuration in space.

The configuration of a rigid body in space can be described by six parameters: three parameters for the position of a point on the rigid body and three other parameters for describing its orientation. Full description of the configuration of this system with n free moving bodies requires $6 \times n$ parameters.

Rigid bodies can move with respect to each other. However in the manipulator these motions are constrained because the bodies are connected. The connections through joints will introduce constraints on the motion of the rigid bodies. Let us now assemble the manipulator by connecting the rigid bodies with the joints. Each joint has 1 DOF allowing a single motion – revolute or prismatic. Thus each joint will introduce 5 constraints on the motion of the free rigid bodies. We have n joints, thus in total the connections will introduce $5 \times n$ constraints on the motion of the system. The number of generalized parameters that are needed to describe the configuration of the system will be the difference between the number of parameters for all free rigid bodies and the number of constraints introduced by the joints,

$$6 \times n - 5 \times n = n.$$

This is also the number of DOF of the system. Thus a manipulator that has n joints, will have n DOF and will require n generalized coordinates to describe its configuration.

As mentioned above, the joints that we will consider are either prismatic or revolute. A prismatic joint $\{i\}$ results in a linear (translational) motion that is measured as a displacement d_i between the two neighboring links. If joint $\{i\}$ was revolute, it would result in a rotational motion that would be measured with an angle θ_i between the corresponding links. To unify the description of the different coordinates associated with linear motion and rotational motion, we introduce a common coordinate q_i that will denote both types. The type of a joint $\{i\}$ is determined by a binary parameter ϵ_i defined as

$$\epsilon_i = \begin{cases} 0 & \text{for a revolute joint } \theta_i; \\ 1 & \text{for a prismatic joint } d_i. \end{cases} \tag{1.1}$$

The i-th joint can be then described by the coordinate q_i, defined as

$$q_i = \overline{\epsilon_i}\theta_i + \epsilon_i d_i \tag{1.2}$$

where

$$\overline{\epsilon_i} = 1 - \epsilon_i$$

The coordinates $q_1, q_2, \ldots, q_n$ provides a minimal set of parameters for describing the manipulator configuration. This is the set of generalized joint coordinates of the manipulator.

In the next section we will introduce representations for the configuration of the end effector of the manipulator.

1.1.2 Joint and Operational Coordinates

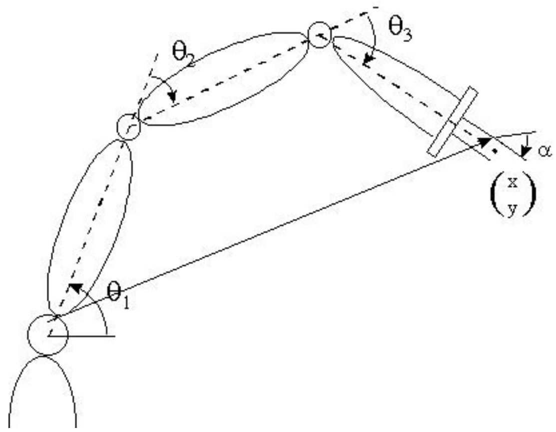

Figure 1.3: 3DOF Revolute Mechanism

Consider the mechanism shown in Figure 1.3. This is a planar mechanism where the links of the manipulator are moving in the plane of the paper and the axes of the joints are perpendicular to that plane. There are three links with three revolute joints described by the parameters θ_1, θ_2 and θ_3 defined as the angles between the axes of the consecutive links (including the ground described by axis X). Once we have defined these 3 parameters we have a unique description of the configuration of the system. The 3 parameters are the three joint coordinates of the system. The space defined by θ_1, θ_2 and θ_3 is called the *joint space* of the manipulator. A point in that space defines a configuration of the mechanism. Inversely every configuration of the mechanism is represented by a point in the joint space.

We can also describe a space that defines the configuration of the end-effector (gripper) of the manipulator, i.e. its position and orientation. For example the position of the gripper can be described by the position of the point at the end-effector in the plane (x, y). The orientation of the gripper can be described by the angle α between the X axis of the fixed base and the vector connecting the base point O and the end-effector point. The end-effector configuration in the example above is defined by the 3 coordinates (x, y, α).

For the general case of spatial manipulators, we will need three Cartesian coordinates, x, y, and z to represent the position of the end-effector, and a three other coordinates, e.g. three angles α, β, and γ, to represent its orientation.

The end-effector position and orientation coordinates provide a description of the manipulator task or operation and are called *task coordinates,* or *operational coordinates*[1].

While the joint coordinates describe the configuration of the entire manipulator, the task or operational coordinates describe the configuration of the end-effector. The next objective is to establish the relationships between these coordinates.

1.1.3 Position and Orientation of Rigid Bodies

We will first introduce the various notions that are needed to develop the relationships between the manipulator and effector coordinates. We start with the basic definition of the position of a point in space.

A point in space can be described as a vector locating this point with respect to some origin or reference point, $\mathcal{O}$. Once the point $\mathcal{O}$ is fixed, a point $\mathcal{P}$ in space can be described by the vector $\mathbf{p} = \vec{\mathcal{O}\mathcal{P}}$, representing the position of this point with respect to $\mathcal{O}$, as illustrated in Figure 1.4.

We distinguish between the vector $\mathbf{p}$ and its components. The vector $\mathbf{p}$ represents the position of point $\mathcal{P}$ with respect to origin $\mathcal{O}$. Its components are determined by the frame with respect to which this vector is evaluated. These components vary from frame to frame. Consider the frame denoted by $\{A\}$ in Figure 1.5, with unit vectors $\hat{X}_A$, $\hat{Y}_A$, and $\hat{Z}_A$. The $\hat{.}$ denotes that these are unit vectors. A point $\mathcal{P}$ described by $\mathbf{p}$ will have in frame $\{A\}$ the components $(p_{X_A}, p_{Y_A}, p_{Z_A})$. If we describe it in a different frame we will have a different set of coordinates. The components of $\mathbf{p}$ in frame $\{A\}$ will be denoted by the column matrix

[1]by extension to the position, these coordinates are sometimes called Cartesian coordinates.

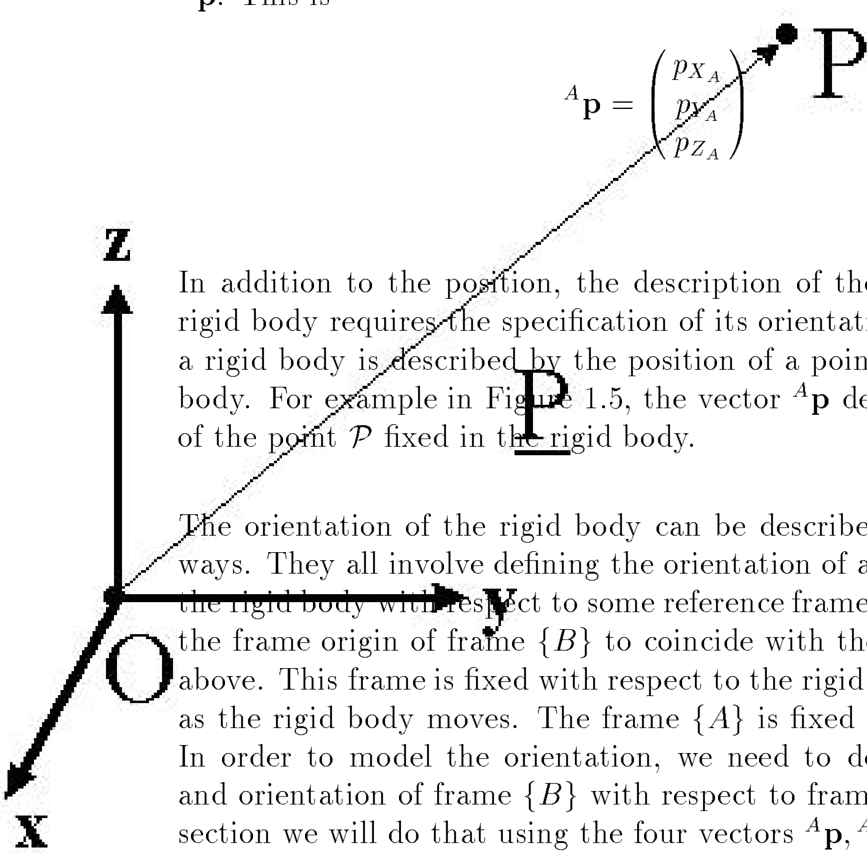

Figure 1.4: Position of a point

${}^{A}\mathbf{p}$. This is

$$
{}^{A}\mathbf{p} = \begin{pmatrix} p_{X_A} \\ p_{Y_A} \\ p_{Z_A} \end{pmatrix} \tag{1.3}
$$

In addition to the position, the description of the configuration of a rigid body requires the specification of its orientation. The position of a rigid body is described by the position of a point fixed in that rigid body. For example in Figure 1.5, the vector ${}^{A}\mathbf{p}$ describes the position of the point $\mathcal{P}$ fixed in the rigid body.

The orientation of the rigid body can be described in many different ways. They all involve defining the orientation of a frame $\{B\}$ fixed in the rigid body with respect to some reference frame $\{A\}$. We will select the frame origin of frame $\{B\}$ to coincide with the point $\mathcal{P}$ described above. This frame is fixed with respect to the rigid body and will move as the rigid body moves. The frame $\{A\}$ is fixed in the environment. In order to model the orientation, we need to describe the position and orientation of frame $\{B\}$ with respect to frame $\{A\}$. In the next section we will do that using the four vectors ${}^{A}\mathbf{p}, {}^{A}X_B, {}^{A}Y_B, {}^{A}Z_B$.

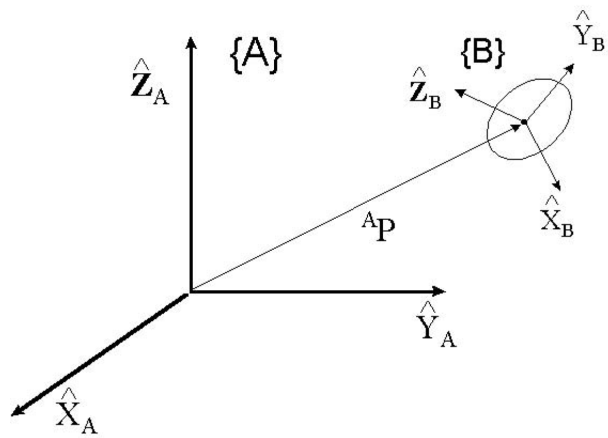

Figure 1.5: Rigid Body Configuration

1.1.4 Rotation Matrix

The relationship between different descriptions of a vector relies on the notion of rotation matrices. A rotation matrix ${}^A_BR$ describes the orientation of frame $\{B\}$ with respect to frame $\{A\}$. In this notation, the leading subscript B and superscript A point to the two frames involved in this transformation.

The columns of the rotation matrix are simply the three unit vectors of frame $\{B\}$ expressed in frame $\{A\}$. If r_{ij} are the elements of this matrix, the rotation matrix is given by

$$ {}^A_BR = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = [\, {}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B \,] \tag{1.4} $$

The components of vector $\hat{X}_B$ in $\{A\}$ are given by the dot product of $\hat{X}_B$ with the vectors $\hat{X}_A$, $\hat{Y}_A$, and $\hat{Z}_A$.

$$ {}^A\hat{X}_B = \begin{bmatrix} \hat{X}_B.\hat{X}_A \\ \hat{X}_B.\hat{Y}_A \\ \hat{X}_B.\hat{Z}_A \end{bmatrix} \tag{1.5} $$

Similarly we can compute ${}^A\hat{Y}_B$ and ${}^A\hat{Z}_B$ and write the rotation matrix in the following form

$$
{}^A_BR = \begin{bmatrix} \hat{X}_B.\hat{X}_A & \hat{Y}_B.\hat{X}_A & \hat{Z}_B.\hat{X}_A \\ \hat{X}_B.\hat{Y}_A & \hat{Y}_B.\hat{Y}_A & \hat{Z}_B.\hat{Y}_A \\ \hat{X}_B.\hat{Z}_A & \hat{Y}_B.\hat{Z}_A & \hat{Z}_B.\hat{Z}_A \end{bmatrix} \tag{1.6}
$$

Note that the dot products above are not expressed in any particular frame – the dot product computation can be performed in any frame with respect to which both vectors are expressed.

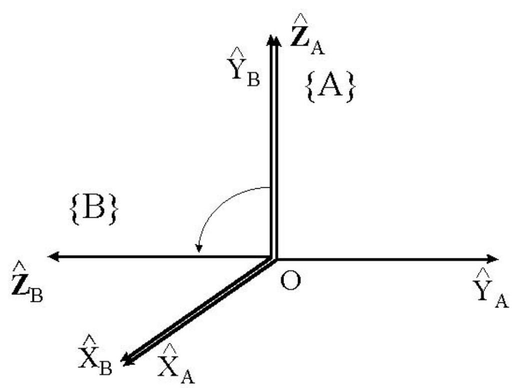

Figure 1.6: Simple Rotation of a frame

An important property of this matrix is that the *rows* of this matrix are the components of the three unit vectors of frame $\{A\}$ expressed with respect to frame $\{B\}$

$$
{}^A_BR = [\,{}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B\,] = \begin{bmatrix} {}^B\hat{X}_A^T \\ {}^B\hat{Y}_A^T \\ {}^B\hat{Z}_A^T \end{bmatrix} = {}^B_AR^T \tag{1.7}
$$

This property is illustrated in the example of Figure 1.6, where frame $\{B\}$ is obtained from frame $\{A\}$ by a 90 degrees rotation about the $\hat{X}$ axis

$$
{}^{A}_{B}R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{matrix} \leftarrow {}^{B}\hat{X}_A^T \\ \leftarrow {}^{B}\hat{Y}_A^T \\ \leftarrow {}^{B}\hat{Z}_A^T \end{matrix} \tag{1.8}
$$

As can be seen, the rotation matrix from frame $\{B\}$ to frame $\{A\}$ is equal to the transpose of the rotation matrix from frame $\{A\}$ to frame $\{B\}$

$$
{}^{A}_{B}R = {}^{B}_{A}R^T
$$

Since the inverse of the rotation matrix describing $\{B\}$ with respect to $\{A\}$ is the matrix describing $\{A\}$ with respect to $\{B\}$, we obtain

$$
{}^{A}_{B}R^{-1} = {}^{B}_{A}R = {}^{A}_{B}R^T
$$

Thus the inverse of a rotation matrix is equal to its transpose

$$
{}^{A}_{B}R^{-1} = {}^{A}_{B}R^T
$$

This is a general property of orthonormal matrices (a matrix with orthogonal unit columns and rows).

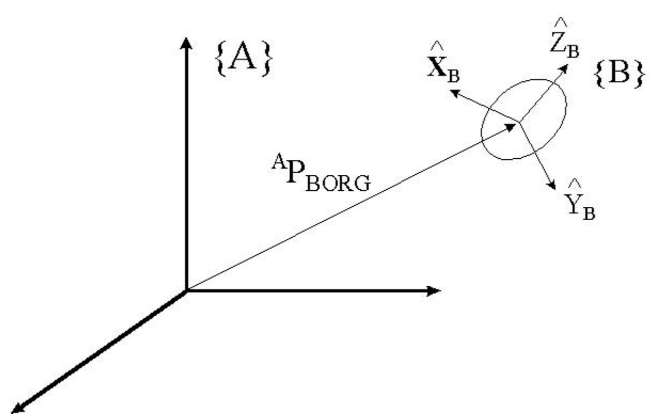

Figure 1.7: Frame description

The quantities that we defined so far allow us to describe frames. A frame describes the position and orientation of a rigid body. It is defined by the three unit vectors ${}^A\hat{X}_B$, ${}^A\hat{Y}_B$ and ${}^A\hat{Z}_B$ and the vector that locates the origin of the frame. The notation we will use for this vector is ${}^A\mathbf{p}_{Borg}$ - the origin of frame $\{B\}$ expressed in frame $\{A\}$. Thus a frame is the set of four vectors ${}^A\hat{X}_B, {}^A\hat{Y}_B, {}^A\hat{Z}_B, {}^A\mathbf{p}_{Borg}$ (see Figure 1.7). A frame $\{B\}$ will be represented by $\{B\} = \{ {}^A_BR \quad {}^A\mathbf{p}_{Borg} \}$.

1.2 Transformations

In order to describe the end-effector configuration with respect to the base of a manipulator, we need to establish the relationships between descriptions in different frames attached to different links along the manipulator's structure.

1.2.1 Pure Rotation Transformation

For example in Figure 1.8, a point $\mathcal{P}$ is defined by the vector connecting it to the origin of frame $\{A\}$. The components in $\{A\}$ of this vector will be different from those computed with respect to frame $\{B\}$, having the same origin.

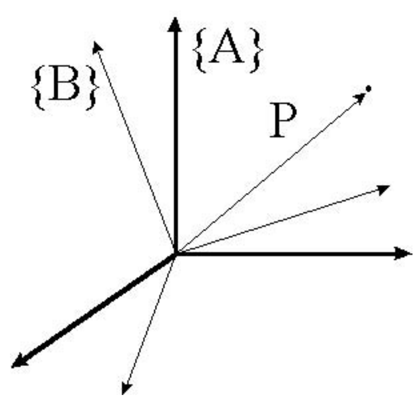

Figure 1.8: Mapping between frames

The coordinates of the vector $\mathbf{p}$ in $\{A\}$ are the dot products of vector $\mathbf{p}$ with the unit vectors X_A, Y_A, and Z_A. The dot product computation can be done in any frame, in particular in frame $\{B\}$

$$ {}^{A}\mathbf{p} = \begin{pmatrix} {}^{B}\hat{X}_A . {}^{B}\mathbf{p} \\ {}^{B}\hat{Y}_A . {}^{B}\mathbf{p} \\ {}^{B}\hat{Z}_A . {}^{B}\mathbf{p} \end{pmatrix} = \begin{pmatrix} {}^{B}\hat{X}_A^T \\ {}^{B}\hat{Y}_A^T \\ {}^{B}\hat{Z}_A^T \end{pmatrix} . {}^{B}\mathbf{p} \tag{1.9} $$

Using the definition of the rotation matrix, yields

$$ {}^{A}\mathbf{p} = {}^{A}_{B}R \cdot {}^{B}\mathbf{p} \tag{1.10} $$

This establishes the relationship between the description of a vector $\mathbf{p}$ expressed with respect to a frame $\{B\}$ to its description with respect to a frame $\{A\}$, having the same origin.

Notice the arrangement in this notation of the leading sub- and superscripts. The leading superscript of $\mathbf{p}$ matches the subscript of the rotation matrix. This arrangement is very effective when dealing with more complex chain multiplication.

1.2.2 Pure Translation Transformation

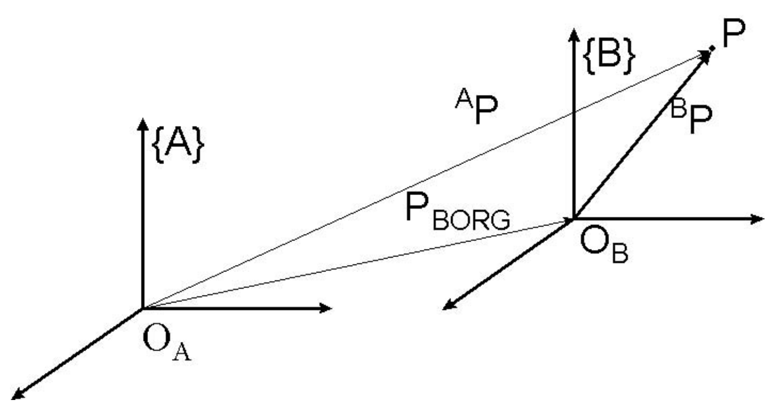

Figure 1.9: Translation of frames

We have so far described relationships for transformations involving pure rotation of frames. Figure 1.9 shows an example where a frame $\{B\}$ is translated with respect to frame $\{A\}$ without rotation. This motion can be described by the vector denoting the position of the origin of frame $\{B\}$ with respect to the origin of frame $\{A\}$, $\mathbf{p}_{Borg/O_A}$.

Let us consider an arbitrary point $\mathcal{P}$ in the space. This point can be described with respect to frame $\{A\}$ (vector $\mathbf{p}_{O_A}$) or frame $\{B\}$ (vector $\mathbf{p}_{O_B}$). This situation is different than before, because now we have two different vectors describing the position of the same point. In the rotation case, we had the same vector described in two different frames. A translation operation is a mapping that is transforming a vector describing some point with respect to some origin point to a vector describing that same point with respect to another origin point. The difference between the two vectors is exactly the vector describing the position of the origin of the frame $\{B\}$ with respect to the origin of frame $\{A\}$, $\mathbf{p}_{Borg/O_A}$, and thus

$$\mathbf{p}_{O_A} = \mathbf{p}_{O_B} + \mathbf{p}_{Borg/O_A} \tag{1.11}$$

As before this vector relationship can be expressed in any frame (in particular with respect to frame $\{A\}$).

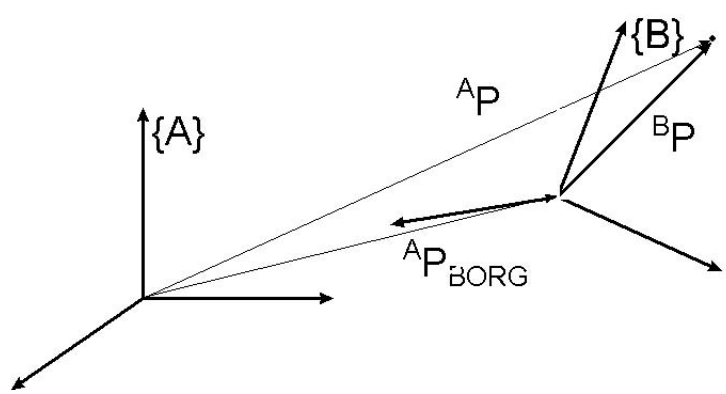

Figure 1.10: Homogeneous Transform

1.2.3 General Transformation

A general transformation would involve both a rotation of a frame and a translation of the rotated frame. We can implement the general transformation by first performing a rotation to make the axes of the two frames parallel, and then translating them. The resulting relationship is

$$^{A}\mathbf{p}_{O_A} = {}^{A}_{B}R\,{}^{B}\mathbf{p}_{O_B} + {}^{A}\mathbf{p}_{Borg/O_A} \tag{1.12}$$

In the next section we will rewrite this equation in a matrix form that can be very useful for uniform fast numerical computations.

1.2.4 Homogeneous Transformation

The general transformation in equation 1.12 can be written in a compact form that is more suitable for compound transformations and propagation of description between links. This is called the *homogeneous transformation*, which is obtained by augmenting the relationship of equation 1.12 by one dimension. We will form a 4×4 matrix, where the primary blocks are the 3×3 rotation matrix and the position vector ${}^{A}\mathbf{p}_{Borg}/O_A$. The last row of the 4×4 matrix is $[0 \quad 0 \quad 0 \quad 1]$. The vectors ${}^{A}\mathbf{p}$ and ${}^{B}\mathbf{p}$ are also augmented by 1 to make them 4-dimensional vectors. The resulting equation is

$$\begin{bmatrix} {}^{A}\mathbf{p}_{O_A} \\ 1 \end{bmatrix} = \begin{bmatrix} & {}^{A}_{B}R & & {}^{A}\mathbf{p}_{Borg/O_A} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\mathbf{p}_{O_B} \\ 1 \end{bmatrix} \tag{1.13}$$

The homogeneous transformation matrix matrix will be denoted by ${}^{A}_{B}T$ and the above relationship can be written as

$$^{A}\mathbf{p}_{O_A} = {}^{A}_{B}T_{(4\times 4)}\,{}^{B}\mathbf{p}_{O_B} \tag{1.14}$$

The homogeneous transform combines both the rotation of $\{B\}$ to $\{A\}$ and the translation of the origin of $\{B\}$ with respect to $\{A\}$. This transform represents one of the basic tools for the kinematics of mechanisms.

Example We will illustrate this concept with the example depicted in Figure 1.11. The origin of frame $\{B\}$ in the figure is translated to a position $[0 \quad 3 \quad 1]$ with respect to frame $\{A\}$. We would like to find the homogeneous transformation between the two frames in the figure. In particular for a point $\mathcal{P}$ defined as $[0 \quad 1 \quad 1]$ in frame $\{B\}$, we would like to find the vector describing this point with respect to frame $\{A\}$.

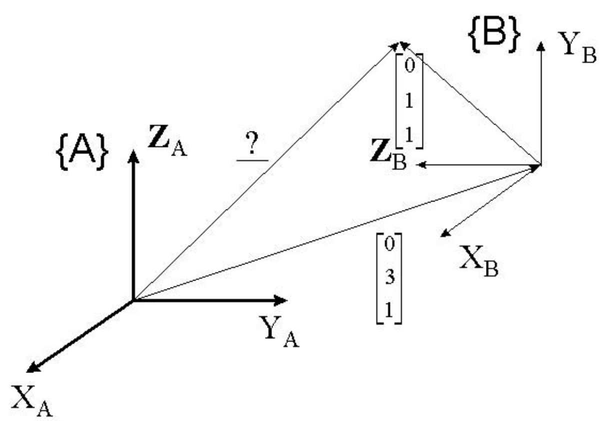

Figure 1.11: Example of Homogeneous Transform

The matrix ${}^A_BT$ is formed as defined earlier using the rotation matrix and the given translation vector. The result is

$$
{}^A_BT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.15}
$$

Since $\mathbf{p}$ in frame $\{B\}$ is

$$
{}^B\mathbf{p} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{1.16}
$$

we can compute the vector $\mathbf{p}$ in frame $\{A\}$ using the relationship ${}^{A}\mathbf{p} = {}^{A}_{B}T^{B}\mathbf{p}$,

$$
{}^{A}\mathbf{p} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} \tag{1.17}
$$

1.2.5 Transforms as Operators

The transforms presented above allow us to change the descriptions of points in space from one frame to another. However, these transforms can be also viewed as operators acting on points and changing their locations in the space. In that case, rotations and translations will be described as **operators** moving one point into another point, with respect to the same frame.

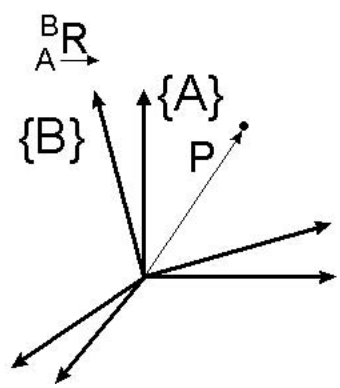

Figure 1.12: Rotation: Changing Description

Let us first consider the rotation matrix. As illustrated in Figure 1.12, the rotation matrix allows to change the description of a point $\mathcal{P}$ from one frame $\{A\}$ to another frame $\{B\}$. This is

$$
{}^{A}\mathbf{p} = {}^{A}_{B}R^{B}\mathbf{p} \tag{1.18}
$$

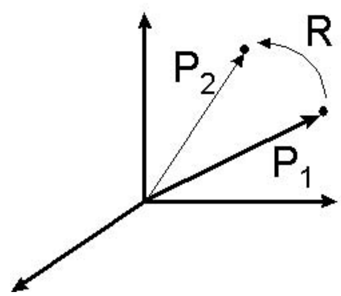

Figure 1.13: Rotational Operator

In Figure 1.13, the rotation matrix is acting on a vector locating a point $\mathcal{P}_1$ and transforming it into a different vector locating a different point $\mathcal{P}_2$ through a rotation the X axis. The rotation matrix in this case is treated as a *rotation operator* about the x axis. In the general case, this operator can act about an arbitrary vector $\mathbf{k}$ with an angle θ. Applying the rotation operator to vector $\mathbf{p}_1$ produces another vector $\mathbf{p}_2$. The corresponding relationship is:

$$\mathbf{p}_2 = \ R_k(\theta)\ \mathbf{p}_1 \tag{1.19}$$

A rotation about the X axis, for example, is given by

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{1.20}$$

If $\mathbf{p}_1$ was given as $[\,0 \quad 2 \quad 1\,]$ and the rotation was of an angle 30^o the resulting vector would be :

$$\mathbf{p}_2 = R_X(\theta)\mathbf{p}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.8 & -0.6 \\ 0 & 0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} = \mathbf{p}_2 = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \tag{1.21}$$

Next we will consider the translation as an operator. With our previous interpretation of translation, a point P_2 (see Figure 1.14) is described

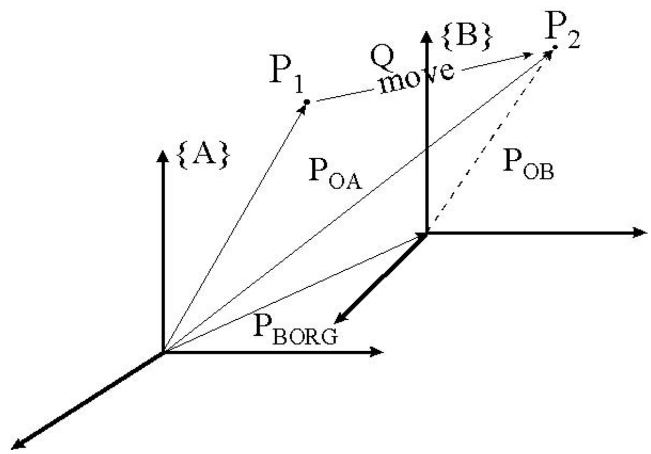

Figure 1.14: Translation Operator

as a vector $\mathbf{p}_{OA}$ with respect to the origin of frame $\{A\}$, or $\mathbf{p}_{OB}$ with respect to the origin of frame $\{B\}$. Here we are describing the same point with two different vectors and the relationship is:

$$\mathbf{p}_{OA} = \mathbf{p}_{OB} + \mathbf{p}_{Borg} \tag{1.22}$$

When viewed as an operator, a translation Q ($Q : P_1 \rightarrow P_2$) moves a point P_1 (described by vector $\mathbf{p}_1$) into a point P_2 (described by vector $\mathbf{p}_2$)

$$\mathbf{p}_2 = \mathbf{p}_1 + Q$$

In this case we have a description of two different points with two different vectors. These vector equations can be expressed in any frame. In a matrix form, the homogeneous transform is

$$\mathbf{p}_2 = T_Q \mathbf{p}_1$$

where

$$T_Q = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.23}$$

(q_x, q_y, q_z) are the components of the vector Q. Note that these are only variables in the representation of the translational operator.

In the general case, the rotational operator and translational operator are combined in a general transformation operator, T, defined below similarly to the homogeneous transform:

$$\mathbf{p}_2 = \begin{pmatrix} R_K(\theta) & & & Q \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{p}_1 \tag{1.24}$$

The operator T acts on $\mathbf{p}_1$ to produce $\mathbf{p}_2$,

$$\mathbf{p}_2 = T \ \mathbf{p}_1$$

This representation constitutes the third interpretation of the homogeneous transformation. The first one is a description of a frame. The second is a transform mapping that changes the description of a point in space. The third is a transform operator that moves points in space.

In the next section, we will define the inverse of a transformation.

1.2.6 Inverse Transforms

To compute the inverse of a generalized transform, we will consider inverses of the rotation and translation transforms. We already showed that the inverse of a rotation is simply given by the transpose of that rotation, $R^{-1} = R^T$.

For a pure translation, the inverse is given by the same vector with an opposite sign. As illustrated in Figure 1.15, $\mathbf{p}_{Borg/O_A} = -\mathbf{p}_{Aorg/O_B}$.

The presence of both translation and rotation in the homogeneous transform slightly complicates the inverse problem. In particular the

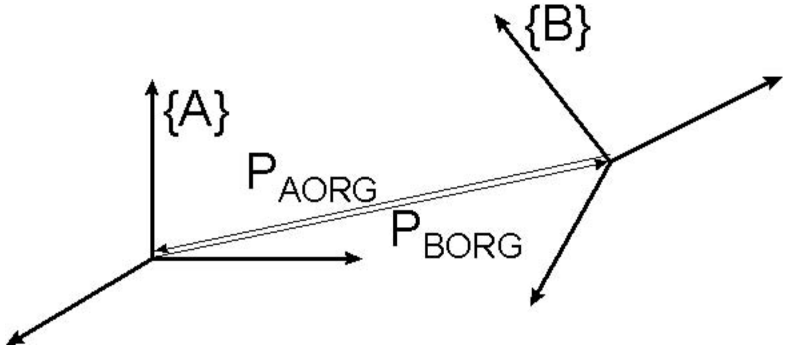

Figure 1.15: Inverse Transform

inverse of the transform is not equal to its transpose because this 4×4 matrix is not orthonormal ($T^{-1} \neq T^T$).

The inverse of a homogeneous transformation matrix

$$ {}^A_BT = \begin{bmatrix} & {}^A_BR & & {}^A\mathbf{p}_{Borg/O_A} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.25} $$

is the matrix

$$ {}^A_BT^{-1} = {}^B_AT = \begin{bmatrix} & {}^A_BR^T & & -{}^A_BR^T.{}^A\mathbf{p}_{Borg/O_A} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} & {}^B_AR & & {}^B\mathbf{p}_{Aorg/O_B} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.26} $$

In the above equation. The rotation part comes directly from the inverse of the rotation matrix, while the translation part represents the vector defining the origin of frame $\{A\}$ expressed with respect to the origin of frame $\{B\}$. The minus sign comes from the inverse of the translation, while the pre-multiplication by ${}^A_BR^T$ is needed to express the vector in the same frame $\{B\}$.

We are now ready to propagate the transform descriptions along the kinematic chain.

1.2.7 Transform Multiplication

We described the transformation from one frame to another as well as its inverse. Next we will combine those transformations so that we can move from some far away frame (attached to the end-effector of the manipulator) to the base frame (attached to the fixed ground). Suppose we have three frames $\{A\}$, $\{B\}$ and $\{C\}$ and we know the transformations from $\{A\}$ to $\{B\}$ and from $\{B\}$ to $\{C\}$. What is the new compound transformation that takes us from frame $\{A\}$ to frame $\{C\}$?

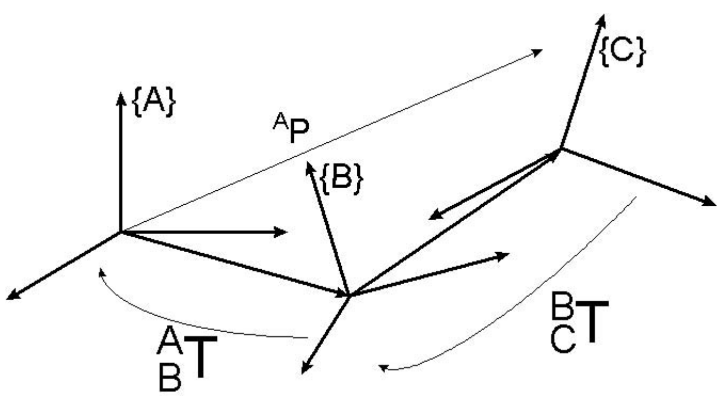

Figure 1.16: Compound Transformations

The result is very simple - it is just the multiplication of the two transformation. We first express:

$$^{B}\mathbf{p} = {}^{B}_{C}T\,{}^{C}\mathbf{p} \tag{1.27}$$

and

$$^{A}\mathbf{p} = {}^{A}_{B}T\,{}^{B}\mathbf{p} = {}^{A}_{B}T\,{}^{B}_{C}T\,{}^{C}\mathbf{p} = {}^{A}_{C}T\,{}^{C}\mathbf{p} \tag{1.28}$$

Thus

$$ {}^{A}_{C}T = {}^{A}_{B}T\,{}^{B}_{C}T \tag{1.29} $$

In other words the 4×4 matrix representing the transformation from $\{A\}$ to $\{C\}$ is ${}^{A}_{C}T$, which is the product of the 4×4 matrices representing the two given transformation ${}^{A}_{B}T$ and ${}^{B}_{C}T$. Since matrix multiplication is not commutative, one has to pay special attention to the order in which the matrices are multiplied. The matrix ${}^{A}_{C}T$ can be written explicitly as:

$$ {}^{A}_{C}T = \begin{bmatrix} {}^{A}_{B}R\,{}^{B}_{C}R & {}^{A}_{B}R\,{}^{B}\mathbf{p}_{Corg} + {}^{A}\mathbf{p}_{Borg} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{1.30} $$

1.2.8 The Transform Equation

As we described earlier, a manipulator consists of many links starting from the fixed base and continuing through to the end-effector. There are frames associated with all those links and we need to propagate the parameters describing the frames in order to build a transformation representing the end-effector frame $\{E\}$ with respect to the fixed base frame $\{B\}$. The manipulators usually work in a real world environment with movable objects (e.g. a machine part that needs to be picked up) placed on top of fixed objects (e.g. a work table). There is a frame describing the work table, as well as a frame associated with the movable machine part. Typically we know the transformation between the manipulator base frame and the work table frame as well as between those of the work table and the machine part. The goal is to calculate the transformation between the frame associated with the machine part that needs to be picked up by the manipulator and the end-effector's frame.

To do that we can use a simple relationship called the transform equation. The idea is that if we start along the chain of bodies involved and we move in the same direction, we will get back to the point from which we started. If we multiply all the transformations involved we should obtain the identity transformation. In the notations of Figure 1.16:

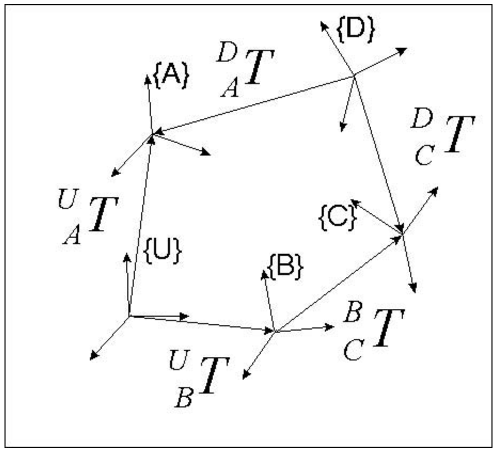

Figure 1.17: Transform Equation

$$ {}^A_U T \, {}^U_B T \, {}^B_C T \, {}^C_D T \, {}^D_A T \equiv I \tag{1.31} $$

From this equation we can extract what is unknown. If for example we need ${}^U_A T$ we can solve the matrix equation and find ${}^U_A T = {}^U_B T \, {}^B_C T \, {}^C_D T \, {}^D_A T$. In terms of the transforms given in the figure ${}^U_A T = {}^U_B T \, {}^B_C T \, {}^D_C T^{-1} \, {}^D_A T$.

In the developments above, we have not explicitly specified the type of parameters that we will would like to use for the description of the position and orientation. The discussion in the next section is concerned with some of these representations.

1.3 Configuration Representations

The end-effector position and orientation is completely defined by the (4×4) homogeneous transformation ${}^B_E T$. However, this representation involves a large number of redundant parameters and a more compact representation of the end-effector configuration is needed. The position and the orientation of the end-effector (depicted in Figure 1.18), with respect to the base can be represented as

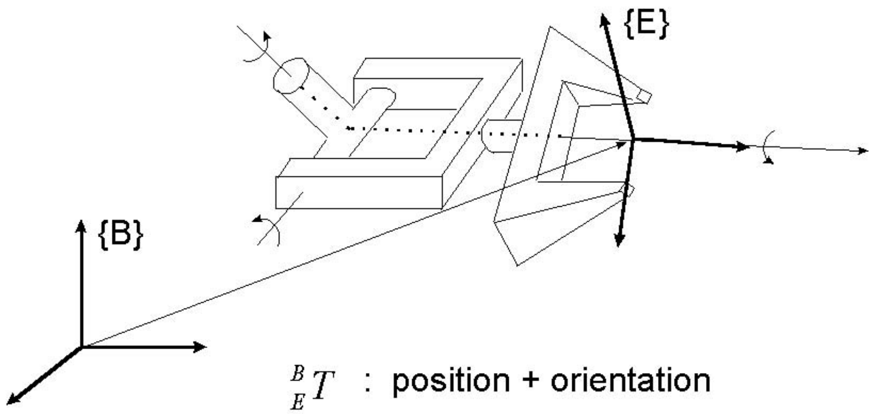

Figure 1.18: End-Effector Configuration

$$X = \begin{bmatrix} X_P \\ X_R \end{bmatrix} \begin{matrix} position \\ orientation \end{matrix} \tag{1.32}$$

There is a variety of sets of 3 parameters for position representations that can be used based on the task that the manipulator is performing. We can use Cartesian coordinates of the end-effector (x, y, z), Cylindrical coordinates (ρ, θ, z) or Spherical coordinators (ρ, θ, ϕ) as depicted in Figure 1.19. Going from one representation to another is quite easy - there are well known explicit formulas for that purpose. The real problem comes when we start considering orientation representation. There is no universal agreement in the field of robotics as to what is the best orientation representation and each representation has advantages and shortcomings.

1.3.1 Direction Cosines

We introduced previously one orientation representation - the rotation matrix. Using frames attached to the end-effector and the base, we can describe the orientation of the end-effector by the compound rotation

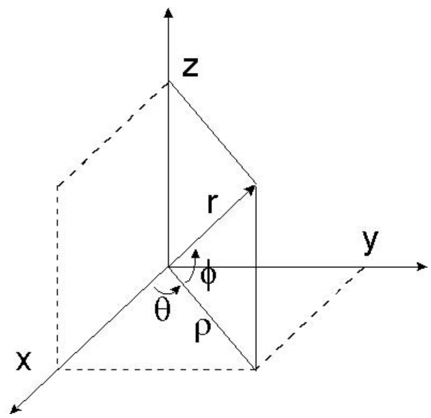

Figure 1.19: Describing the position of a point

matrix developed in the previous sections. The *direction cosines* representation is simply the set of nine parameters involved in the rotation matrix. This matrix can be written as:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = [\, r_1 \quad r_2 \quad r_3 \,] \tag{1.33}$$

The columns of the rotation matrix can be concatenated and arranged in a 9×1 vector:

$$\mathbf{x}_r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}_{(9\times 1)} \tag{1.34}$$

This is the direction cosines representation of the orientation. There are 6 constraints amongst these 9 parameters - 3 because the columns of the matrix are unit vectors and 3 because these vectors are perpendicular.

$$|r_1| = |r_2| = |r_3| = 1 \tag{1.35}$$

and

$$r_1.r_2 = r_1.r_3 = r_2.r_3 = 0 \tag{1.36}$$

Thus of these 9 parameters only 3 are independent. Clearly for most cases it is better to find a more compact representation.

1.3.2 Euler and Fixed Angle Representation

A very common selection of orientation parameters is the three-angle representations. There are many different choices for these angles.

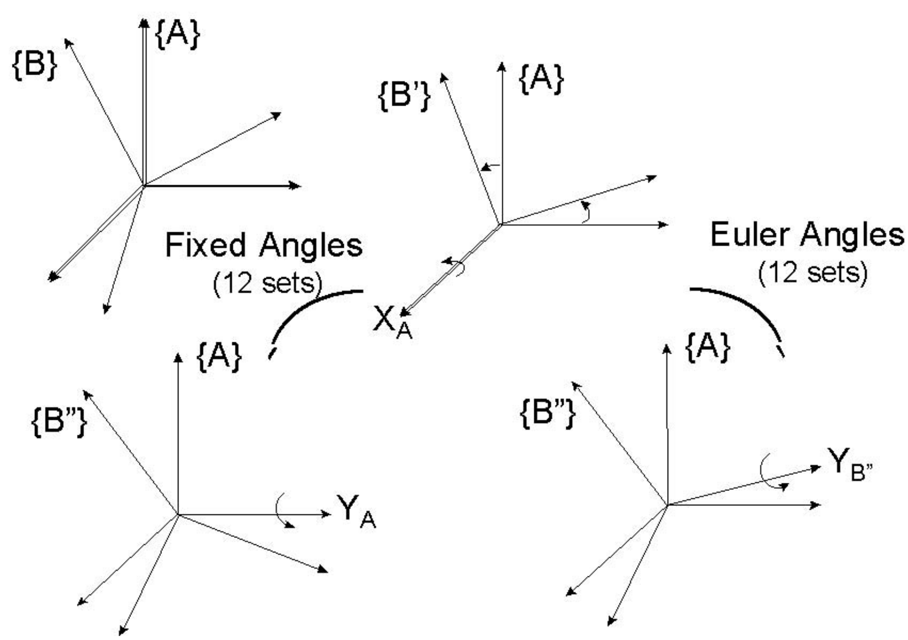

Figure 1.20: Three angles representation

Let us consider the two frames $\{A\}$ and $\{B\}$ illustrated in Figure 1.20. Imagine we start from the configuration where $\{B\}$ is coincident with frame $\{A\}$. To reach the final configuration of $\{B\}$, we can proceed in the following manner. First we rotate about X_A to achieve frame $\{B'\}$. Next we have two choices: (i) rotate with respect to one of the axes of $\{A\}$ that are fixed, or (ii) rotate with respect to one of the axes of $\{B'\}$, obtained from the first rotation. If we proceed with respect to the axes of $\{A\}$ we will obtain the so-called *fixed angles* representation. If we proceed with respect to the axes of $\{B'\}$ and continue with the new frames that we generate, we will obtain the *Euler angle* representation. In fact these two representations are dual and for every fixed angle representation there is a corresponding Euler angle representation.

We need to do a total of three rotations with three angles α, β and γ. There are 12 possibilities for both the fixed and the Euler angle representations. The notation XYZ means that we have a first rotation about axis X followed by a rotation about Y and finally a rotation about axis Z. Naturally we do not need to consider alternatives where there are two consecutive rotations about the same axis since they are equivalent to one rotation with the sum of the two angles.

Let us consider for example the $Z-Y-X$-Euler angle representation, illustrated in figure 1.21. Starting with frame $\{A\}$ we rotate about axis Z_A of an angle α to obtain frame $\{B'\}$. The corresponding rotation matrix is ${}^{A}_{B'}R = R_z(\alpha)$. Next we rotate about the newly generated Y axis of frame $\{B'\}$ of an angle β and obtain the frame $\{B''\}$. The rotation matrix here is ${}^{B'}_{B''}R = R_y(\beta)$. Finally we rotate about the axis X of $\{B''\}$ of an angle γ to obtain frame $\{B\}$ with a rotation matrix ${}^{B''}_{B}R = R_x(\gamma)$. The resulting matrix ${}^{A}_{B}R$ is simply the product

$${}^{A}_{B}R = {}^{A}_{B'}R \; {}^{B'}_{B''}R \; {}^{B''}_{B}R$$

or

$${}^{A}_{B}R \;=\; R_Z(\alpha)\; R_Y(\beta)\; R_X(\gamma)$$

Let us now consider the $X-Y-Z$-fixed angles representation. Fixed angle rotations can be best understood from the point of view of small instantaneous rotations. They also have an analog in aerospace engineering. Let us think in terms of airplane control with the Z axis pointing vertically up and the X axis pointing in the direction of the flight. Then rotation about the Z axes is *yaw*, rotation about the X axes is *roll* and rotation about the Y axes is *pitch*. The sequence now is: γ rotation about X of $\{A\}$, then a β rotation about Y of $\{A\}$ and a α rotation about Z of $\{A\}$.

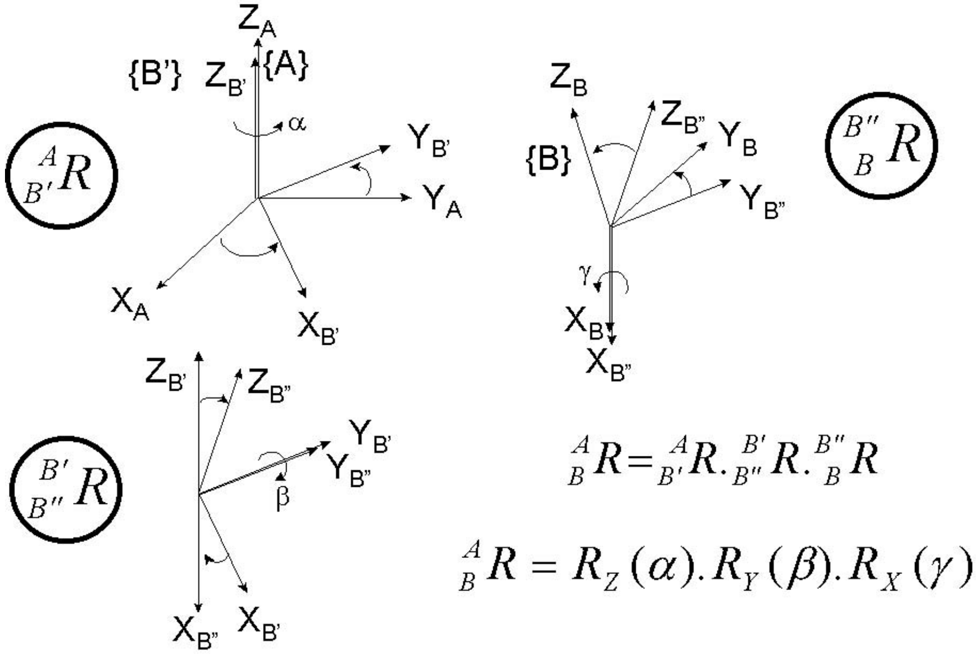

Figure 1.21: Euler Angles (Z-Y-X)

The rotation about X of an angle γ is a rotation operator which acts on a vector **v** transforming it into into $R_x(\gamma)v$. The result of the first operator is subject to the second rotation resulting in $R_y(\beta)(R_x(\gamma)v)$, and finally from the third rotation leads to $R_z(\alpha)(R_y(\beta)(R_x(\gamma)v))$. In summary

$$R_X(\gamma) : v \rightarrow R_X(\gamma)v \tag{1.37}$$

$$R_Y(\beta) : (R_X(\gamma)v) \rightarrow R_Y(\beta)(R_X(\gamma).v) \tag{1.38}$$

$$R_Z(\alpha) : (R_Y(\beta)R_X(\gamma)v) \rightarrow R_Z(\alpha)(R_Y(\beta)R_X(\gamma)v) \tag{1.39}$$

Comparing this result with the Euler angle rotation obtained above, we can see that the $X - Y - Z$-fixed angles rotation of angles (γ, β, α) is

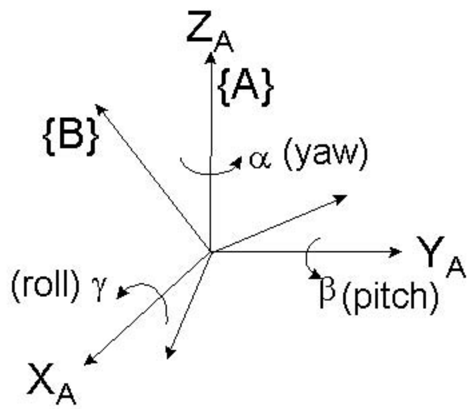

Figure 1.22: Fixed Angles (X-Y-Z)

equivalent to the $Z-Y-X$-Euler angles rotations of angles (α, β, γ). This illustrates the duality principle mentioned above.

$$
{}^{A}_{B}R = {}^{A}_{B}R_{XYZ}(\gamma, \beta, \alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma) \tag{1.40}
$$

1.3.3 Inverse of an Orientation Representation

The end-effector position and orientation is determined by the compound homogeneous transformation ${}^{B}_{E}T$ which will be computed as a function of joint angles through the so-called forward kinematics, as discussed in Chapter 2. The rotation part of this transformation ${}^{B}_{E}R$ contains the information about the orientation of the end-effector. Given the elements r_{ij} of the matrix ${}^{B}_{E}R$, the problem is to extract from this matrix the parameters selected to represent the orientation. For the Euler angle representation this is to find the three angles α, β and γ corresponding to the elements r_{ij} found from the forward kinematics (the propagation of the link parameters along the kinematic chain). For the $Z'Y'X'$ Euler angle representation with angles α, β and γ, the matrix ${}^{A}_{B}R$ is

$$ {}^A_BR = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} c\alpha.c\beta & c\alpha.s\beta.s\gamma - s\alpha.c\gamma & c\alpha.s\beta.c\gamma + s\alpha.s\gamma \\ s\alpha.c\beta & s\alpha.s\beta.s\gamma + c\alpha.c\gamma & s\alpha.s\beta.c\gamma - c\alpha.s\gamma \\ -s\beta & c\beta.s\gamma & c\beta.c\gamma \end{bmatrix} \tag{1.41} $$

We have used here the abbreviation $c\alpha$ and $s\alpha$ to denote $cos(\alpha)$ and $sin(\alpha)$.

There are 9 trigonometric equations with 3 independent parameters α, β and γ. In this case, β can be found from r_{11}, r_{21} and r_{31},

$$ \left. \begin{array}{r} \cos\beta = c\beta = \sqrt{r_{11}^2 + r_{21}^2} \\ \sin\beta = s\beta = -r_{31} \end{array} \right\} \rightarrow \beta = A\tan 2(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \tag{1.42} $$

This assumes that $r_{11}^2 + r_{21}^2$ is not zero. Knowing β, we can easily find α from r_{21} and r_{11}, and γ from r_{32} and r_{33}, namely:

$$ \alpha = Atan2(\frac{r_{21}}{c\beta}, \frac{r_{11}}{c\beta}) \tag{1.43} $$

$$ \gamma = Atan2(\frac{r_{32}}{c\beta}, \frac{r_{33}}{c\beta}) \tag{1.44} $$

We have used the 2-argument $Atan2(y, x)$ function which computes $tan^{-1}\left(\frac{y}{x}\right)$, but uses the signs of both x and y to determine the quadrant in which the resulting angle lies.

If $r_{11}^2 + r_{21}^2 = 0$, that means that $c\beta = 0,\ s\beta = \pm 1$ and we are at a *singularity of the representation*. Here it is for $\beta = \pm 90^o$. In this case, the angles α and γ cannot be determined. We can only determine $\alpha + \gamma$ or $\alpha - \gamma$. Namely, if $c\beta = 0,\ s\beta = +1$ then

$$ {}^A_BR = \begin{pmatrix} 0 & -s(\alpha-\gamma) & c(\alpha-\gamma) \\ 0 & c(\alpha-\gamma) & s(\alpha-\gamma) \\ -1 & 0 & 0 \end{pmatrix} \tag{1.45} $$

And if $c\beta = 0,\ s\beta = -1$ then

$$ {}^A_BR = \begin{pmatrix} 0 & -s(\alpha+\gamma) & -c(\alpha+\gamma) \\ 0 & c(\alpha+\gamma) & -s(\alpha+\gamma) \\ 1 & 0 & 0 \end{pmatrix} \tag{1.46} $$

As an example, for the configuration shown in Figure 1.23, from $R_{Z'Y'X'}(\alpha, \beta, \gamma)$ we can derive

$$ \alpha = 0, \ \beta = 0, \ \gamma = 90^o \tag{1.47} $$

Formulas for finding the corresponding rotation angles can be similarly obtained for all Euler or fixed angle representations.

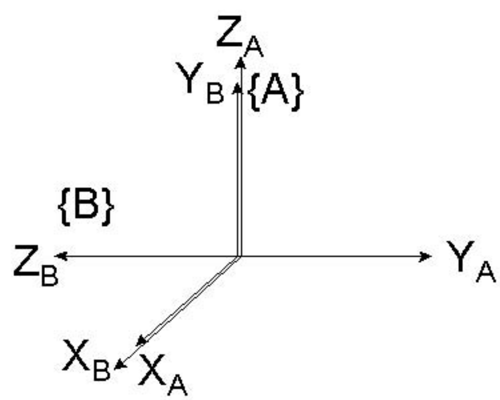

Figure 1.23: Example of a Z-Y-X Euler Rotation

1.3.4 Equivalent Angle - Axis Representation

So far we have considered rotations about the primary axes of the frames attached to the objects. It can be shown that any rotation from one frame to another can be represented by a rotation about some axis with some angle θ. Given two frames $\{A\}$ and $\{B\}$ with a common origin (i.e. one frame can be rotated into the other), we can find an axis K and an angle θ such that $\{B\}$ can be obtained from $\{A\}$ via a rotation about K of angle θ. This representation is called the *equivalent angle - axis* representation, which is illustrated in Figure 1.24.

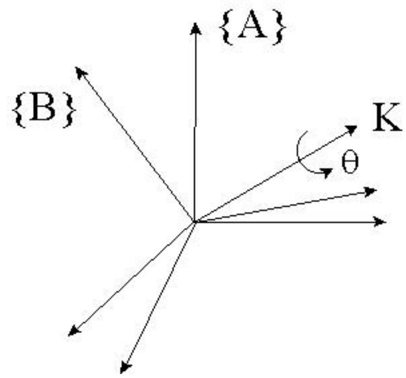

Figure 1.24: Equivalent angle - axis representation

If k_x, k_y, and k_z are the components of the unit vector $\mathbf{k}$, the rotation matrix about $\mathbf{k}$ with an angle θ can be represented by the vector $\mathbf{k}$ scaled by the angle θ as

$$X_r = \theta\mathbf{k} = \begin{bmatrix} \theta k_X \\ \theta k_Y \\ \theta k_Z \end{bmatrix} \tag{1.48}$$

To solve inversely for $\mathbf{k}$ and θ we can use

$$R_K(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix} \tag{1.49}$$

Here

$$v\theta = 1 - c\theta \tag{1.50}$$

Given a rotation matrix

$$R_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{1.51}$$

we can determine

$$\theta = Arc\cos(\frac{r_{11} + r_{22} + r_{33} - 1}{2}) \tag{1.52}$$

and

$$^{A}\mathbf{k} = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}, \ \textit{singularity for } \sin\theta = 0 \tag{1.53}$$

Note that this is a 3 parameter representation (two independent parameters for the unit vector **k** and one angle θ). Note also that for configurations where $\sin(\theta) = 0$, we will have a singularity of the representation.

The next representation avoids such singularities.

1.3.5 Euler Parameters

Sometimes we would like to have a redundant representation, i.e. use more than three parameters, but use as few as possible, ideally use four parameters.

With three parameters in some cases we can not access some of the configurations of the manipulator. Nine parameters (direction cosines) are far too many but with four we will have a minimal singularity-free representation.

To derive the Euler parameters we will use a unit vector **w** with components (w_x, w_y, w_z) and a rotation about it of an angle θ. The Euler parameters are

$$\varepsilon_1 = w_x \sin\frac{\theta}{2} \tag{1.54}$$

$$\varepsilon_2 = w_y \sin\frac{\theta}{2} \tag{1.55}$$

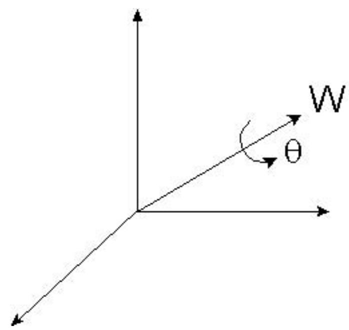

Figure 1.25: Euler parameters representation

$$\varepsilon_3 = w_z \sin\frac{\theta}{2} \tag{1.56}$$

$$\varepsilon_4 = \cos\frac{\theta}{2} \tag{1.57}$$

As we can see, the sum of the squares of these parameters is one,

$$|W| = 1, \quad \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \varepsilon_4^2 = 1 \tag{1.58}$$

This normality condition shows that only three of the parameters are independent. The parameters $(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$ define the unit hyper-sphere in four - dimensional space because of the Normality Condition.

The rotation matrix associated with Euler parameters is

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 1-2\varepsilon_2^2-2\varepsilon_3^2 & 2(\varepsilon_1\varepsilon_2-\varepsilon_3\varepsilon_4) & 2(\varepsilon_1\varepsilon_3+\varepsilon_2\varepsilon_4) \\ 2(\varepsilon_1\varepsilon_2+\varepsilon_3\varepsilon_4) & 1-2\varepsilon_1^2-2\varepsilon_3^2 & 2(\varepsilon_2\varepsilon_3-\varepsilon_1\varepsilon_4) \\ 2(\varepsilon_1\varepsilon_3-\varepsilon_2\varepsilon_4) & 2(\varepsilon_2\varepsilon_3+\varepsilon_1\varepsilon_4) & 1-2\varepsilon_1^2-2\varepsilon_2^2 \end{bmatrix} \tag{1.59}$$

The inverse problem for ε_4 is easy to solve since the sum of the diagonal elements is $4 * (\varepsilon_4)^2 - 1$ (using the Normality Condition), i.e.

$$r_{11} + r_{22} + r_{33} = 3 - 4(\varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2) = 4\varepsilon_4^2 - 1 \tag{1.60}$$

or

$$\varepsilon_4 = \frac{1}{2}\sqrt{1 + r_{11} + r_{22} + r_{33}} \tag{1.61}$$

The rest of the parameters are easily determined using ε_4 in the denominator, i.e.

$$\varepsilon_1 = \frac{r_{32} - r_{23}}{4\varepsilon_4}, \quad \varepsilon_2 = \frac{r_{13} - r_{31}}{4\varepsilon_4}, \quad \varepsilon_3 = \frac{r_{21} - r_{12}}{4\varepsilon_4} \tag{1.62}$$

The only possible singularity occurs when $\varepsilon_4 = 0$. As it turns out, we can do what we did with ε_4 for any of the other parameters. For example taking $r_{11} - r_{22} - r_{33}$ we can easily find ε_1, etc. Because of the normality condition, it is not possible for all four of the Euler parameters to be zero at the same time. Thus one of the ε_i's will be non-zero and we can always find an unique solution to the inverse problem. In fact the following is true:

Lemma: For all rotations, at least one of the Euler parameters is greater than or equal to 1/2.

Thus we can find the maximal ε and use that to solve for the rest of the parameters. In particular, if

$$\varepsilon_1 = \max_i \{\varepsilon_i\} \tag{1.63}$$

then:

$$\varepsilon_1 = \frac{1}{2}\sqrt{r_{11} - r_{22} - r_{33} + 1} \tag{1.64}$$

$$\varepsilon_2 = \frac{(r_{21} + r_{12})}{4\varepsilon_1} \tag{1.65}$$

$$\varepsilon_3 = \frac{(r_{31} + r_{13})}{4\varepsilon_1} \tag{1.66}$$

$$\varepsilon_4 = \frac{(r_{32} - r_{23})}{4\varepsilon_1} \tag{1.67}$$

Similarly, if

$$\varepsilon_2 = \max_i \{\varepsilon_i\} \tag{1.68}$$

then

$$\varepsilon_2 = \frac{1}{2}\sqrt{-r_{11} + r_{22} - r_{33} + 1} \tag{1.69}$$

If

$$\varepsilon_3 = \max_i \{\varepsilon_i\} \tag{1.70}$$

then

$$\varepsilon_3 = \frac{1}{2}\sqrt{-r_{11} - r_{22} + r_{33} + 1} \tag{1.71}$$

We can also find the Euler parameters using the Euler angles from the formulas below:

$$\varepsilon_1 = \sin\frac{\beta}{2}\cos\frac{\alpha - \gamma}{2} \tag{1.72}$$

$$\varepsilon_2 = \sin\frac{\beta}{2}\sin\frac{\alpha - \gamma}{2} \tag{1.73}$$

$$\varepsilon_3 = \cos\frac{\beta}{2}\sin\frac{\alpha + \gamma}{2} \tag{1.74}$$

$$\varepsilon_4 = \cos\frac{\beta}{2}\cos\frac{\alpha+\gamma}{2} \tag{1.75}$$

These formulas constitute another proof that the Euler parameters do not have singularities since we only use α, $\beta+\gamma$ and $\beta-\gamma$ in the above equations, which can be always determined uniquely.

In the next chapter we will define a set of link related parameters which will help us derive the forward kinematics formulas for a general manipulator.

1.4 Exercises

1. A vector ${}^AP$ is rotated about $\hat{Z}_A$ by ψ degrees and is subsequently rotated about $\hat{Y}_A$ by ϕ degrees. Give the rotation matrix which accomplishes these rotations in the given order. What is the result if $\psi = 45°$ and $\phi = 60°$?

2. A frame {B} is located as follows: initially coincident with a frame {A} we rotate {B} about $\hat{Y}_B$ by ϕ and then we rotate the resulting frame about $\hat{Z}_B$ by ψ degrees. Give the rotation matrix, ${}^A_BR$ which will change the description of vectors from ${}^BP$ to ${}^AP$. What is the result if $\psi = 45°$ and $\phi = 60°$?

3. A *velocity* vector is given by

$$ {}^BV = \begin{bmatrix} 6.0 \\ 1.0 \\ 5.0 \end{bmatrix}. $$

Given

$$ {}^B_AT = \begin{bmatrix} 0.8 & 0.0 & 0.6 & 4.0 \\ 0.0 & 1.0 & 0.0 & 7.0 \\ -0.6 & 0.0 & 0.8 & 2.5 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}, $$

compute ${}^AV$.

4. For sufficiently small rotations so that the approximations $\sin\theta \approx \theta$, $\cos\theta \approx 1$, and $\theta^2 \approx 0$ hold:

   (a) Derive the rotation matrix equivalent to a rotation of θ about a general unit axis $\hat{K}$. Start with Eqn. 1.49 for your derivation.

   (b) Show that two infinitesimal rotations commute; that is, show $R_K(\theta)R_{K'}(\phi) = R_{K'}(\phi)R_K(\theta)$.

5. Determine the Euler parameters $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ for the following rotation matrix:

$$R = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}.$$

Chapter 2

Kinematic Model of Manipulators

2.1 Direct Kinematics

The descriptions that we introduced so far are of general natureand are independent of the structure of a particular manipulator. The spatial descriptions allow us to calculate the position and orientation of the frame associated with the end-effector of a manipulator with respect to the frame associated with its fixed base. In this chapter we will introduce a set of parameters specific to robotic manipulators. They can be used to describe rotational and translational motion in the joints connecting the links of the manipulator. We will show how to establish relationships between these parameters for neighbouring links. We will use these relationships to propagate descriptions along the chain of links in a manipulator and derive its forward and inverse kinematic models.

2.1.1 Link Description

A manipulator consists of a chain of links starting from the base (typically fixed in the workspace) propagating to the end-effector (the gripper that interacts with the environment). Consecutive links are connected by joints which exert the degree of freedom of the motion of the link. For clarity of presentation we will mainly concern ourselves with simple rotational and translational motion of the links. By default each joint will have one degree of freedom (dof) - rotational or translational.

Let us consider the i-th link in the kinematic chain. This link is connecting two joints. At the end of link i there is an axis with respect to which the next link, $i+1$, is going to move. There is also an axis in the beginning of link i. Those two axes are lines in a three dimensional space. They are characterized by a common normal. This common normal has a length that we will call link length. This is one of the parameters describing link.

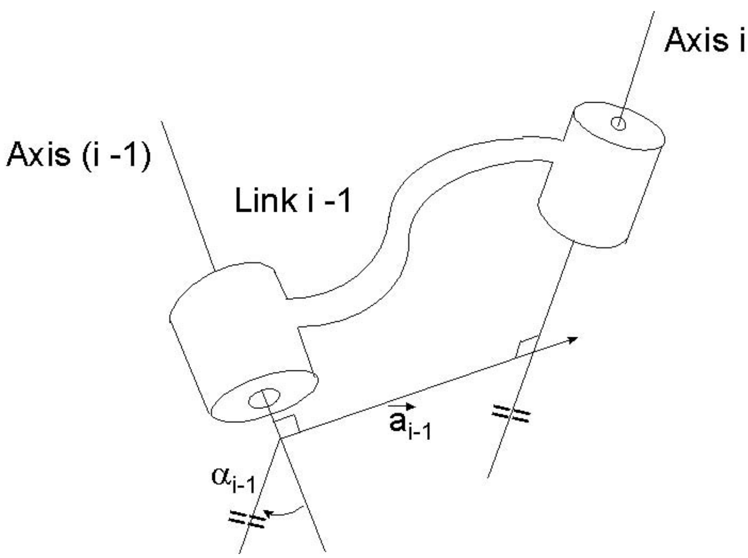

Figure 2.1: Link description

In Figure 2.1 a_{i-1} (the "link length") denotes the length along the common normal from axis $(i-1)$ to axis (i).

To define two axes in space, in addition to the common normal, we need to compute the angle between the axes. In particular we can draw a parallel line to axis (i) at the point where the common normal intersects the axis $(i-1)$. The angle between this parallel line and axis $(i-1)$, denoted by α_{i-1}, will be called "link twist". This angle is measured in the right- hand sense about the vector defined by a_{i-1} directed from axis $(i-1)$ to axis (i) along the common normal. Figure 2.1 illustrates this notation.

Often in industrial robots there are consecutive axes that intersect at a point. For example in the manipulator known as the "Stanford Sheinman Arm" which we will consider later, axes 2 and 3 intersect at a right angle. In this case the twist angle α_{i-1} is $+90^o$ or -90^o . As shown in Figure 2.2 when there are intersecting axes, the definition of link twist will be free in terms of direction.

Alternatively two consecutive axes can be parallel. This is the case in the example of three links planar manipulator that we used in the previous chapter. All three axes in that example are parallel and perpendicular to the plane of the manipulator. In that case the link twist is 0^o.

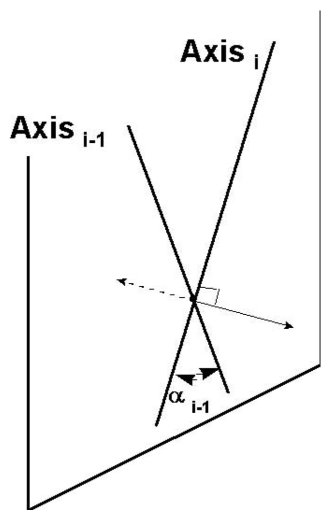

Figure 2.2: Intersecting joint axes

The link length and the link twist describe the configuration of a particular link in the kinematic chain. There are two other parameters that describe the connections between any two consecutive links. Link $(i-1)$ is determined by its corresponding joints $i-1$ and i. Let us consider the next link (i) in the chain. That link will have a link length, a_i, and angle, α_i. If we consider the line along axis (i), the distance along this line between the common normal for link $(i-1)$ and the common normal for link (i) (between axes (i) and $(i+1)$) is a parameter called "link offset". It is denoted by d_i in our terminology.

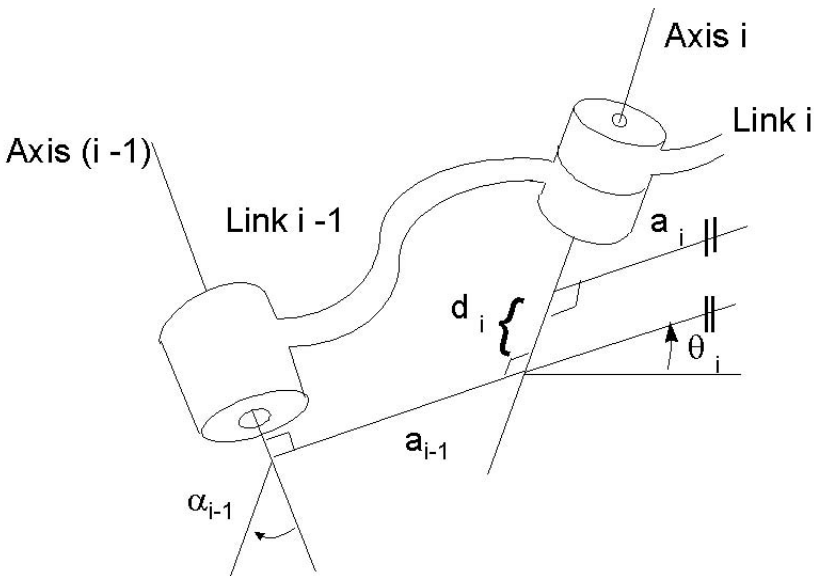

Figure 2.3: Relationship between links

The angle between the two common normals mentioned above, measured about axis (i), is the forth parameter in the representation. It is called "joint angle" and it is denoted by θ_i. All parameters are depicted in Figure 2.3.

The link length and the link twist are always constant since we consider only rigid links. However the link offset and the joint angle can be either constant or variable. In particular, for a revolute joint i the joint angle θ_i is variable and the link offset d_i is constant. Alternatively for a prismatic joint i, the link offset, d_i, is a variable and the joint angle, θ_i, is constant. For the three link planar manipulator from the previous chapter all joint angles are variable. For the Stanford Scheinman Arm one of the link offsets is a variable (along the prismatic joint).

We will introduce a convention for assigning the values of the four link parameters for unique representation of the links. Those parameters determine the complete relationship between link (i) and $(i-1)$. They will define the homogeneous transformation between those two links.

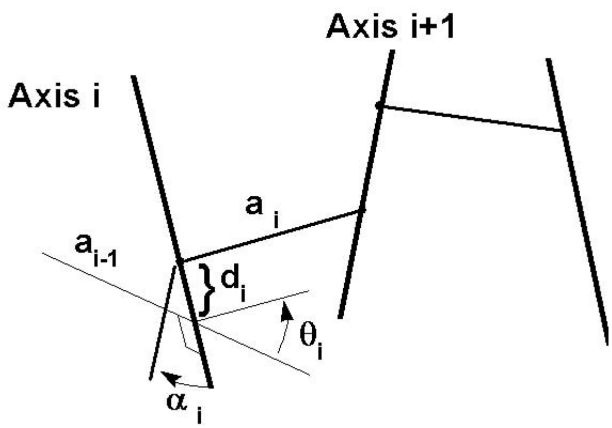

Figure 2.4: Frames propagation between links

The first and the last links in the chain need special convention consideration. Clearly from the definition above, a_i and α_i depend on axes (i) and $(i+1)$. We can also propagate those along the chain as depicted in Figure 2.4. Thus having the axes in space will define $a_1, a_2, \ldots, a_{n-1}$ and α_1, α_2, $\ldots$, α_{n-1}. We do have the freedom of choosing a_0, α_0 and a_n, α_n. By convention we try to assign zeroes to everything we can. Thus we will try to select $a_0 = \alpha_0 = a_n = \alpha_n = 0$. This will also determine how the frame attached to the base is selected. In other words we will select frame 0 and frame N so that the parameters above are zeroes.

Similar considerations can be used for the two other parameters θ_i and d_i. Those parameters by definition depend on links $(i-1)$ and (i). Just as above $\theta_2, \ldots, \theta_{N-1}$ and $d_2, \ldots, d_{N-1}$ are clearly defined.

For link 1 there are two options: In Figure 2.5, if axis 1 is a revolute one, we will select $d_1 = 0$. In that case θ_1 is variable depending on the motion of axis (1).

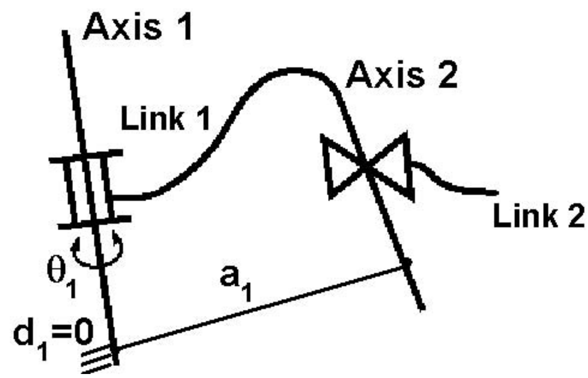

Figure 2.5: Revolute First Link

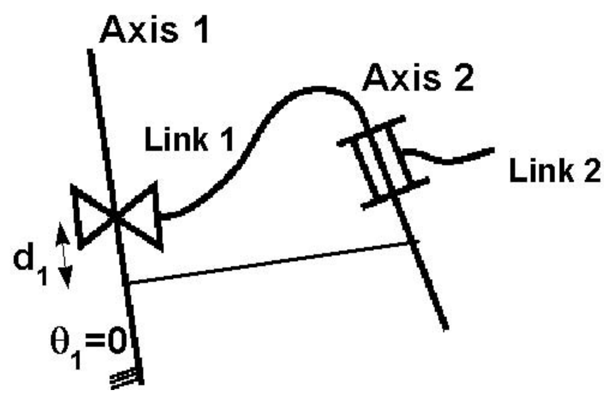

Figure 2.6: Prismatic First Link

If axis (1) is prismatic (as in Figure 2.6) we select $\theta_1 = 0$ since d_1 is variable. Similarly if axis N is revolute we select $d_N = 0$ and if it is prismatic, then $\theta_N = 0$.

These four parameters $(\alpha_i, a_i, \theta_i, d_i)$ describe the relationship between two links. They are called the Denavit-Hartenberg parameters. For each joint, three of these parameters will be fixed and one will be variable. That variable is either θ_i if the joint is revolute or d_i if it is prismatic. The first two parameters provide the description of the link itself and the next two describe the connection with the next link. A transformation between two successive links can be expressed in terms of these four parameters. In order to do that, we need to attach frames to the links.

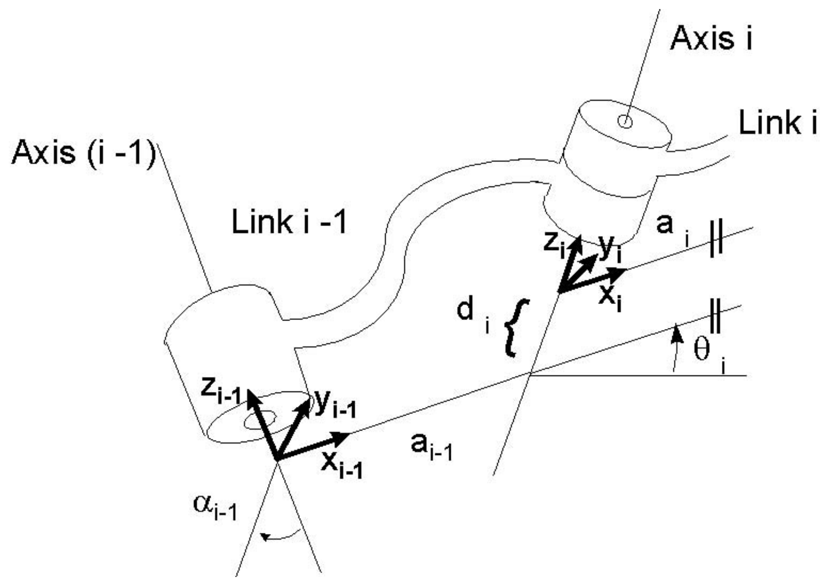

Figure 2.7: Affixing frames to links

The next step in the process is to attach frames to the links. By convention we select as the origin for frame $(i-1)$ the point at the intersection of the common normal between links $(i-1)$ and (i) with the axis $(i-1)$. Axis $(i-1)$ itself will be used for the Z_{i-1} axis and the X_{i-1} points along the common normal from axis $(i-1)$ to axis (i). The Y axis is selected to make a direct frame (Cartesian frame with the right hand rule).

The frame consisting of X_i, Y_i and Z_i is defined in a similar fashion. From that definition it can be seen that if joint (i) is revolute, the joint variable is the angle between X_{i-1} and X_i. If it is prismatic the distance d_i between X_{i-1} and X_i along Z_i is the variable. The assigned frames are depicted in Figure 2.7.

When there are intersecting axes we will need to deal with the freedom in assigning the directions. For example Figure 2.8 illustrates the case when axes (i) and $(i+1)$ are intersecting. In this case we take the

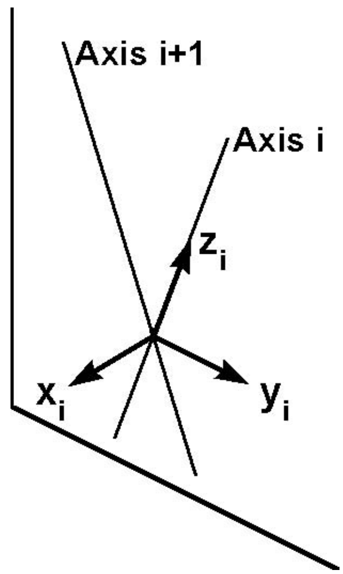

Figure 2.8: Frames on intersecting axes

perpendicular to both in the point of intersection and assign X_i along it in such direction that the angle is measured from axis (i) to $(i+1)$ in a positive sense. The origin of the frame is at the intersection of the two axes.

The first and the last link in the kinematic chain require special attention. Let us consider the first link of the mechanism with frame 0 attached to the fixed base. There are 2 possible cases: If the first joint is revolute (depicted in Figure 2.9), then frame 1 attached to link (1) rotates with respect to the fixed base. We have a freedom in selecting the reference frame 0. We will select it in such way so that $a_0 = \alpha_0 = d_1 = 0$. Thus when $\theta_1 = 0 \rightarrow \{0\} \equiv \{1\}$ (the zero and the first frame coincide). This assignment will simplify the computations. In some cases however we might want to choose frame 0 differently to facilitate measurements with respect to alternative fixed frames.

If the first joint is prismatic (as pictured in Figure 2.10), we can choose frame 0 to be parallel to frame 1 with a displacement d_1 between the origins. Thus $a_0 = 0, \alpha_0 = 0, \theta_1 = 0$. When $d_1 = 0$ the two frames are

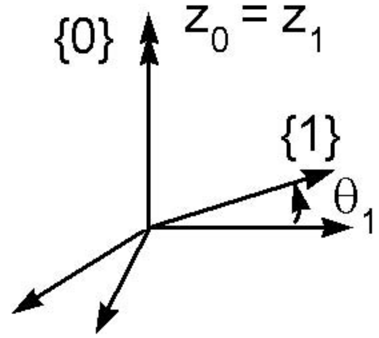

Figure 2.9: Frames on revolute first link

coincident again, i.e.

$$d_1 = 0 \rightarrow \{0\} \equiv \{1\} \tag{2.1}$$

The last link is handled similarly. If the last joint is revolute we set frame 0 so that $d_N = 0$, thus for $\theta_n = 0 \rightarrow x_n || x_{n-1}$ (the axes are collinear) as in Figure 2.11.

If the joint is prismatic, as in Figure 2.12, we will select frame 0 so that $\theta_N = 0$. Then if $d_N = 0$ the two X axes are again collinear.

To summarize, we introduced the four parameters in Figure 2.13: Link Length a_i is the distance between (Z_i, Z_{i+1}) along X_i. Link Twist α_i is the angle between (Z_i, Z_{i+1}) about X_i. Link Offset d_i is the distance between (X_{i-1}, X_i) along Z_i. Joint Angle θ_i is the angle between (X_{i-1}, X_i) about Z_i. At any time one of these four parameters will be variable and the rest will be constant.

There is a simple procedure that can be followed to define the frame attachment along the kinematic chain. We start by defining the axes along the joints. Next we define the common normals. The origins of the frames go at the intersections of those normals with the joint axes as in Figure 2.14. The Z-axes of the frames point along the joint axes at each of the origins. The X-axes point along the common normals

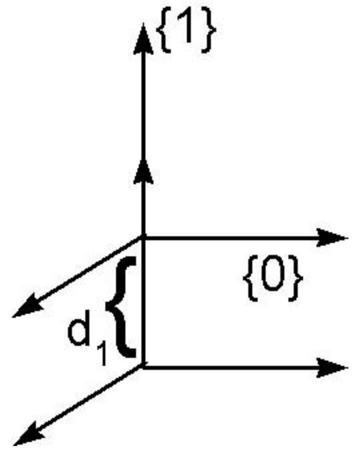

Figure 2.10: Frames on prismatic first link

at each of the origins. The Y-axes are defined by the right-hand rule, perpendicular to the X and Z axes at each of the origins.

2.1.2 Examples

Let us consider a simple example depicted in Figure 2.15 - a RRR (revolute-revolute-revolute) manipulator. This is a planar example with three joint variables - θ_1, θ_2, θ_3, that describe the rotation about each of the axes.

All Z axes are perpendicular to the plane of the manipulator. The X-axes are along the common normals as denoted. The origins of frames 1, 2 and 3 are at the joint axes and the Y axes are defined so that the frames are direct. X_3 is selected so that it is collinear with X_2 for $\theta_3 = 0$ and X_2 is selected to be collinear with X_1 when $\theta_2 = 0$. Similarly the origin of frame 0 coincides with the origin of frame 1 and X_0 is collinear with X_1 when $\theta_1 = 0$.

For clarity of the representation we will arrange the D&H (Denavit and Hartenberg) parameters in a table, where for each link (i) the entries are: $\alpha_{i-1}, a_{i-1}, d_i$ and θ_i. The parameters are arranged in such order

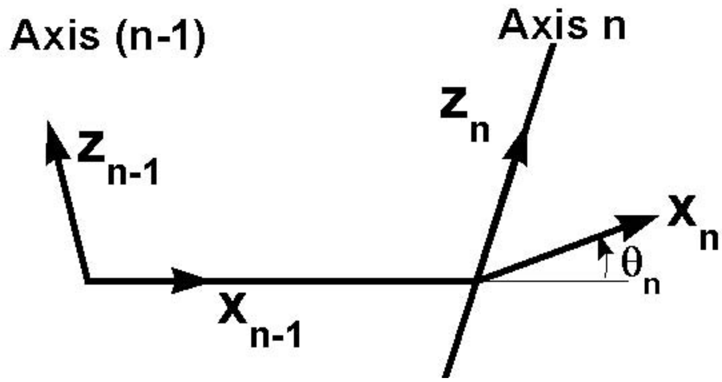

Figure 2.11: Frames on last revolute link

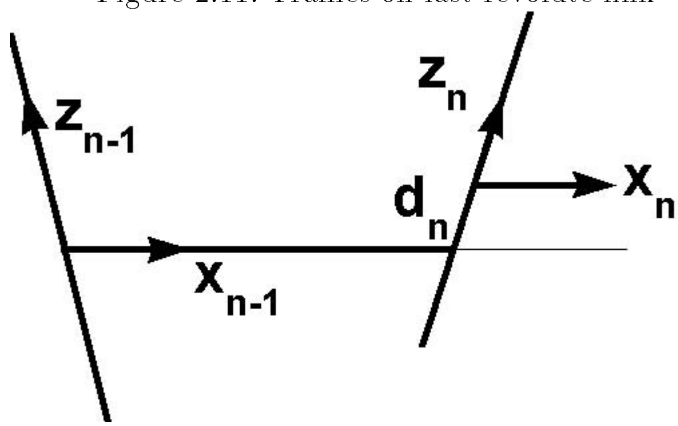

Figure 2.12: Frames on last prismatic link

because, as we will see later, that combination of parameters is used to compute the transformation matrix from frame $i-1$ to frame i.

$$\begin{array}{ccccc} i & \alpha_{i-1} & a_{i-1} & d_i & \theta_i \\ 1 & 0 & 0 & 0 & \theta_1 \\ 2 & 0 & l_1 & 0 & \theta_2 \\ 3 & 0 & l_2 & 0 & \theta_3 \end{array} \tag{2.2}$$

Since all Z axes are parallel (perpendicular to the plane of the manipulator), $\alpha_0 = \alpha_1 = \alpha_2 = 0$. We chose frame 0 so that $\alpha_0 = d_1 = 0$. Because all X axes are in the same plane, $d_1 = d_2 = d_3 = 0$. The joint

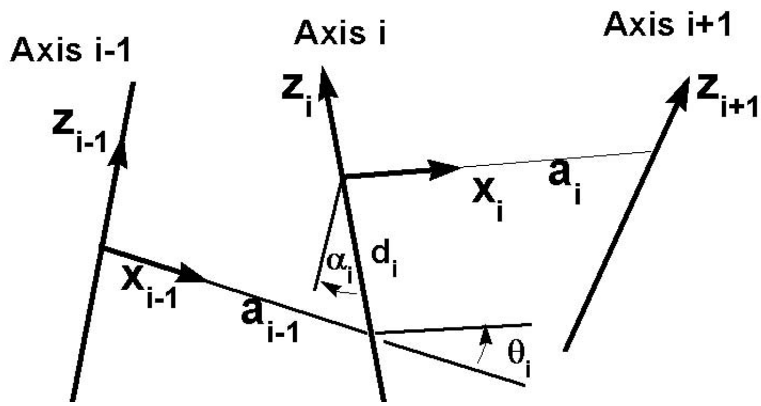

Figure 2.13: Frames assignment

angles θ_1, θ_2 and θ_3 are variables, and a_1 and a_2 are constants denoted by l_1 and l_2 in the example.

Another more complicated example is described in the Figure 2.16. The mechanism in the Figure starts with a revolute joint (denoted by the tapered cylinder). The first joint is connected to the ground as denoted by the slanted marks on the first axis. The output of the first joint is itself an axis for the second joint which is prismatic. This is denoted by the two small triangles with a common vertex in the figure. This joint translates along its axis. The third joint is revolute and perpendicular to the plane of the paper - it is denoted by the circle with a point in the middle (a top view of the joint). The output of the corresponding link is the fourth joint of the mechanism which is also revolute. At the end of the mechanism there is a gripper, the symbol of which is shown in the figure.

Typically a mechanism like this is given in some configuration. To describe it, we need to assign frames, find the D&H parameters and build the table for the mechanism. The frames positions are depicted in the figure. Note that axis Z_3 is perpendicular to the plane of the paper and is represented as a point. Note also that there is freedom in the assignment of frame 2, which is typically resolved by moving from

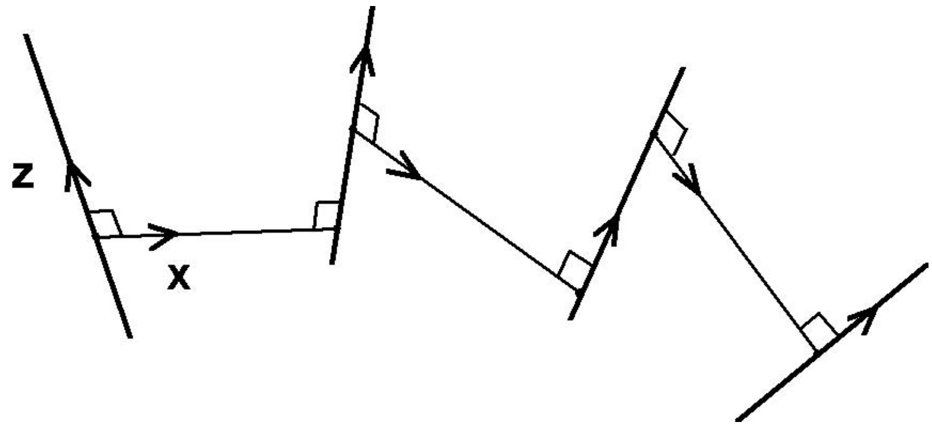

Figure 2.14: Frame attachment propagation

link (0) to link (1) and assigning the Z axis pointing upwards.

In this example the origins O_3 and O_4 coincide because the third and the fourth axes intersect at that point (and are perpendicular). This configuration is sometimes called a "wrist point" and it is very common for a number of manipulators. In addition to the four frames associated with the links, we can introduce a frame 5 at the end-effector point. This frame is selected so that its origin is at the point defined by the end-effector and the corresponding frame is parallel to frame 4. To complete the table of D&H parameters we need to introduce the distances L_2, L_4 and L_5.

If the axes were assigned alternatively (downward vs. upward), some of the angles in the D&H table would have been -90^o rather than $+90^o$. For every static configuration of the mechanism we can also add a column in the table that shows the values of the variables at that state (e.g. $\theta_1 = 0$, $d_2 = L_1$, $\theta_3 = 0$, $\theta_4 = 180^o$).

We will leave it as an exercise for the reader to build the corresponding D&H table.

2.1.3 Propagation of Frames

The important role of the D&H parameters is in determining the transformation matrices between the frames. For link $(i-1)$ with joints $(i-1)$

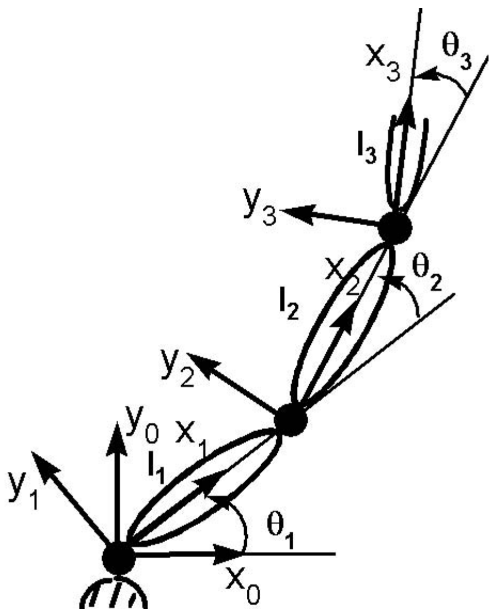

Figure 2.15: RRR Manipulator

and (i) we can draw frames $i-1$ and i as shown in Figure 2.17. We can also determine $\alpha_{i-1}, a_{i-1}, d_i$ and θ_i. These parameters will be used to calculate the transformation matrix ${}^{(i-1)}_{i}T$ from frame $i-1$ to frame i.

In order to compute this transformation matrix, we need to introduce three additional frames. First we will translate frame i to frame P so that the origin of frame P lies on the common perpendicular of the axes $(i-1)$ and (i). The transformation T_i^P is a simple operator of translation D_z along the Z axis with magnitude d_i. The second frame we will introduce is denoted by Q and is the result of a simple rotation R_z about the axis Z at an angle θ_i so that the X axis of frame Q is along the common perpendicular. The next step is to translate frame Q along the common perpendicular to a new frame R whose origin coincides with the origin of frame $i-1$. This is a simple translation D_x along the X axis at a distance a_{i-1}. Finally, we can rotate frame R

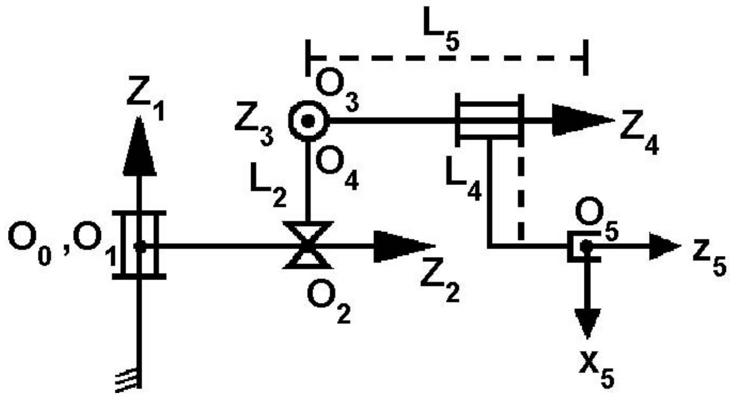

Figure 2.16: RPRR Mechanism

into frame $i-1$ via a simple rotation R_x about the X axis on an angle α_{i-1}. The compound transformation that takes us from frame $i-1$ to frame i is:

$$ {}^{i-1}_{i}T = {}^{i-1}_{R}T^{R}_{Q}T^{Q}_{P}T^{P}_{i}T \tag{2.3}$$

This matrix can be written as the product of the operators above, i.e

$$ {}^{i-1}_{i}T(\alpha_{i-1}, a_{i-1}, \theta_i, d_i) = R_x(\alpha_{i-1})D_x(a_{i-1})R_z(\theta_i)D_z(d_i) \tag{2.4}$$

If we multiply the matrices corresponding to the simple rotations and translations about the major axes, the result is:

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i.c\alpha_{i-1} & c\theta_i.c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}.d_i \\ s\theta_i.s\alpha_{i-1} & c\theta_i.s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}.d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

This is the principal formula describing the relationship between two successive frames using the link parameters.

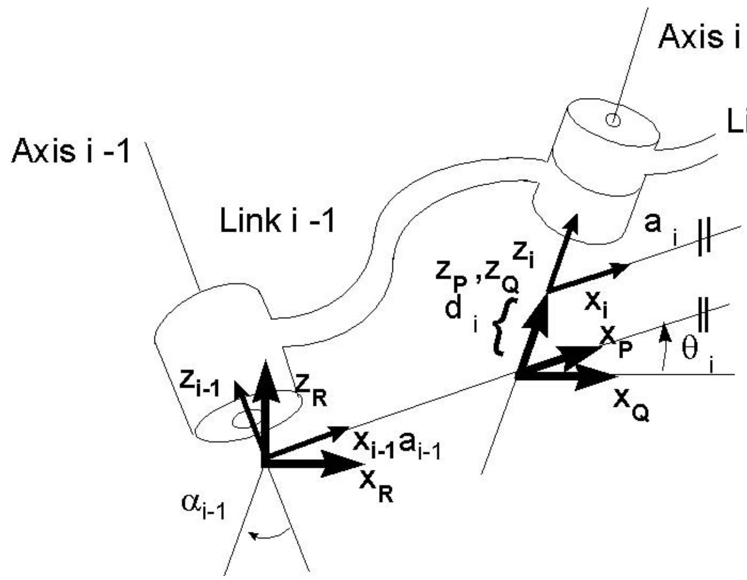

Figure 2.17: Frames for manipulator kinematics

2.1.4 Kinematics of Manipulators

Using the kinematic chain and the relationship from the previous section, we can compute the transformation from the base link (0) of the manipulator to the last end-effector link by multiplying the transformation matrixes for the consecutive frames

$$ {}^{0}_{N}T = {}^{0}_{1}T {}^{1}_{2}T \ldots {}^{N-1}_{N}T \tag{2.6} $$

In the case of the Stanford Sheinman Arm, the frame attachment is illustrated in Figure 2.18. Note that the offset of the arm is denoted as d_2 in the figure and it is NOT a variable. The six variables are θ_1, θ_2, d_3, θ_4, θ_5 and θ_6. The wrist point is where the origins of frames $4, 5$ and 6 are located. The D&H table is given below. The overall transformation matrix is computed in the Appendix.

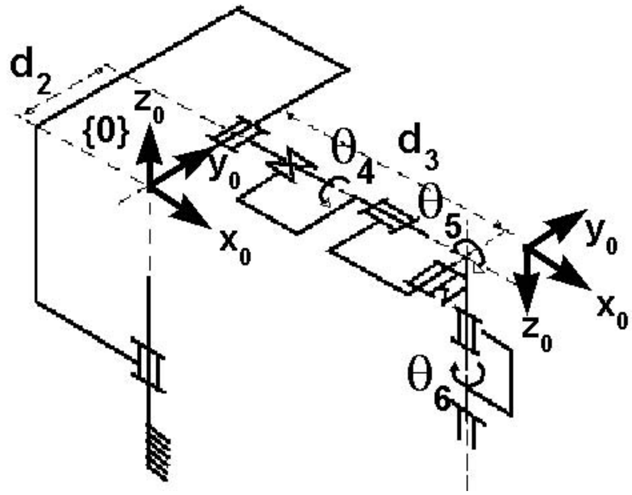

Figure 2.18: Frames on Stanford Sheinman's arm

$$
\begin{array}{ccccc}
 & \alpha_{i-1} & a_{i-1} & d_i & \theta_i \\
1 & 0 & 0 & 0 & \theta_1 \\
2 & -90^o & 0 & d_2 & \theta_2 \\
3 & 90^o & 0 & d_3 & 0 \\
4 & 0 & 0 & 0 & \theta_4 \\
5 & -90^o & 0 & 0 & \theta_5 \\
6 & 90^o & 0 & 0 & \theta_6
\end{array}
\tag{2.7}
$$

The following set of equations depicts a representation for the final position and orientation of the mechanism as a function of the joint variables and parameters. The position is given in Cartesian coordinates and the orientation is given by the direction cosines of the end-effector in frame 0.

$$\begin{bmatrix} X_R \\ X_P \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ X_P \end{bmatrix} = \begin{bmatrix} C_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] - S_1(S_4C_5C_6 + C_4S_6) \\ S_1[C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6] + C_1(S_4C_5C_6 + C_4S_6) \\ -S_2(C_4C_5C_6 - S_4S_6) - C_2S_5C_6 \\ C_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] - S_1(-S_4C_5S_6 + C_4C_6) \\ S_1[-C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6] + C_1(-S_4C_5S_6 + C_4C_6) \\ S_2(C_4C_5S_6 + S_4C_6) + C_2S_5S_6 \\ C_1(C_2C_4S_5 + S_2C_5) - S_1S_4S_5 \\ S_1(C_2C_4S_5 + S_2C_5) + C_1S_4S_5 \\ -S_2C_4S_5 + C_2C_5 \\ C_1S_2d_3 - S_1d_2 \\ S_1S_2d_3 + C_1d_2 \\ C_2d_3 \end{bmatrix} \quad (2.8)$$

2.1.5 Direct Kinematics

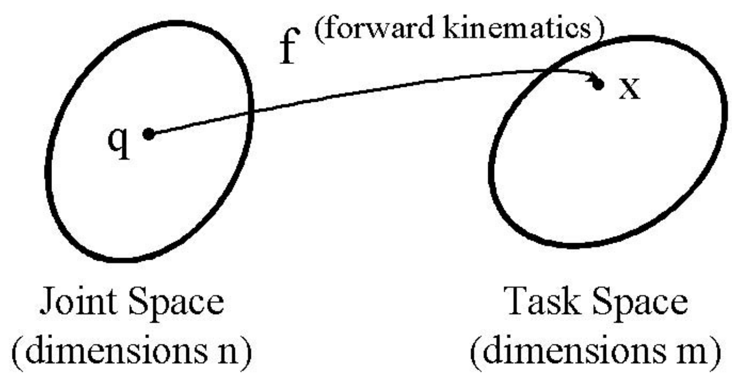

Figure 2.19: Direct Kinematics Mapping

The representation that we have discussed so far is known as the forward (or direct) kinematics of the mechanism. As depicted in Figure 2.19 it is a mapping between the joint space of dimension n and the task space of the manipulator of dimension m.

The joint space is formed by all possible values for the joint variables.

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ . \\ . \\ . \\ q_n \end{bmatrix} \tag{2.9}$$

The parameters (q_i) represent angles (θ_i) for revolute joints and displacements (d_i) for prismatic joints. The common notation for the cases of revolute and prismatic joints is:

$$q_i = \bar{\varepsilon}_i \theta_i + \varepsilon_i d_i \tag{2.10}$$

where

$$\varepsilon_i = \left[\left\{ \begin{matrix} 0 & \mathit{revolute\ joint} \\ 1 & \mathit{prismatic\ joint} \end{matrix} \right. \right] \tag{2.11}$$

In that notation

$$\bar{\varepsilon}_i \equiv 1 - \varepsilon_i \tag{2.12}$$

The task space of the manipulator is formed by all possible values for the position and orientation of the end-effector of the manipulator.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_m \end{bmatrix} \tag{2.13}$$

The relationship $\mathbf{x} = f(\mathbf{q})$ describes the forward kinematics. Given the function f for any set of values for the joint variables, we can find the corresponding task coordinates of the manipulator.

For a vector $\mathbf{q}$

$$\mathbf{q} = (\, q_1 \quad q_2 \quad \dots \quad q_n \,)^T \tag{2.14}$$

${}^0_nT = {}^0_nT(q)$ defines the forward kinematics and can be written as:

$$\mathbf{x} = f(\mathbf{q}) \tag{2.15}$$

This relationship is sometimes called the "Geometric Model" of the manipulator because it is determined solely by knowing the geometry of the manipulator.

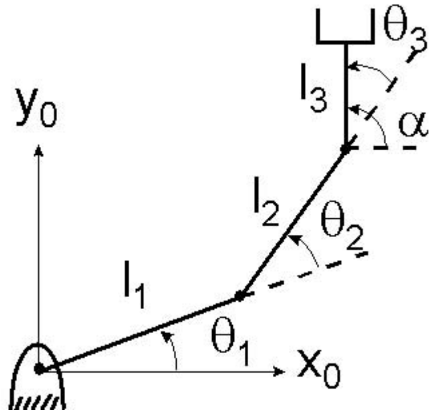

Figure 2.20: 3 dof example

For the example that we considered earlier in Figure 2.20 the vector $X(x, y, \alpha)$ for the position and the orientation of the planar RRR manipulator is:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix} = \begin{bmatrix} l_1c_1 + l_2c_{12} \\ l_1s_1 + l_2s_{12} \\ \theta_1 + \theta_2 + \theta_3 \end{bmatrix} \tag{2.16}$$

Note: The notation c_{12} denotes $\cos(\theta_1+\theta_2)$. Similarly $s_{12} = \sin(\theta_1+\theta_2)$.

The orientation α of the end-effector is simply the sum of the joint angles $\theta_1 + \theta_2 + \theta_3$ (as expected for a planar mechanism). Using this representation, for any given set of joint variables $\mathbf{q}$ we can find a unique position and orientation of the end-effector. Unfortunately the inverse is not true. If we are given the vector $\mathbf{x}$, there are a number of possible values of $\mathbf{q}$ that will result in the same $\mathbf{x}$ using the formulas above. Finding these values of $\mathbf{q}$ is the goal of the Inverse Kinematics method which we will describe next.

2.2 Inverse Kinematics

2.2.1 Existence and Multiplicity of Solutions

Finding the inverse kinematics of a mechanism is a difficult task because of the multiplicity or non-existence of potential solutions. The formulas defining the direct kinematics typically involve trigonometric equations (when revolute joints are present). Solving these equations for the joint angles and the link offsets is not at all trivial. At the same time solving the inverse kinematics is an important practical problem. Usually the goal for the manipulator motion is defined in task coordinates and there is a need to be able to quickly compute the necessary joint variables trajectory for achieving this motion.

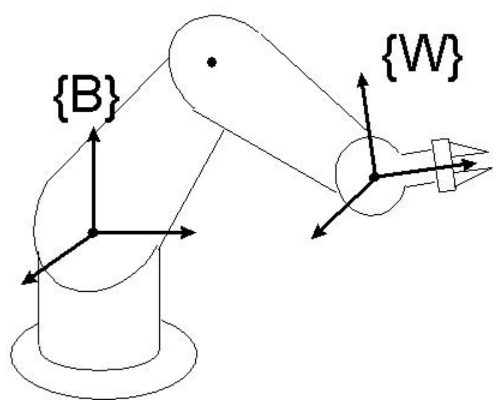

Figure 2.21: Base to Wrist frame kinematics

In the general case, if we consider the 6 dof manipulator pictured in Figure 2.21 with joint variables $q_1, q_2, \ldots, q_6$, the forward kinematics is given by the transformation from the base frame B to the wrist frame W - ${}^{B}_{W}T = {}^{0}_{6}T(q_1, q_2, q_3, q_4, q_5, q_6)$.

The end- effector position X_P and orientation X_R form the vector

$$\mathbf{x} = \begin{bmatrix} X_P \\ X_R \end{bmatrix} = f(\mathbf{q}) \tag{2.17}$$

The inverse problem is to find $\mathbf{q}$ given ${}^{B}_{W}T$ or $\mathbf{x}$, or to find $\mathbf{q} = f^{-1}(\mathbf{x})$. In matrix form this problem can be written as:

$${}^{0}_{6}T(q_1, q_2, q_3, q_4, q_5, q_6) = {}^{B}_{W}T \tag{2.18}$$

The right side of the equation is known (e.g. as numbers) and the system has 12 equations with 6 unknowns (the joint variables). Out of those 12 equations only 6 are independent (3 for the position and 3 for the orientation). Thus effectively we have 6 equations with 6 unknowns. However the equations are highly non-linear and involve trigonometric functions (if revolute joints are present). The system typically have an infinite number of solutions, but can occasionally have no solutions at all.

We will illustrate the notion of multiple solutions with some examples in the next section.

2.2.2 Closed Form Solutions

There are two possible approaches to solving the general system described in the equation above - **algebraic** and **geometric**. We will illustrate these approaches in the example of the 3 dof mechanism used throughout this chapter (see figure 2.20).

Consider first the **geometric** solution. Using the cosine theorem in Figure 2.22 we can solve for the second joint angle θ_2. The equation:

$$l_1^2 + l_2^2 + 2l_1l_2 \cos\theta_2 = x_0^2 + y_0^2 \tag{2.19}$$

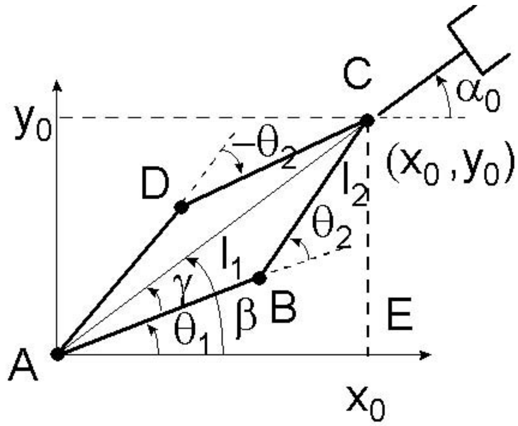

Figure 2.22: Geometric Solution example

gives us

$$\cos\theta_2 = \frac{(x_0^2 + y_0^2) - (l_1^2 + l_2^2)}{2l_1 l_2} \tag{2.20}$$

from which we can determine two solutions: θ_2 and $-\theta_2$. Similarly we can find θ_1, i.e. the cosine theorem gives us

$$l_2^2 = l_1^2 + (x_0^2 + y_0^2) - 2l_1\sqrt{x_0^2 + y_0^2}\cos\gamma \tag{2.21}$$

from which

$$\cos\gamma = \frac{x_0^2 + y_0^2 + l_1^2 - l_2^2}{2l_1\sqrt{x_0^2 + y_0^2}} \tag{2.22}$$

We can also compute the angle β from

$$\tan\beta = \frac{y_0}{x_0} \tag{2.23}$$

The first joint variable θ_1 is simply

$$\theta_1 = \beta \pm \gamma \tag{2.24}$$

depending on the sign of θ_2. Since we have computed θ_1 and θ_2, we can find θ_3 from:

$$\theta_3 = \alpha_0 - (\theta_1 + \theta_2) \tag{2.25}$$

Overall there are two possible solutions, based on the two different values for θ_2.

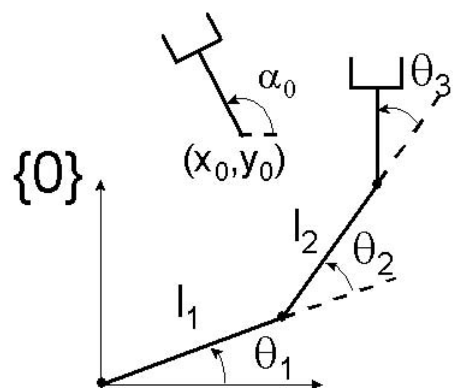

Figure 2.23: Algebraic Solution example

We can solve the same problem using an **algebraic** approach depicted in Figure 2.23. The starting point is the relationship defining the forward kinematics ${}^0_3T \equiv {}^B_WT$. We can write this explicitly as:

$$\begin{pmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\alpha_0 & -s\alpha_0 & 0 & x_0 \\ s\alpha_0 & c\alpha_0 & 0 & y_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.26}$$

The right side of this equation is derived following the observation that the end-effector point is at a position (x_0, y_0) with respect to the base frame, and at an angle α_0 with the X_0 axis of the base frame. The left side was derived using the propagation of D&H parameters described in the previous section.

Equating the elements (1,1) and (2,1) on both sides of the equation we can obtain:

$$\left.\begin{array}{l}\cos(\theta_1+\theta_2+\theta_3)=\cos\alpha_0\\ \sin(\theta_1+\theta_2+\theta_3)=\sin\alpha_0\end{array}\right\}\Rightarrow\theta_1+\theta_2+\theta_3=\alpha_0 \tag{2.27}$$

To find θ_1 and θ_2 we use elements (1,4) and (2,4) of the matrix:

$$l_1c_1+l_2c_{12}=x_0 \tag{2.28}$$

and

$$l_1s_1+l_2s_{12}=y_0 \tag{2.29}$$

In order for (x_0, y_0) to be in the workspace of the manipulator, we need:

$$-1\le\cos\theta_2=\frac{(x_0^2+y_0^2)-(l_1^2+l_2^2)}{2l_1l_2}\le 1 \tag{2.30}$$

From this relationship we obtain

$$\theta_2=A\tan 2(\pm\sqrt{1-\cos^2\theta_2},\cos\theta_2) \tag{2.31}$$

Now we can again use formulas (2.29) and (2.30) for θ_1 rewritten as:

$$\left.\begin{array}{l}(l_1+l_2c_2)c_1-(l_2s_2)s_1=x_0\\ (l_1+l_2c_2)s_1+(l_2s_2)c_1=y_0\end{array}\right\} \tag{2.32}$$

which can also be written as

$$\left.\begin{array}{l}k_1c_1-k_2s_1=x_0\\ k_1s_1+k_2c_1=y_0\end{array}\right\} \tag{2.33}$$

In the above formula we have grouped all terms depending on the variable θ_2 in the functions k_1 and k_2. An alternative representation for k_1 and k_2 is:

$$(k_1, k_2) \xrightarrow[\tan\gamma = k_2/k_1]{r = \sqrt{k_1^2 + k_2^2}} \begin{pmatrix} k_1 = r\cos\gamma \\ k_2 = r\sin\gamma \end{pmatrix} \tag{2.34}$$

Using that representation we can obtain

$$x_0 = r\cos(\theta_1 + \gamma) \tag{2.35}$$

$$y_0 = r\sin(\theta_1 + \gamma) \tag{2.36}$$

Solving the last two equations for θ_1 we can obtain:

$$\theta_1 = A\tan 2(y_0, x_0) - A\tan 2(k_2, k_1) \tag{2.37}$$

To complete the algebraic solution, θ_3 can be found using the fact that $\theta_1 + \theta_2 + \theta_3 = \alpha_0$.

As in the geometric case there are two pssible solutions of the inverse kinematics problem determined by the two possible values for θ_2 in the formulas above.

2.2.3 Pieper's solution

The same approach can be used to find the inverse kinematics solution for a general six dof mechanism with the last three axes intersecting (see Figure 2.24). This solution is part of a class of solutions first derived by Pieper in his thesis work at Stanford University.

The last three frames of the mechanism have the same origin - point $P = O_4$. Let us consider the representation of the vector from the fixed origin of the manipulator to that point in the different frames of the figure. In frame 1 this vector is ${}^1\mathbf{p} = {}^1_2T(\theta_2){}^2\mathbf{p}$. In vector form the representation can be written as:

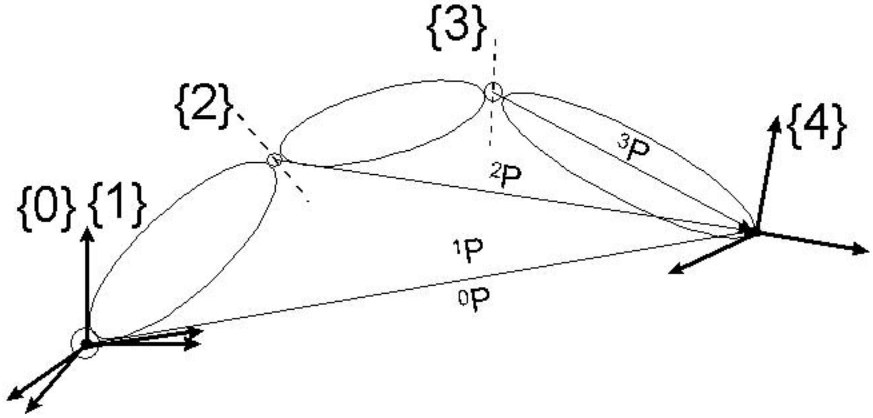

Figure 2.24: Pieper's solution example

$$^{1}\mathbf{p} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 1 \end{bmatrix} \tag{2.38}$$

where $g_i = g_i(c_2, s_2, f_i)$ are functions of θ_2 and the remainder of the terms grouped in the expressions f_i-s.

In frame 3 the same vector is

$$^{3}\mathbf{p} = \begin{bmatrix} a_3 \\ -s\alpha_3.d_4 \\ c\alpha_3.d_4 \\ 1 \end{bmatrix} \tag{2.39}$$

Since the coordinates of $^{2}\mathbf{p}$ are only dependent on the variable θ_3 we can write

$$^{2}\mathbf{p} = \begin{bmatrix} f_1(\theta_3) \\ f_2(\theta_3) \\ f_3(\theta_3) \\ 1 \end{bmatrix}, \qquad ^{2}\mathbf{p} = {}^{2}_{3}T(\theta_3)\,^{3}\mathbf{p} \tag{2.40}$$

and similarly

$$
{}^{0}\mathbf{p} = \begin{bmatrix} c_1 g_1 - s_1 g_2 \\ s_1 g_1 + c_1 g_2 \\ g_3 \\ 1 \end{bmatrix}, \qquad {}^{0}\mathbf{p} = {}^{0}_{1}T(\theta_1)\,{}^{1}\mathbf{p} \tag{2.41}
$$

These relationships are resolved moving forward along the chain. We know that ${}^{0}\mathbf{p} \equiv \mathbf{p}_0$. From there:

$$
\begin{bmatrix} c_1 g_1 - s_1 g_2 \\ s_1 g_1 + c_1 g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \tag{2.42}
$$

From the first two elements

$$
\left.\begin{matrix} c_1 g_1 - s_1 g_2 = x_0 \\ s_1 g_1 + c_1 g_2 = y_0 \end{matrix}\right\} \theta_1 \; if \;\; g_1 \; and \; g_2 \;\; are \; known \tag{2.43}
$$

Thus θ_1 can be computed as

$$
\theta_1 \;\; = Atan2(y_0, x_0) - Atan2(g_2, g_1) \tag{2.44}
$$

From the equation above we can derive the following relationships for θ_2:

$$
\theta_2 : \left[\{ \begin{matrix} g_1^2 + g_2^2 + g_3^2 = x_0^2 + y_0^2 + z_0^2 = r_0^2 \\ g_3 = Z_0 \end{matrix} \right] \tag{2.45}
$$

Here $g_i = g_i(c_2, s_2, f_1, f_2, f_3)$. Those equations can be re-written as;

$$
(k_1 c_2 + k_2 s_2) 2a_1 + k_3 = r_0^2 \tag{2.46}
$$

$$
(k_1 s_2 - k_2 c_2) s\alpha_1 + k_4 = Z_0 \tag{2.47}
$$

where $k_i = k_i(f_1, f_2, f_3)$. Thus θ_2 is known if k_i-s are known.

If we solve the last set of equations for θ_3 we obtain:

$$\theta_3 : (r_0^2 - k_3)^2 s^2\alpha_1 + (Z_0 - k_4)^2 4a_1^2 = 4a_1^2 s^2\alpha_1(k_1^2 + k_2^2) \tag{2.48}$$

where

$$k_i = k_i(f_i(c_3, s_3)) \tag{2.49}$$

Finally we will use the method of "Reduction to Polynomials" to solve the transcedental equations for k_i. This method relies on a change of variable $u = \tan\frac{\theta}{2}$ which reduces to:

$$u = \tan\frac{\theta}{2} \Rightarrow \left[\left\{ \begin{array}{l} \cos\theta = \frac{1-u^2}{1+u^2} \\ \sin\theta = \frac{2u}{1+u^2} \end{array} \right.\right] \tag{2.50}$$

If we use this change of variables for θ_3 and k_i we can denote $\theta_3 : k_i = k_i(u, u^2)$ and obtain:

$$Au^4 + Bu^3 + Cu^2 + Du + E = 0, \quad with\ u = \tan\frac{\theta_3}{2} \tag{2.51}$$

There are a number of well known methods for solving 4th degree polynomials.

Having solved this equation, we can find u (which will give us θ_3), then k_i and work back along the chain to solve for the rest of the parameters.

The last three variables θ_4, θ_5 and θ_6 will be computed using the equation ${}^0_6R(\Theta) \equiv R_0$ which can also be written as:

$${}^0_6R(\Theta) = {}^0_1R(\theta_1){}^1_2R(\theta_2){}^2_3R(\theta_3){}^3_4R(\theta_4){}^4_5R(\theta_5){}^5_6R(\theta_6) \tag{2.52}$$

The first three matrices in the right side of the equation depend on the variables that we have already computed. They can be grouped in one known term. The fourth matrix can be written as:

$${}^3_4R(\theta_4) = {}^3_4R|_{\theta_4=0}R_Z(\theta_4) \tag{2.53}$$

and the overall equation can be expressed as:

$$[{}^0_4R|_{\theta_4=0}(\theta_1,\theta_2,\theta_3)][R_Z(\theta_4){}^4_5R(\theta_5,\theta_6)] = R_0 \tag{2.54}$$

Another way to write this equation is:

$$R(\theta_4,\theta_5,\theta_6) = R_0' \tag{2.55}$$

In that representation we have moved all the known quantities (computed using θ_1, θ_2 and θ_3) on the right side of the equation. We have also combined them with matrix R_0 to form the now known matrix R_0' on the right.

The equation above can be solved for θ_4, θ_5 and θ_6 using the Euler Angle approach (among others).

That completes the algebraic solution for the 6 dof revolute manipulator using Pieper's solution. Many of the commercially used robots have a geometry that matches this configuration or are similarly designed to allow an explicit algebraic approach.

When the algebraic or the geometric approaches are not possible, the only option is to use Iterative Solutions to the inverse kinematics problem. However there are a number of cases where a solution to the inverse kinematics problem does not exist. We will illustrate this in the next section.

2.2.4 Existence of Solution

Since the kinematic equations are often described by trigonometric equations we might not have a solution to these equations. Let us consider the following example: Figure 2.25 depicts a 3 dof manipulator with all axes perpendicular to the surface of the paper at the points of the joints. The manipulator is drawn in some configuration. The task is to move the manipulator to a certain final configuration at point (x_f, y_f) and orientation α_f.

We can derive the forward kinematics equations as described by the matrix T_3^0. θ_1, θ_2 and θ_3 are the joint variables for the revolute joints.

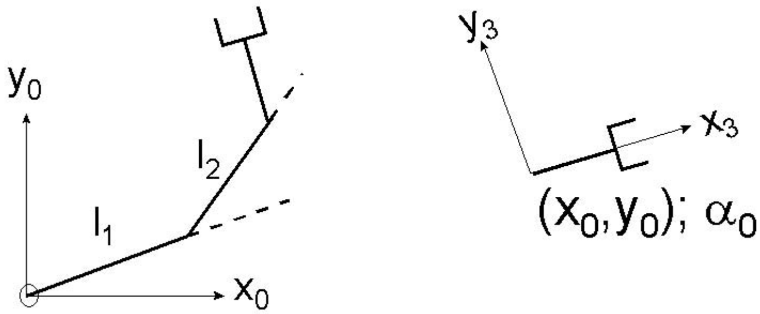

Figure 2.25: Existence of solution example

We can also write this transformation matrix in terms of the position and orientation of the end-effector as shown in the following equation.

$$ {}^0_3T = \begin{pmatrix} c_{123} & -s_{123} & 0 & l_1c_1 + l_2c_{12} \\ s_{123} & c_{123} & 0 & l_1s_1 + l_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c\alpha_0 & -s\alpha_0 & 0 & x_0 \\ s\alpha_0 & c\alpha_0 & 0 & y_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.56} $$

In order for the solution to exist, we need to find values for the joint variables θ_1, θ_2 and θ_3 that make these two matrices equal and possible. One condition like that is:

$$ (l_1 - l_2)^2 \leq x_0^2 + y_0^2 \leq (l_1 + l_2)^2 \tag{2.57} $$

In this example $\theta_1 + \theta_2 + \theta_3 = \alpha$ and if the condition above is satisfied, from the first two elements of the last column we can find θ_1 and θ_2 giving us a solution to the inverse kinematics problem.

The points that can be reached by the manipulator are depicted in Figure 2.26. In that figure the center represents the fixed base of the manipulator. Consider first all the points in the inside circle with radius $l_1 - l_2$. In this case l_1 (the length of the first link) is larger than l_2 (the

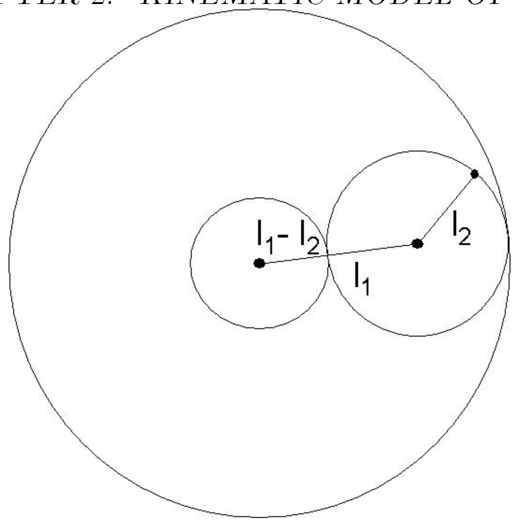

Figure 2.26: Workspace of a manipulator

length of the second link). None of the points in the inside circle can be reached geometrically by the manipulator. We can also come to this conclusion by looking at the forward kinematic expressions, i.e. there will not be a solution that allows us to reach these points.

The points that we can reach with the manipulator define the "workspace" of the manipulator. In the figure above, the points that can be reached by the manipulator lie in the area enclosed by the large outside circle (with radius $l_1 + l_2$) and the small inside circle (with radius $l_1 - l_2$). For all other points, there is no solution to the inverse kinematics problem.

In the example so far we have assumed that the links can rotate through a full 360^o around their axes (by convention from -180^o to $+180^o$). However in practice there are always "joint limits" defined by the mechanical design of the manipulator. For the example of our manipulator with joint limits defined as:

$$0 \leq \theta_1 \leq 180^o \tag{2.58}$$

$$0 \leq \theta_2 \leq 180^o \tag{2.59}$$

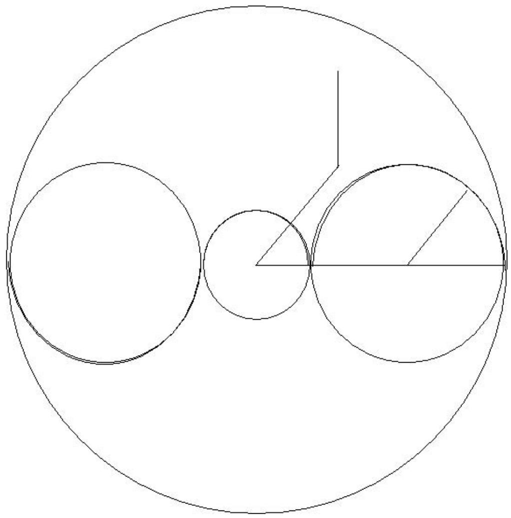

Figure 2.27: Workspace with joint limits

the workspace is the area about the highlighted region in Figure 2.27 (much more complicated than the case in Figure 2.26 and close to half of its workspace).

If there are inverse kinematics solutions for the manipulator (i.e. the workspace is not empty), there are still a number of interesting questions to consider. One of them is the question about the **number of possible solutions**. To analyze this topic better we will consider two types of workspace - Reachable and Dextrous workspace.

One and the same point in the manipulator workspace can be reached via different configurations of the manipulator. "**Reachable** workspace" is the set of points that can be reached in at least one configuration of the manipulator. Conversely "**dextrous** workspace" is the set of points that can be reached by any possible orientation of the end-effector. Obviously the dextrous workspace is a subset of the reachable workspace.

Consider the example of three links RRR manipulator in Figure 2.28 with lengths of the links $l_1 > l_2 > l_3$. The reachable workspace is the donut defined by the outside circle (with a center at the base of the manipulator and a radius $l_1+l_2+l_3$) and the inside circle (with a center at the base of the manipulator and a radius $l_1 - l_2 - l_3$). On the other

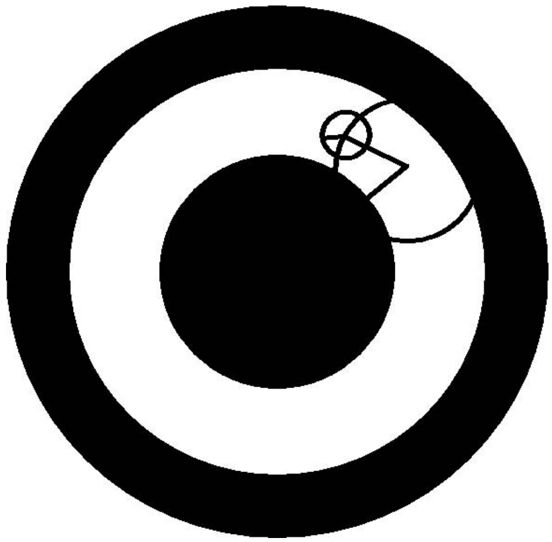

Figure 2.28: Dextrous workspace example

side the dextrous workspace is the inside unshaded donut with center at the manipulator base and radius: for the outside circle $l_1 + l_2 - l_3$ and for the inside circle $l_1 - l_2 + l_3$.

Dextrous workspace is especially important in motion planning with obstacles when we need to approach and depart from certain positions with different orientations of the end-effector, or we need to regrasp the objects in the workspace for transportation.

By definition, in the dextrous workspace there are an infinite number of solutions for the inverse kinematics problem. However even in the reachable workspace we can have more than one configuration for the manipulator reaching a given point. In that case we might want to be able to choose one of those solutions to work with. This choice is usually dictated by the rest of the problem setup. If for example we are trying to move from point A to point B in the workspace in Figure 2.29, we can choose the manipulator configuration for point B that is more easily and continuously reached from point A. Analytically this is described by choosing the smaller distance between

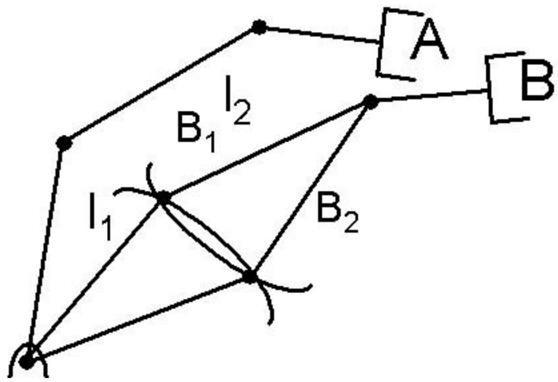

Figure 2.29: Multiplicity of solutions

$$C_1 = \left\| \Theta_{(B_1)} - \Theta_{(A)} \right\| \tag{2.60}$$

and

$$C_2 = \left\| \Theta_{(B_2)} - \Theta_{(A)} \right\| \tag{2.61}$$

Analogously we might prefer to move smaller links or assembly of links rather than to move large links, and use that as a the criteria for choice of solution.

The question of number of solutions for classes of manipulators have been studied heavily in the robotics literature. As a result there are a number of theoretical results that determine the number of solutions for particular manipulators. For example it has been shown that for a 6 dof manipulator with revolute joints for which all link parameters are non zero , there are 16 possible solutions for the inverse kinematics problem. If one of the link parameters is zero, we still have 16 solutions to the inverse kinematics. However if two link parameters are zeroes there are only 8 solutions, and if three parameters are zeroes we are down to 4 possible solutions. These results are only for the number of solutions and do not tell us about the range of motion of the manipulator.

We can summarize some of these theoretical results as follows: 6R manipulators have exactly 16 solutions (some of them might be the same), 5RP manipulators have 16 solutions, 4R2P manipulators have 8 solutions, 3R3P manipulators have 2 solutions. Theoretically we can have 16, 14, 12, ..., 2 possible solutions in general.

For in-parallel structures (which we do not consider in this text) there are up to 40 possible solutions.

The number of solutions for the Puma robot are illustrated graphically in Figure 2.30.

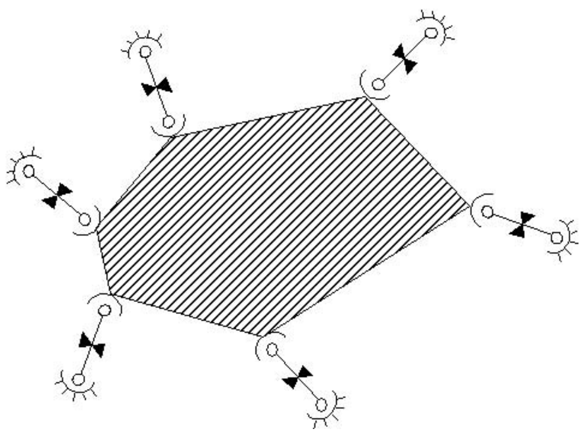

Figure 2.30: 8 solutions for a Puma manipulator

There are 4 different manipulator solutions shown in the figure that achieve the same position and orientation of the end-effector. In addition we can substitute θ_4 with $\theta_4 - 180^o$ and θ_5 with $-\theta_5$. Thus we have 8 possible solutions for that manipulator, for:

$$\theta_4 \rightarrow \theta_4 + 180^o \tag{2.62}$$

$$\theta_5 \rightarrow -\theta_5 \tag{2.63}$$

$$\theta_6 \rightarrow \theta_6 + 180^o \tag{2.64}$$

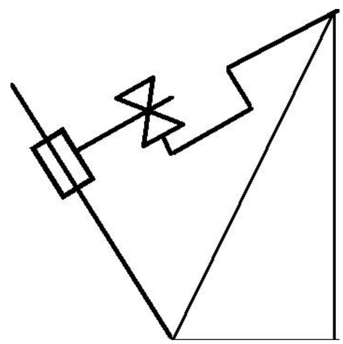

Figure 2.31: 2 solutions for Scheinman arm

For the Stanford Scheinman Arm there are 2 possible configurations for the manipulator which are depicted in Figure 2.31.

In general a manipulator is considered "solvable" if all possible sets of solutions can be determined. We need to be able to find all possible solutions at all possible points with a particular algorithm. It has been shown that 6 dof open-chain mechanisms are solvable. We can either find a "closed-form" solution (an analytical one) or a numerical solution. The numerical solutions typically involve a certain degree of approximation and can be achieved by numerous possible approaches. We will concentrate in this text on closed-form, explicit algebraic solutions to the inverse kinematics problem. These solutions typically exist for a large class of mechanisms. In addition, almost all robots in practice are specifically designed so that they have an explicit closed-form inverse kinematics solution.

There are different sufficient conditions that can guarantee the existence of closed-form solutions. In this section we described the case when the manipulator has three intersecting neighboring axes. Pieper and other researchers have shown that closed-form solutions exist as well when the intersecting axes are either prismatic or revolute and they can be anywhere within the kinematic chain.

Chapter 3

The Jacobian

3.1 Introduction

We have thus so far established the mathematical models for the forward kinematics and inverse kinematics of a manipulator. These models describe the relationships between the static configurations of a mechanism and its end-effector. The focus in this chapter is on the models associated with the velocities and static forces of articulated mechanisms and the Jacobian matrix which is central to these models.

Assuming the manipulator is at a given configuration, $\mathbf{q}$, let us imagine that all its joints undertook a set of infinitesimally small displacements, represented by the vector $\delta\mathbf{q}$. At the end effector, there will be a corresponding set of displacements of the position and orientation $\mathbf{x}$, represented by the vector $\delta\mathbf{x}$. The goal in this chapter is to establish the relationship between $\delta\mathbf{x}$ and $\delta\mathbf{q}$. By considering the time derivatives of $\mathbf{x}$ and $\mathbf{q}$, this same relationship can be viewed as a relationship between the velocities $\dot{\mathbf{x}}$ and $\dot{\mathbf{q}}$. The relationship between $\dot{\mathbf{x}}$ and $\dot{\mathbf{q}}$ is described by the Jacobian matrix. Because of the duality between forces and velocities, this matrix as we will see later in this chapter is key to the relationship between joint torques and end effector forces.

3.2 Differential Motion

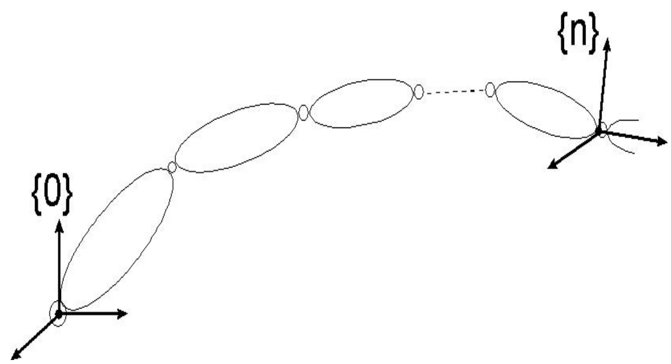

Figure 3.1: A Manipulator

Let us consider the function **f** that maps the space defined by variable **q** to the space defined by the variable **x**. Both **q** and **x** are vector variables (n and m- dimensional resp.), related by

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} f_1(q) \\ f_2(q) \\ \vdots \\ f_m(q) \end{pmatrix} \tag{3.1}$$

As described above we can consider the infinitesimal motion of the relationship $\mathbf{x} = \mathbf{f}(\mathbf{q})$. If we write it for each component of **x** and **q** we can derive the following set of equations for $\delta x_1, \delta x_2, \ldots, \delta x_m$ as functions of $\delta q_1, \delta q_2, \ldots, \delta q_n$

$$\delta x_1 = \frac{\partial f_1}{\partial q_1}\delta q_1 + \cdots + \frac{\partial f_1}{\partial q_n}\delta q_n \tag{3.2}$$

$$\begin{aligned} \vdots &= \vdots \\ \delta x_m &= \frac{\partial f_m}{\partial q_1}\delta q_1 + \cdots + \frac{\partial f_m}{\partial q_n}\delta q_n \end{aligned} \tag{3.3}$$

The above equations can be written in vector form as follows

$$\delta\mathbf{x} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \delta\mathbf{q} \tag{3.4}$$

The matrix in the above relationship is called the Jacobian matrix and is function of $\mathbf{q}$.

$$J(\mathbf{q}) \equiv \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \tag{3.5}$$

In general, the Jacobian allows us to relate corresponding small displacements in different spaces. If we divide both sides of the relationship by small time interval (i.e. differentiate with respect to time) we obtain a relationship between the velocities of the mechanism in joint and Cartesian space.

$$\dot{\mathbf{x}}_{(m\times 1)} = J(\mathbf{q})_{m\times n}\dot{\mathbf{q}}_{(n\times 1)} \tag{3.6}$$

3.2.1 Example: RR Manipulator

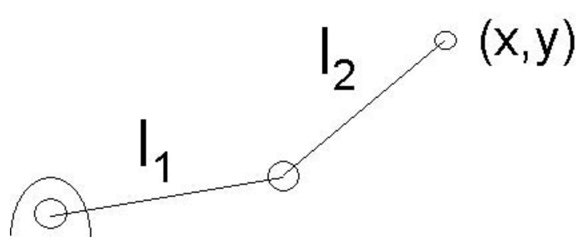

Figure 3.2: A 2 link example

The Jacobian is a mxn matrix from its definition. To illustrate the Jacobian, let us consider the following example. Take a two link manipulator in the plane with revolute joints and axis of rotation perpendicular to the plane of the paper. Let us first derive the positional part of a Jacobian. First from the forward kinematics we derive the description of the position and orientation of the end-effector in Cartesian space with respect to the joint coordinates θ_1 and θ_2.

$$x = l_1 c_1 + l_2 c_{12} \tag{3.7}$$

$$y = l_1 s_1 + l_2 s_{12} \tag{3.8}$$

The instantaneous motion of the position vector (x, y) is

$$\delta x = -(l_1 s_1 + l_2 s_{12})\delta\theta_1 - l_2 s_{12}\delta\theta_2 \tag{3.9}$$

$$\delta y = (l_1 c_1 + l_2 c_{12})\delta\theta_1 + l_2 c_{12}\delta\theta_2 \tag{3.10}$$

If we group the coefficients in front of $\delta\theta_1$ and $\delta\theta_2$ we obtain a matrix equation which can be written as

$$\delta x = \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \begin{bmatrix} -y & -l_2 s_{12} \\ x & l_2 c_{12} \end{bmatrix} \begin{pmatrix} \delta\theta_1 \\ \delta\theta_2 \end{pmatrix} \tag{3.11}$$

The $2x2$ matrix in the above equation is the Jacobian, $J(\mathbf{q})$.

$$\delta\mathbf{x} = J(\mathbf{q})\delta\mathbf{q} \tag{3.12}$$

As we can see this matrix is a function of the vector $\mathbf{q} = (\theta_1, \theta_2)$.

$$J \equiv \begin{pmatrix} \frac{\partial x}{\partial\theta_1} & \frac{\partial x}{\partial\theta_2} \\ \frac{\partial y}{\partial\theta_1} & \frac{\partial y}{\partial\theta_2} \end{pmatrix} \tag{3.13}$$

Now if we consider the differentiation *w.r.t.* time, we can write the relationship between $\dot{\mathbf{x}}$ and $\dot{\mathbf{q}}$.

$$\dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}} \tag{3.14}$$

3.2.2 Example: Stanford Scheinman Arm

As another example, we describe below the Jacobian associated with the end effector position of the Stanford Scheinman arm. The first three joint variables here are θ_1, θ_2 and d_3. From the forward kinematics we can observe that the position of the end-effector as a function of θ_1, θ_2 and d_3 is:

$$\mathbf{x}_p = \begin{bmatrix} c_1 s_2 d_3 - s_1 d_2 \\ s_1 s_2 d_3 + c_1 d_2 \\ c_2 d_3 \end{bmatrix} \tag{3.15}$$

If we differentiate with respect to the joint vector $(\theta_1, \theta_2, d_3, \theta_4, \theta_5, \theta_6)$ we obtain the following Jacobian for the position of the end-effector.

$$\dot{\mathbf{x}}_p = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{bmatrix} -y & c_1 c_2 d_3 & c_1 s_2 & 0 & 0 & 0 \\ x & s_1 c_2 d_3 & s_1 s_2 & 0 & 0 & 0 \\ 0 & -s_2 d_3 & c_2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} \tag{3.16}$$

We defined the position part as $\mathbf{x}_p$ and the corresponding part of the Jacobian will be denoted as J_p.

$$\dot{\mathbf{x}}_{p_{(3\times1)}} = J_{p_{(3\times6)}}(\mathbf{q})\dot{\mathbf{q}}_{(6\times1)} \tag{3.17}$$

For the orientation we will derive a Jacobian associated with the end-effector orientation representation, $\mathbf{x}_r$.

$$\dot{\mathbf{x}}_r = J_r(\mathbf{q})\dot{\mathbf{q}} \tag{3.18}$$

In our example the orientation part is given in terms of direction cosines $(r_{11}, r_{12}, \ldots, r_{33})$. When we differentiate those *w.r.t.* the joint variables, we will obtain the Jacobian for this orientation representation.

$$\mathbf{x}_r = \begin{bmatrix} \mathbf{r}_1(q) \\ \mathbf{r}_2(q) \\ \mathbf{r}_3(q) \end{bmatrix} \tag{3.19}$$

$$\dot{\mathbf{x}}_r = \begin{pmatrix} \dot{\mathbf{r}}_1 \\ \dot{\mathbf{r}}_2 \\ \dot{\mathbf{r}}_3 \end{pmatrix}_{(9\times 1)} = \begin{pmatrix} \frac{\partial \mathbf{r}_1}{\partial q_1} & \dots & \frac{\partial \mathbf{r}_1}{\partial q_6} \\ \frac{\partial \mathbf{r}_2}{\partial q_1} & \dots & \frac{\partial \mathbf{r}_2}{\partial q_6} \\ \frac{\partial \mathbf{r}_3}{\partial q_1} & \dots & \frac{\partial \mathbf{r}_3}{\partial q_6} \end{pmatrix}_{(9\times 6)} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_6 \end{pmatrix}_{(6\times 1)} \tag{3.20}$$

This is a 9×6 matrix because we are using the redundant direction cosines representation for the orientation. As time derivatives the relationship between $\dot{\mathbf{q}}$ and $\dot{\mathbf{x}}_r$ (the derivative of the orientation) is described by J_r (Jacobian of the orientation). Finally we can put the position and the orientation part together below.

$$\dot{\mathbf{x}}_p = J_p(\mathbf{q})\dot{\mathbf{q}} \tag{3.21}$$

$$\dot{\mathbf{x}}_r = J_r(\mathbf{q})\dot{\mathbf{q}} \tag{3.22}$$

The above equations can be combined as

$$\begin{pmatrix} \dot{\mathbf{x}}_p \\ \dot{\mathbf{x}}_r \end{pmatrix} = \begin{pmatrix} J_p(\mathbf{q}) \\ J_r(\mathbf{q}) \end{pmatrix} \dot{\mathbf{q}} \tag{3.23}$$

We can see that this Jacobian is a 12×6 matrix.

$$\dot{\mathbf{x}}_{(12\times 1)} = J_x(\mathbf{q})_{(12\times 6)}\dot{\mathbf{q}}_{(6\times 1)} \tag{3.24}$$

We should also note that so far we have not used any explicit frame in which we are describing those quantities, i.e. these equations are valid for any common frame that the variables are described in.

The above matrix is clearly dependent on the end effector representation. If we have selected a different representation for the orientation or the position of the end-effector we will obtain a different Jacobian matrix.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_r \end{bmatrix}$$

$$\mathbf{x} = \mathbf{f}(\mathbf{q}) \longrightarrow J_x = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}}$$

Typically the position, $\mathbf{x}_p$ is represented by the three Cartesian coordinates of a point on the end- effector (x, y, z). However we can also use spherical or cylindrical coordinates for that end-effector point and this will lead to a different Jacobian J_p. The orientation can be also described by different sets of parameters - Euler angles, direction cosines, Euler parameter, equivalent axis parameters, etc.

Depending on the representation used we will have different dimension of the orientation component of the Jacobian - $3 \times n$ for Euler angles, $9 \times n$ for direction cosines, $4 \times n$ for Euler parameters or equivalent axis parameters, where n is the number of degrees of freedom of the mechanism.

3.3 Basic Jacobian

We will introduce a unique Jacobian that is associated with the motion of the mechanism.

As we mentioned earlier, the Jacobian we have talked so far about depends on the representation used for the position and orientation of the end-effector.

If we use spherical coordinates for the position and direction cosines for the orientation we will obtain one Jacobian (12 for 6 DOF robot) very different from the one that results from Cartesian coordinates for the position and Euler parameters for the orientation (7×6 matrix for a 6 DOF robot).

Defined from the differentiation of $\mathbf{x} = \mathbf{f}(\mathbf{q})$ with respect to $\mathbf{q}$, the Jacobian is dependent on the representation $\mathbf{x}$ of the end-effector position and orientation. Since the kinematic properties of a mechanism are independent of the selected representation, it is important for the kinematic model to also be representation-independent. The Jacobian associated with such a model is unique. This Jacobian will be called the basic Jacobian.

The basic Jacobian matrix establishes the relationships between joint velocities and the corresponding (uniquely-defined) linear and angular velocities at a given point on the end-effector.

$$\begin{pmatrix} \mathbf{v} \\ \omega \end{pmatrix}_{(6\times 1)} = J_0(q)_{(6\times n)}\dot{\mathbf{q}}_{(n\times 1)} \tag{3.25}$$

Linear velocities are the time derivatives of the Cartesian coordinates of the end-effector position vector. However this is not the case for any orientation representation. For example if we take (α, β, γ) Euler angles, their derivatives are not the angular velocities. In fact angular velocities do not have a primitive function, no representation of the orientation has derivatives equal to the angular velocities. The angular velocity is defined as an instantaneous quantity. However, the time derivative of any representation of the orientation is related to the angular velocity. This is also the case for general position representation. These relationships are of the form

$$\dot{\mathbf{x}}_p = E_p(\mathbf{x}_p)\mathbf{v} \tag{3.26}$$
$$\dot{\mathbf{x}}_r = E_r(\mathbf{x}_r)\omega \tag{3.27}$$

Here $\dot{x_p}$ is the time derivative of the position part of the end-effector representation and $\dot{x_r}$ is the time derivative of the orientation part. The matrices E_p and E_r are only dependent on the particular position or orientation representation of the end-effector. Using E_p and E_r we will be able to obtain the Jacobian for the particular representation as a function of the basic Jacobian.

3.3.1 Example: E_p, E_r

As an illustration, if for example we use Cartesian coordinates for the end-effector position and $\alpha - \beta - \gamma$ Euler angles for the end-effector orientation

$$\mathbf{x}_p = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{3.28}$$

$$\mathbf{x}_r = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \tag{3.29}$$

the corresponding matrices for E_p and E_r are:

$$E_p(\mathbf{x}_p) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.30}$$

$$E_r(\mathbf{x}_r) = \begin{pmatrix} -\frac{s\alpha . c\beta}{s\beta} & \frac{c\alpha . c\beta}{s\beta} & 1 \\ c\alpha & s\alpha & 0 \\ \frac{s\alpha}{s\beta} & -\frac{c\alpha}{s\beta} & 0 \end{pmatrix} \tag{3.31}$$

As mentioned earlier E_p is the unit 3×3 matrix for that example.

3.3.2 Relationship: J_x and J_O

The basic Jacobian, J_0, is defined as

$$\begin{pmatrix} \mathbf{v} \\ \omega \end{pmatrix} = J_0(\mathbf{q})\dot{\mathbf{q}} \tag{3.32}$$

We will denote J_v and J_ω as the linear and angular velocity parts of this matrix.

$$\begin{cases} \mathbf{v} = J_v \dot{\mathbf{q}} \\ \omega = J_\omega \dot{\mathbf{q}} \end{cases} \tag{3.33}$$

Using the definitions of E_p and E_r above

$$\dot{\mathbf{x}}_p = E_p \mathbf{v} \Rightarrow \dot{\mathbf{x}}_p = (E_p J_v)\dot{\mathbf{q}} \tag{3.34}$$

and

$$\dot{\mathbf{x}}_r = E_r \omega \Rightarrow \dot{\mathbf{x}}_r = (E_r J_\omega)\dot{\mathbf{q}} \tag{3.35}$$

we can derive the following relationships between J_p and J_r and the basic Jacobian's components J_v and J_w.

$$\begin{cases} J_{X_P} = E_P J_v \\ J_{X_R} = E_R J_\omega \end{cases} \tag{3.36}$$

The above relationships can also be arranged in a matrix form by introducing the matrix $E_{(6\times 6)}$

$$J_x = \begin{pmatrix} J_p \\ J_r \end{pmatrix} = \begin{pmatrix} E_p & 0 \\ 0 & E_r \end{pmatrix} \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \tag{3.37}$$

Using E, the relationship between J_x and the basic Jacobian J_0 becomes

$$J_x(\mathbf{q}) = E(\mathbf{x}) J_0(\mathbf{q}) \tag{3.38}$$

with

$$\begin{pmatrix} \mathbf{v} \\ \omega \end{pmatrix} = J_0(q)\dot{\mathbf{q}} \tag{3.39}$$

For the example above

$$E_p = I_3; J_p = J_v \tag{3.40}$$

and

$$E = \begin{pmatrix} I & 0 \\ 0 & E_r \end{pmatrix} \tag{3.41}$$

3.4 Linear/Angular Motion

In this section we further analyze the linear and angular velocities associated with multi-body systems. Let us consider a point P described by a position vector $\mathbf{p}$ with respect to the origin of a fixed frame {A}. If the point P is moving with respect to frame {A}, the linear velocity of the point P with respect to frame {A} is the vector $\mathbf{v}_{P/A}$. As a vector, the linear velocity can be expressed in any frame - $\{A\}, \{B\}, \{C\}$ with the coordinates ${}^A\mathbf{v}_{P/A}$, ${}^B\mathbf{v}_{P/A}$, ${}^C\mathbf{v}_{P/A}$. The relationships between these coordinates, involve the rotation transformation matrices introduced earlier. Naturally if the point P is fixed in frame $\{A\}$, the linear velocity vector of P with respect to $\{A\}$ will be zero.

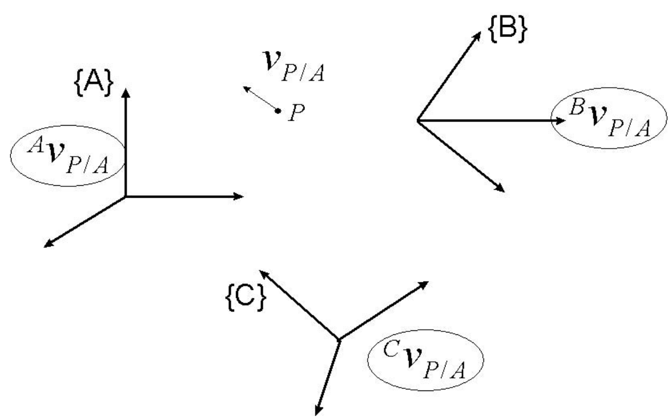

Figure 3.3: Linear Velocity

3.4.1 Pure Translation

Let us now consider a pure translation of frame {A} with respect to another frame {B}. The linear velocity of point P with respect to {B} is $\mathbf{v}_{P/B}$. If $\mathbf{v}_{A/B}$ represents the velocity of the origin of frame {A} with respect to frame {B}, the two vectors of linear velocities of point P with respect to {A} and {B} are related by

$$v_{P/B} = v_{A/B} + v_{P/A}$$

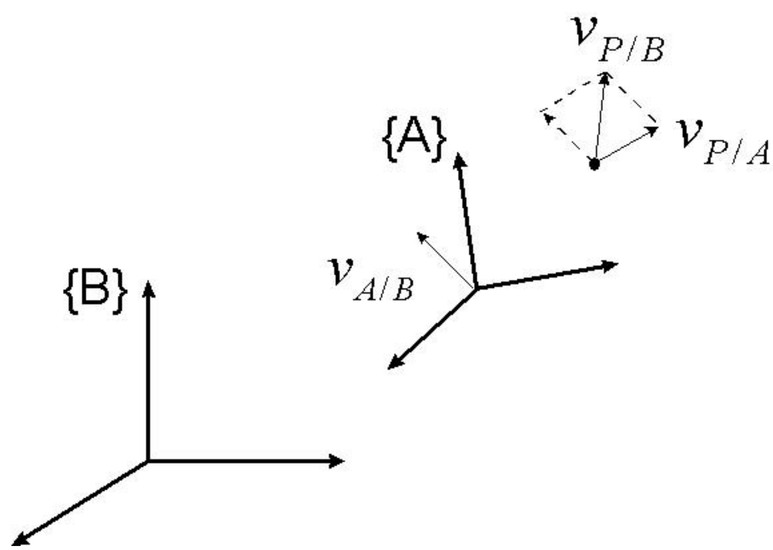

Figure 3.4: Pure Translation

3.4.2 Pure Rotation

To analyze the rotation of a rigid body, we need to define a point fixed in the body and an axis of rotation passing through this point. The body rotates about this axis and all the points along this axis are fixed *w.r.t.* this rotation. This rotation is described by a quantity called angular velocity, represented by the vector Ω.

A point P on the rotating rigid body is moving with a linear velocity $\mathbf{v}_P$, which is dependent on the magnitude of Ω and on the location of P with respect to the axis of rotation.

Different points on the rigid body will have different linear velocities. If we select a point O in the body along the axis of rotation the position vector $\mathbf{p}$ measured from O to P will be perpendicular to the linear

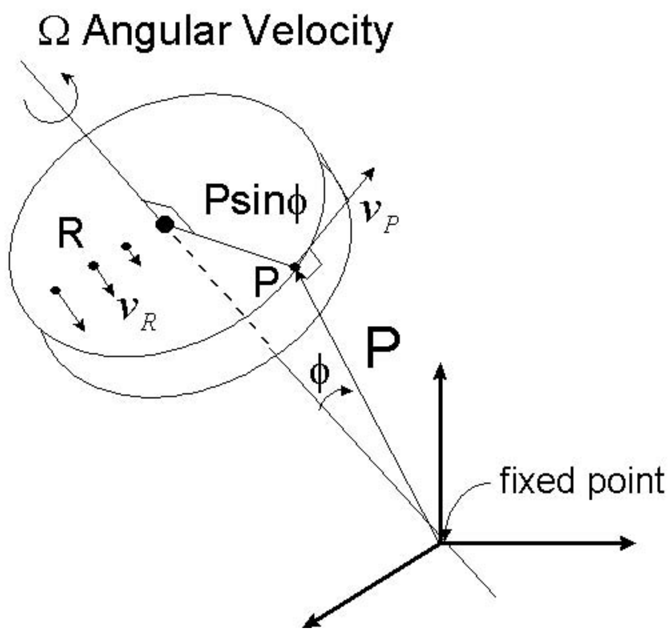

Figure 3.5: Rotational Motion

velocity vector $\mathbf{v}_p$. In addition from mechanics we know that the vector $\mathbf{v}_p$ is also perpendicular to the axis of rotation and in particular to Ω (the angular velocity vector). The magnitude of $\mathbf{v}_p$ is proportional to the magnitude of Ω (the rate of rotation) and to the distance to the axis of rotation, in other words to the magnitude of $\mathbf{p}\sin(\phi)$, as illustrated in Figure 3.5. Here ϕ is the angle between the axis of rotation and the position vector $\mathbf{p}$. Thus we can derive the following relationship

$$\mathbf{v}_P = \Omega \times \mathbf{p} \tag{3.42}$$

Using the definition of cross product operator, the above vector relationship can be described in the matrix form as

$$\mathbf{v}_P = \Omega \times \mathbf{p} \Rightarrow \mathbf{v}_P = \widehat{\Omega}\mathbf{p} \tag{3.43}$$

For instance. let us consider the components of vectors, Ω and $\mathbf{p}$.

$$\Omega = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \quad \text{and} \quad \mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{3.44}$$

With the cross product operator, the linear velocity of a point P is

$$\mathbf{v}_P = \widehat{\Omega}\mathbf{p} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y \\ \Omega_z & 0 & -\Omega_x \\ -\Omega_y & \Omega_x & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \tag{3.45}$$

3.4.3 Cross Product Operator and Rotation Matrix

Consider the rotation matrix between a frame fixed with respect to the rigid axis and frame moving with the rotated body. The cross product operator $\widehat{\Omega}$ can be expressed in terms of this rotation matrix.

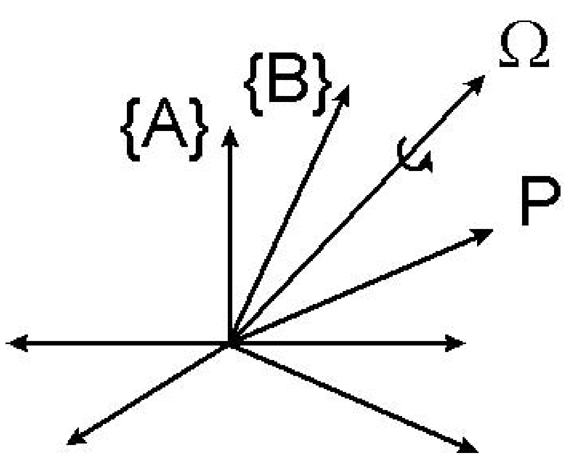

Figure 3.6: Rotation and Cross Product

Consider a pure rotation about an axis with an angular velocity Ω. Let P be a point fixed in body B. Then the velocity of P in B is zero, i.e.

$$\mathbf{v}_{P/B} = 0 \tag{3.46}$$

The representations ${}^B\mathbf{p}$ and ${}^A\mathbf{p}$ of the position vector $\mathbf{p}$ in frames $\{A\}$ and $\{B\}$ are related by the rotation matrix ${}^A_BR$

$${}^A\mathbf{p} = {}^A_BR\,{}^B\mathbf{p} \tag{3.47}$$

Let us differentiate *w.r.t.* time the above relationship

$${}^A\dot{\mathbf{p}} = {}^A_B\dot{R}\,{}^B\mathbf{p} + {}^A_BR\,{}^B\dot{\mathbf{p}}$$

Noting that the second term is equal to zero (since $\mathbf{v}_{P/B} = 0$), the relationship becomes

$${}^A\dot{\mathbf{p}} = {}^A_B\dot{R}\,{}^B\mathbf{p}$$

Transforming ${}^B\mathbf{p}$ to ${}^A\mathbf{p}$ by premultiplication of ${}^A_BR^T\,{}^A_BR = I$, yields

$$\begin{aligned} {}^A\dot{\mathbf{p}} &= {}^A_B\dot{R}\,(I)\,{}^B\mathbf{p} = {}^A_B\dot{R}\,({}^A_BR^T\,{}^A_BR)\,{}^B\mathbf{p} \qquad (3.48) \\ {}^A\dot{\mathbf{p}} &= {}^A_B\dot{R}\,{}^A_BR^T({}^A_BR\,{}^B\mathbf{p}) = ({}^A_B\dot{R}\,{}^A_BR^T)\,{}^A\mathbf{p} \qquad (3.49) \end{aligned}$$

The above relationship can be written in vector form for any rotating frame

$$\dot{\mathbf{p}} = \dot{R}R^T\mathbf{p} \tag{3.50}$$

Observing that $\dot{\mathbf{p}}$ is linear velocity of $\mathbf{v}_P$, we obtain

$$\widehat{\Omega} = \dot{R}R^T \tag{3.51}$$

3.4.4 Example: Rotation about axis Z

Consider the rotation of frame about the axis Z of a fixed frame. Measured by the angle θ, the corresponding rotation matrix is

$$R = \begin{pmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.52}$$

The derivative *w.r.t.* time is

$$\dot{R} = \begin{pmatrix} -s\theta\dot{\theta} & -c\theta\dot{\theta} & 0 \\ c\theta\dot{\theta} & -s\theta\dot{\theta} & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.53}$$

or

$$\dot{R}.R^T = \begin{pmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.54}$$

Clearly vector ω here is just

$$\Omega = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta} \end{pmatrix} \tag{3.55}$$

and we can verify that

$$\widehat{\Omega} = \begin{pmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.56}$$

Thus the relationship above is verified.

$$\widehat{\Omega} = \dot{R}R^T \tag{3.57}$$

3.5 Combined Linear and Angular Motion

Now we consider motions involving both linear and angular velocities, as illustrated in Figure 3.7.

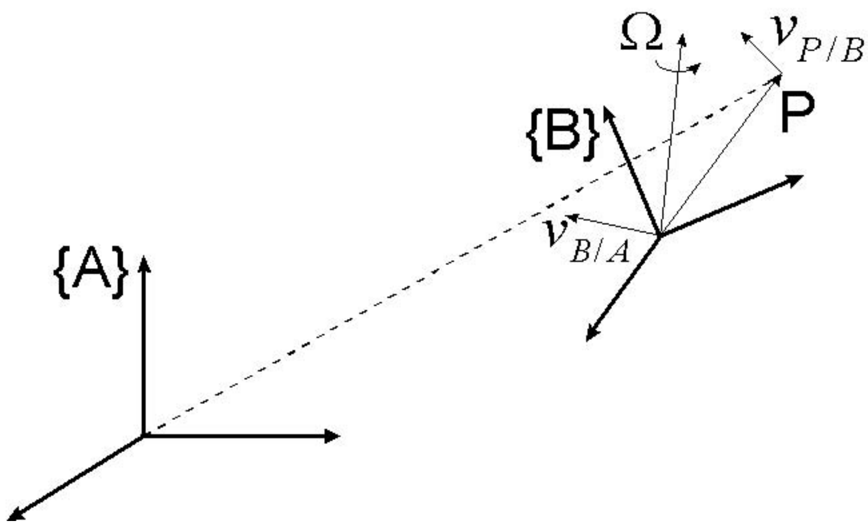

Figure 3.7: Linear and Angular Motion

The corresponding relationship is:

$$\mathbf{v}_{P/A} = \mathbf{v}_{B/A} + \mathbf{v}_{P/B} + \Omega \times \mathbf{p}_B \tag{3.58}$$

In order to perform this addition we need to have all quantities expressed in the same reference frame. In frame {A} the equation is

$${}^A\mathbf{v}_{P/A} = {}^A\mathbf{v}_{B/A} + {}^A_BR^B\mathbf{v}_{P/B} + {}^A\Omega_B \times {}^A_BR^B\mathbf{p}_B \tag{3.59}$$

3.6 Jacobian: Velocity Propagation

When we have several rigid bodies connected in a mechanism, we need to propagate the velocities from frame {0} to frame $\{n\}$.

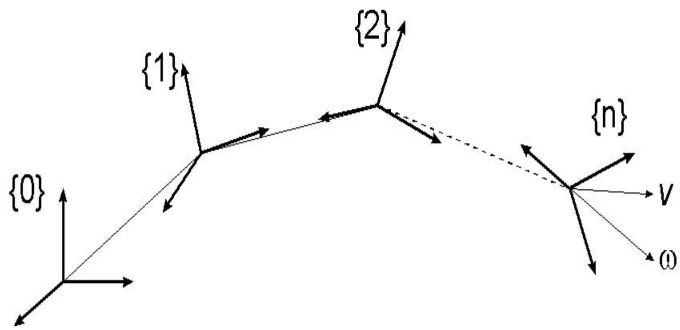

Figure 3.8: A Spatial Mechanism

The linear and angular velocity at the end-effector can be computed by propagation of velocities through the links of the manipulator. By computing and propagating linear and angular velocities from the fixed base to the end-effector, we establish the relationship between joint velocities and end- effector velocities. This provides an iterative method to compute the Basic Jacobian.

Consider two consecutive links i and $i+1$.

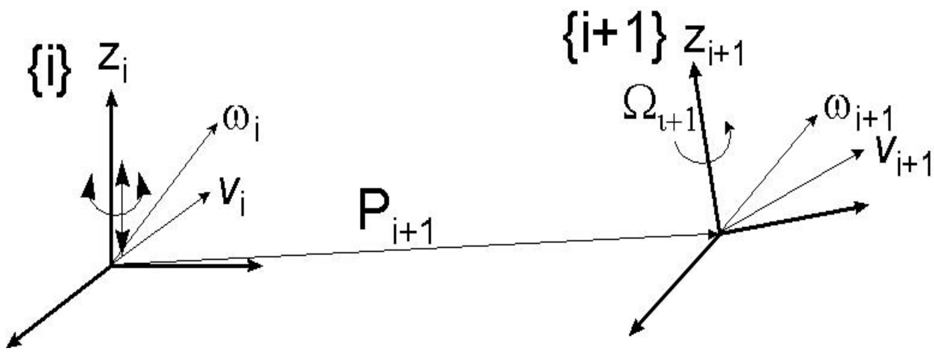

Figure 3.9: Velocity Propagation

The angular velocity of link $i+1$ is equal to the angular velocity of link i plus the local rotation of link $i+1$ represented by Ω_{i+1}.

$$\omega_{i+1} = \omega_i + \Omega_{i+1} \tag{3.60}$$

This local rotation is simply given by the derivative $\dot{\theta}_{i+1}$ of the angle of rotation of the link along the axis of rotation $\mathbf{z}_{i+1}$.

$$\Omega_{i+1} = \dot{\theta}_{i+1}\mathbf{z}_{i+1} \tag{3.61}$$

For the linear velocity the expression is slightly more complicated. The linear velocity at link $i+1$ is equal to the one at link i plus the contribution of the angular velocity of link i ($\omega_i \times \mathbf{p}_{i+1}$) plus the contribution of the local linear velocity associated with a prismatic joint (this is $\dot{d}_{i+l}\mathbf{z}_{i+1}$) if joint $i+1$ was prismatic.

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \omega_i \times \mathbf{p}_{i+1} + \dot{d}_{i+1}\mathbf{z}_{i+1} \tag{3.62}$$

If we use these equations we can propagate them from the beginning to the end of the chain. If the computation of velocities is done in the local frame, the result will be obtained in frame n. The end-effector linear and angular velocities in the base frame are

$$\begin{pmatrix} {}^0v \\ {}^0\omega \end{pmatrix} = \begin{pmatrix} {}^0_nR & 0 \\ 0 & {}^0_nR \end{pmatrix} \begin{pmatrix} {}^nv \\ {}^n\omega \end{pmatrix} \tag{3.63}$$

The above expressions are linear functions of $\dot{\mathbf{q}}$, from which the basic Jacobian can be extracted. This iterative procedure is suitable for numerical computations of the Jacobian. The procedure, however, does not provide a description of the special structure of the Jacobian matrix. The next section addresses this aspect and presents a method for an explicit form of the Jacobian.

3.7 Jacobian: Explicit Form

Consider a general mechanism and let us examine how the velocities at the joints affect the linear and angular velocities at the end effector.

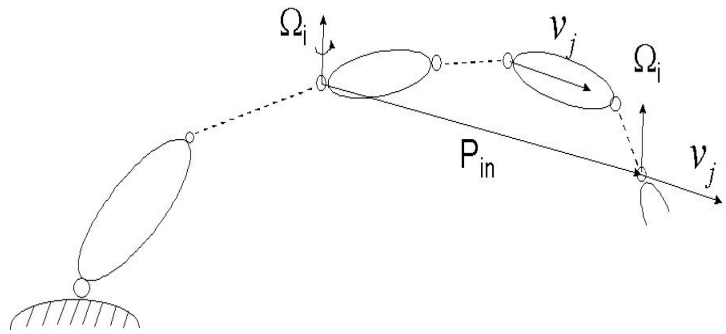

Figure 3.10: Explicit Form of the Jacobian

The velocity of a link with respect to the proceeding link is dependent on the type of joint that connects them. If the joint is a prismatic one, then the link linear velocity with respect to the previous link is along the prismatic joint axis, $\mathbf{z}_i$ with a magnitude of $\dot{q}_i$.

$$\mathbf{v}_i = \mathbf{z}_i \dot{q}_i \tag{3.64}$$

Similarly for a revolute joint the angular velocity is about the revolute joint axis with a magnitude of $\dot{q}_i$.

$$\Omega_i = \mathbf{z}_i \dot{q}_i \tag{3.65}$$

The local velocity at each joint contributes to the end effector velocities. A revolute joint creates both an angular rotation at the end-effector and a linear velocity. The linear velocity depends on the distance between the end-effector point and the joint axis. It involves the cross product of Ω_i with the vector locating this point. The angular velocity, Ω_i is transfered down the chain to the end-effector. A prismatic joint j creates only a linear velocity $\mathbf{v}_j$ that gets transfered down to the end-effector.

The total contribution of joint velocities of the mechanism to the end effector linear velocity is therefore

$$\mathbf{v} = \sum_{i=1}^{n} [\epsilon_i v_i + \bar{\epsilon}_i(\Omega_i \times \mathbf{p}_{in})] \tag{3.66}$$

Similarly the end effector angular velocity is the sum

$$\omega = \sum_{i=1}^{n} \bar{\epsilon}_i \Omega_i \tag{3.67}$$

Substituting the expressions of $\mathbf{v}_i$ and Ω_i from equations 3.64 and 3.65, we obtain

$$\mathbf{v} = \sum_{i=1}^{n} [\epsilon_i \mathbf{z}_i + \bar{\epsilon}_i(\mathbf{z}_i \times \mathbf{p}_{in})]\dot{q}_i \tag{3.68}$$

$$\omega = \sum_{i=1}^{n} \bar{\epsilon}_i \mathbf{z}_i \dot{q}_i \tag{3.69}$$

The end-effector velocity is:

$$\mathbf{v} = (\epsilon_1 \mathbf{z}_1 + \bar{\epsilon}_1(\mathbf{z}_1 \times \mathbf{p}_{1n}))\dot{q}_1 + (\epsilon_2 \mathbf{z}_2 + \bar{\epsilon}_2(\mathbf{z}_2 \times \mathbf{p}_{2n}))\dot{q}_2 + \cdots \tag{3.70}$$

or

$$\mathbf{v} = \left[\, (\epsilon_1 \mathbf{z}_1 + \bar{\epsilon}_1(\mathbf{z}_1 \times \mathbf{p}_{1n})) \quad (\epsilon_2 \mathbf{z}_2 + \bar{\epsilon}_2(\mathbf{z}_2 \times \mathbf{p}_{2n})) \quad \cdots \right] . \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \tag{3.71}$$

and it can be written as:

$$\mathbf{v} = J_v \dot{\mathbf{q}} \tag{3.72}$$

where J_v is the linear motion Jacobian. Similarly the end-effector angular velocity is:

$$\omega = \bar{\epsilon}_1 \mathbf{z}_1 \dot{q}_1 + \bar{\epsilon}_2 \mathbf{z}_2 \dot{q}_2 + \cdots + \bar{\epsilon}_n \mathbf{z}_n \dot{q}_n \tag{3.73}$$

or

$$\omega = \begin{bmatrix} \bar{\epsilon}_1 \mathbf{z}_1 & \bar{\epsilon}_2 \mathbf{z}_2 & \cdots & \bar{\epsilon}_n \mathbf{z}_n \end{bmatrix} . \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \tag{3.74}$$

and it can be written as:

$$\omega = J_\omega \dot{\mathbf{q}} \tag{3.75}$$

where J_ω is the angular motion Jacobian. Combining the linear and angular motion parts leads to the basic Jacobian

$$\left\{ \begin{matrix} \mathbf{v} = J_v \dot{q} \\ \omega = J_\omega \dot{\mathbf{q}} \end{matrix} \right\} \rightarrow \begin{pmatrix} \mathbf{v} \\ \omega \end{pmatrix} = J \dot{\mathbf{q}} \tag{3.76}$$

or

$$J = \begin{pmatrix} J_v \\ J_\omega \end{pmatrix} \tag{3.77}$$

The equations provide the expressions for the matrices J_v and J_w. The derivation of the matrix J_v involves new quantities $\mathbf{p}_{1n}, \mathbf{p}_{2n}, \ldots, \mathbf{p}_{nn}$ that need to be computed.

A simple approach to compute J_v is to use the direct differentiation of the Cartesian coordinates of the point on the end-effector

$$\mathbf{v} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \dot{\mathbf{x}}_P = \frac{\partial x_P}{\partial q_1} \dot{q}_1 + \frac{\partial x_P}{\partial q_2} \dot{q}_2 + \cdots + \frac{\partial x_P}{\partial q_n} \dot{q}_n \tag{3.78}$$

$$\mathbf{v} = \begin{pmatrix} \frac{\partial x_P}{\partial q_1} & \frac{\partial x_P}{\partial q_2} & \cdots & \frac{\partial x_P}{\partial q_n} \end{pmatrix} \dot{\mathbf{q}} = J_v \dot{\mathbf{q}} \tag{3.79}$$

For J_w all that we need is to compute the $\mathbf{z}-$vectors associated with revolute joints. Overall the Jacobian takes the form

$$J = \begin{pmatrix} \frac{\partial x_P}{\partial q_1} & \frac{\partial x_P}{\partial q_2} & \cdots & \frac{\partial x_P}{\partial q_n} \\ \bar{\epsilon}_1 \mathbf{z}_1 & \bar{\epsilon}_2 \mathbf{z}_2 & \cdots & \bar{\epsilon}_n \mathbf{z}_n \end{pmatrix} \tag{3.80}$$

Note that ϵ is zero for a revolute joint and one for a prismatic one. To express the Jacobian in particular frame, all we need is to have all the quantities expressed in that frame.

$${}^0J = \begin{pmatrix} \frac{\partial {}^0x_P}{\partial q_1} & \frac{\partial {}^0x_P}{\partial q_2} & \cdots & \frac{\partial {}^0x_P}{\partial q_n} \\ \bar{\epsilon}_1 {}^0\mathbf{z}_1 & \bar{\epsilon}_2 {}^0\mathbf{z}_2 & \cdots & \bar{\epsilon}_n {}^0\mathbf{z}_n \end{pmatrix} \tag{3.81}$$

The components of ${}^0\mathbf{z}_i$ can be found as ${}^0\mathbf{z}_i = {}^0_iR\,{}^i\mathbf{z}_i$ (${}^i\mathbf{z}_i$ is of course $(\,0 \quad 0 \quad 1\,)$). Thus all we need for the angular motion Jacobian is the last column of the rotation matrix.

$$\begin{pmatrix} X \\ X \\ X \end{pmatrix} = \begin{pmatrix} & & X \\ & & X \\ & & X \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.82}$$

$${}^0\mathbf{z}_i = {}^0_iR\mathbf{z} \tag{3.83}$$

with

$$\mathbf{z} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.84}$$

The overall Jacobian is then found as:

$${}^0J = \begin{pmatrix} \frac{\partial}{\partial q_1}({}^0x_P) & \frac{\partial}{\partial q_2}({}^0x_P) & \cdots & \frac{\partial}{\partial q_n}({}^0x_P) \\ \bar{\epsilon}_1({}^0_1R\mathbf{z}) & \bar{\epsilon}_2({}^0_2R\mathbf{z}) & \cdots & \bar{\epsilon}_n({}^0_nR\mathbf{z}) \end{pmatrix} \tag{3.85}$$

3.7.1 Example: Stanford Scheinman Arm

As we have shown previously, we first introduce frames, define the D&H parameters and calculate the D&H table. Then we calculate the transformation matrices, namely:

$$ {}^{0}_{1}T = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.86} $$

$$ {}^{1}_{2}T = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ -s_2 & c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.87} $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.88} $$

$$ {}^{3}_{4}T = \begin{bmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.89} $$

$$ {}^{4}_{5}T = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.90} $$

$$ {}^{5}_{6}T = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.91} $$

Next we express each of the frames *w.r.t.* the $\{0\}$ frame, i.e. we calculate the transformation matrices:

$$
{}^0_2T = \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & -s_1d_2 \\ s_1c_2 & -s_1s_2 & c_1 & c_1d_2 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.92}
$$

$$
{}^0_3T = \begin{bmatrix} c_1c_2 & -s_1 & c_1s_2 & c_1d_3s_2 - s_1d_2 \\ s_1c_2 & c_1 & s_1s_2 & s_1d_3s_2 + c_1d_2 \\ -s_2 & 0 & c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.93}
$$

$$
{}^0_4T = \begin{bmatrix} c_1c_2c_4 - s_1s_4 & -c_1c_2s_4 - s_1c_4 & c_1s_2 & c_1d_3s_2 - s_1d_2 \\ s_1c_2c_4 + c_1s_4 & -s_1c_2s_4 + c_1c_4 & s_1s_2 & s_1d_3s_2 + c_1d_2 \\ -s_2s_4 & s_2c_4 & c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.94}
$$

$$
{}^0_5T = \begin{bmatrix} X & X & -c_1c_2s_4 - s_1c_4 & c_1d_3s_2 - s_1d_2 \\ X & X & -s_1c_2s_4 + c_1c_4 & s_1d_3s_2 + c_1d_2 \\ X & X & s_2s_4 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.95}
$$

$$
{}^0_6T = \begin{bmatrix} X & X & c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2s_5 & c_1d_3s_2 - s_1d_2 \\ X & X & s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2s_5 & s_1d_3s_2 + c_1d_2 \\ X & X & -s_2c_4s_5 + c_5c_2 & d_3c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.96}
$$

As we can see the origin of frame $\{3\}$ is the same as the one for $\{4\}, \{5\}$ and $\{6\}$. All we need to keep for the computation of the orientation is just the third column of the transformation matrices. Now we can fill the 6×6 Jacobian in this case using the information above.

$$
J = \begin{pmatrix} \frac{\partial x_P}{\partial q_1} & \frac{\partial x_P}{\partial q_2} & \frac{\partial x_P}{\partial q_3} & 0 & 0 & 0 \\ {}^0\mathbf{z}_1 & {}^0\mathbf{z}_2 & 0 & {}^0\mathbf{z}_4 & {}^0\mathbf{z}_5 & {}^0\mathbf{z}_6 \end{pmatrix} \tag{3.97}
$$

As we can see the 3×1 representation of the third orientation vector is 0 (since it is a prismatic link). Similarly, we easily fill the rest of the matrix as a function of ${}^0\mathbf{z}_1, {}^0\mathbf{z}_2, \ldots, {}^0\mathbf{z}_6$ and $\frac{\partial x_P}{\partial q_1}, \frac{\partial x_P}{\partial q_2}, \frac{\partial x_P}{\partial q_3}$. These expressions can be easily calculated using the transforms we calculated before. The Jacobian, J, is

$$\begin{bmatrix} -c_1 d_2 - s_1 s_2 d_3 & c_1 c_2 d_3 & c_1 s_2 & 0 & 0 & 0 \\ -s_1 d_2 + c_1 s_2 d_3 & s_1 c_2 d_3 & s_1 s_2 & 0 & 0 & 0 \\ 0 & -s_2 d_3 & c_2 & 0 & 0 & 0 \\ 0 & -s_1 & 0 & c_1 s_2 & -c_1 c_2 s_4 - s_1 c_4 & c_1 c_2 c_4 s_5 - s_1 s_4 s_5 + c_1 s_2 s_5 \\ 0 & c_1 & 0 & s_1 s_2 & -s_1 c_2 s_4 + c_1 c_4 & s_1 c_2 c_4 s_5 + c_1 s_4 s_5 + s_1 s_2 s_5 \\ 1 & 0 & 0 & c_2 & s_2 s_4 & -s_2 c_4 s_5 + c_5 c_2 \end{bmatrix}$$

Of course all quantities in this matrix are expressed in frame 0. Note that the horizontal dimension of the basic Jacobian depends on the number of DOF of the mechanism, while the vertical one is six (3 for the position and 3 for the orientation).

3.7.2 Jacobian in a Different Frame

As we mentioned above, we may want to express the Jacobian in different frames. The transformation matrix between two frames is

$${}^B J = \begin{pmatrix} {}^B_A R & 0 \\ 0 & {}^B_A R \end{pmatrix} {}^A J \tag{3.98}$$

In practice the best frame to compute the Jacobian is in the middle of the chain because that makes the expressions of the elements of the Jacobian least complicated. Moving to frame $\{0\}$ can be done using the above transformation.

3.8 Kinematic Singularities

The work space of a manipulator generally contains a number of particular configurations that locally limits the end-effector mobility. Such configurations are called singular configurations. At a singular configuration, the end-effector locally loses the ability to move along or rotate about some direction in Cartesian space.

Note that these singularities are related to the kinematics of the manipulator and are obviously different from the singularities of the representation we have discussed earlier, which uniquely arise from the type of the selected representation.

For example the kinematic singularities for a 2 DOF revolute arm (see Figure 3.11) are the configurations where the two links are collinear. The end-effector cannot move along the common link direction.

Another example of singularity is the wrist singularity, which is common for the Stanford Scheinman Arm and the PUMA. This is the configuration when axis 4 and axis 6 are collinear, the end effector cannot rotate about the normal to the plane defined by axes 4 and 5.

In such configurations, instantaneously the end effector cannot rotate about that axis. In other words even though we can vary the joint velocities, the resulting linear or angular velocity at the end effector will be zero. Since the Jacobian is the mapping between these velocities, the analysis of kinematic singularities is closely connected to the Jacobian.

At a singular configuration, some columns of the Jacobian matrix become linearly dependent and the Jacobian losses rank. The phenomenon of singularity can then be studied by checking the determinant of the Jacobian, which is zero at singular configurations.

$$\det[J(\mathbf{q})] = 0 \tag{3.99}$$

Consider again the example of the 2 DOF revolute manipulator illustrated in Figure 3.11.

From simple geometric considerations we derive the coordinates of the end-effector point.

$$x = l_1 c_1 + l_2 c_{12} \tag{3.100}$$

$$y = l_1 s_1 + l_2 s_{12} \tag{3.101}$$

This leads to the Jacobian

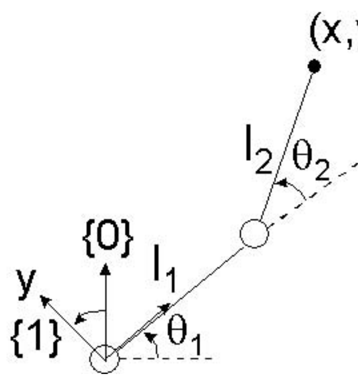

Figure 3.11: 2 DOF Example

$$
{}^0J = \begin{pmatrix} -y & -l_2 s_{12} \\ x & l_2 c_{12} \end{pmatrix} \tag{3.102}
$$

We now express the Jacobian in frame $\{1\}$ to further simplify its expression.

$$
{}^1J = {}^1_0R\,{}^0J
$$

with

$$
{}^0_1R = \begin{pmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{pmatrix} \tag{3.103}
$$

Thus:

$$
{}^1J = \begin{pmatrix} -l_2 s_2 & -l_2 s_2 \\ l_1 + l_2 c_2 & l_2 c_2 \end{pmatrix} \tag{3.104}
$$

The above expressions show how the manipulator approaches a singularity as the angle θ_2 goes to zero. When $s_2 = 0$ the first row becomes $\begin{pmatrix} 0 & 0 \end{pmatrix}$ and the rank of the Jacobian is 1.

$$
{}^1J = \begin{pmatrix} 0 & 0 \\ l_1 + l_2 & l_2 \end{pmatrix} \tag{3.105}
$$

Note that the determinant of the Jacobian does not depend on the frame where the matrix is defined.

$$\det[{}^B J] = \det \begin{bmatrix} {}^B_A R & 0 \\ 0 & {}^B_A R \end{bmatrix} \det[{}^A J] \tag{3.106}$$

Consider a small joint displacement $(\delta\theta_1, \delta\theta_2)$ from the singular configuration. The corresponding end effector displacement $(\delta x, \delta y)$ expressed in frame 1 is

$$\begin{pmatrix} {}^1\delta x \\ {}^1\delta y \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ l_1 + l_2 & l_2 \end{pmatrix} \begin{pmatrix} \delta\theta_1 \\ \delta\theta_2 \end{pmatrix} \tag{3.107}$$

Thus:

$${}^1\delta x = 0 \tag{3.108}$$

and

$${}^1\delta y = (l_1 + l_2)\delta\theta_1 + l_2\delta\theta_2 \tag{3.109}$$

3.9 Jacobian at Wrist/End-Effector

The point at the end effector, where linear and angular velocities are evaluated, varies with the robot's task, grasped object, or tools. Each selection of the end-effector point corresponds to a different Jacobian. The simplest Jacobian corresponds to the wrist point. The wrist point is fixed with respect to the end effector and the Jacobian for any other point can be computed from the wrist Jacobian.

Consider a point E at the end effector located with respect to the wrist point (origin of frame $\{n\}$) by a vector $\mathbf{p}_{we}$. The linear velocity, $\mathbf{v}_e$ at point E is

$$\mathbf{v}_e = \mathbf{v}_n + \omega_n \times \mathbf{p}_{ne} \tag{3.110}$$

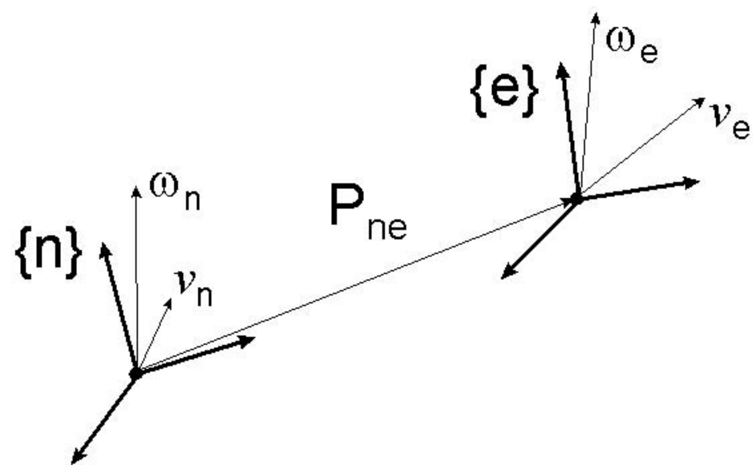

Figure 3.12: Jacobian at the End-effector

Since the angular velocity is the same at both points (E being fixed with respect to W), we have

$$\begin{cases} \mathbf{v}_e = \mathbf{v}_n - \mathbf{p}_{ne} \times \omega_n \\ \omega_e = \omega_n \end{cases} \tag{3.111}$$

Replacing $\mathbf{p}_{ne}\times$ by the cross product operator $\widehat{\mathbf{p}}_{ne}$ yields

$$^0J_e = \begin{pmatrix} I & -^0\widehat{\mathbf{p}}_{ne} \\ 0 & I \end{pmatrix} {^0J_n} \tag{3.112}$$

3.9.1 Example: 3 DOF RRR Arm

Let us consider the 3 DOF revolute planar mechanism shown in Figure 3.13.

The position coordinates of the end-effector wrist point are

$$x_W = l_1 c_1 + l_2 c_{12} \tag{3.113}$$
$$y_W = l_2 s_1 + l_2 s_{12} \tag{3.114}$$

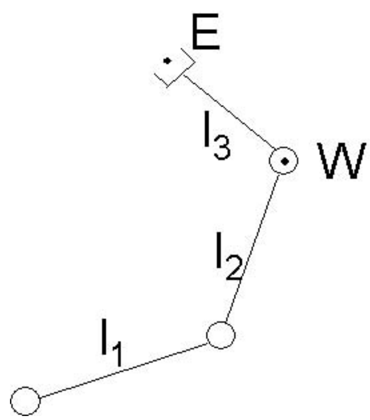

Figure 3.13: 3 DOF Example

at the end-effector point is:

$$x_E = l_1 c_1 + l_2 c_{12} + l_3 c_{123} \tag{3.115}$$

$$y_E = l_2 s_1 + l_2 s_{12} + l_3 s_{123} \tag{3.116}$$

The Jacobian in frame $\{0\}$ for the wrist point is

$$J_W = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.117}$$

To get the Jacobian at the end-effector we will use vector $\mathbf{p}_{WE}$ in frame $\{0\}$:

$${}^0J_E = \begin{pmatrix} I & -{}^0\widehat{\mathbf{p}}_{WE} \\ 0 & I \end{pmatrix} {}^0J_W \tag{3.118}$$

Thus the cross product operator is

$$
{}^0\mathbf{p}_{WE} = \begin{bmatrix} l_3c_{123} \\ l_3s_{123} \\ 0 \end{bmatrix} \Rightarrow {}^0\hat{\mathbf{p}}_{WE} = \begin{pmatrix} 0 & 0 & l_3s_{123} \\ 0 & 0 & -l_3c_{123} \\ -l_3s_{123} & l_3c_{123} & 0 \end{pmatrix} \tag{3.119}
$$

The Jacobian at point E is

$$
{}^0J_E = \begin{pmatrix} I & -{}^0\hat{\mathbf{p}}_{WE} \\ 0 & I \end{pmatrix} \begin{pmatrix} {}^0J_{v(W)} \\ {}^0J_{\omega(W)} \end{pmatrix} = \begin{pmatrix} {}^0J_{v(W)} - {}^0\hat{\mathbf{p}}_{WE}{}^0J_{\omega(W)} \\ {}^0J_{\omega(W)} \end{pmatrix} \tag{3.120}
$$

Here:

$$
-{}^0\hat{\mathbf{p}}_{WE}{}^0J_{\omega(W)} = \begin{pmatrix} -l_3s_{123} & -l_3s_{123} & -l_3s_{123} \\ l_3c_{123} & l_3c_{123} & l_3c_{123} \\ 0 & 0 & 0 \end{pmatrix} \tag{3.121}
$$

If we perform the multiplications we obtain the following 3 columns for the Jacobian associated with the linear velocity and the overall Jacobian follows.

$$
{}^0J_E = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} & -l_2s_{12} - l_3s_{123} & -l_3s_{123} \\ l_1c_1 + l_2c_{12} + l_3c_{123} & l_2c_{12} + l_3c_{123} & l_3c_{123} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{3.122}
$$

We can verify these results using the time derivatives of (x, y, z) as before.

3.10 Static Forces

Another application of the Jacobian is to define the relationship between forces applied at the end-effector and torques needed at the joints to support these forces. We described the relationship between linear velocities and angular velocities at the end effector and the joint velocities. Here we consider the relationship between end-effector forces and moments as related to joint torques. We will denote by **f** and **n** the static force and moment applied by the end-effector to the environment. $\tau_1, \tau_2, \ldots, \tau_n$ are the torques needed at the joints of the manipulator to produce **f** and **n**.

3.10.1 Force Propagation

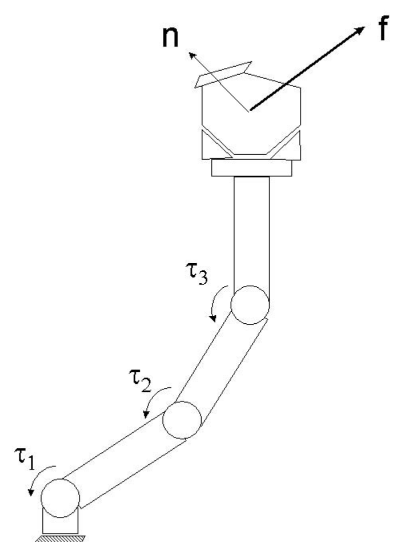

Figure 3.14: Static Forces

One way to establish this relationship is through propagation of the forces along the kinematic chain, similar to the velocity propagation from link to link. In fact as we will see later in considering the dynamics of the manipulator, velocities are propagated up the kinematic chain after which forces are propagated back in the opposite direction. As we propagate we can eliminate internal forces that are supported by the structure. This is done by projecting all forces at the joints. To analyze the static forces, let us imagine that we isolate the links of the manipulator into components, which can be treated as separate rigid bodies.

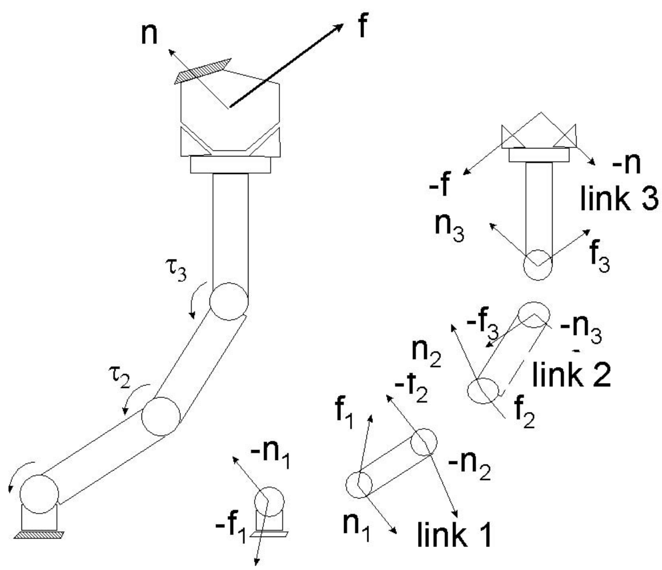

Figure 3.15: Force and Moments Cancelation

For each rigid body, we will consider all forces and moments that act on it and will then set the conditions for bring it to equilibrium. Let us consider the rigid body i (link i). In order for this rigid body to be at equilibrium, the sum of all forces and moments with respect to any point on the rigid body must be equal to zero. For link i, we have

$$\mathbf{f}_i + (-\mathbf{f}_{i+1}) = 0 \tag{3.123}$$

We select the origin of frame $\{i\}$ for the moment computation. This leads to the equation

$$\mathbf{n}_i + (-\mathbf{n}_{i+1}) + \mathbf{p}_{i+1} \times (-\mathbf{f}_{i+1}) = 0 \tag{3.124}$$

The above two relationships can be written recursively as follows:

$$\mathbf{f}_i = \mathbf{f}_{i+1} \tag{3.125}$$

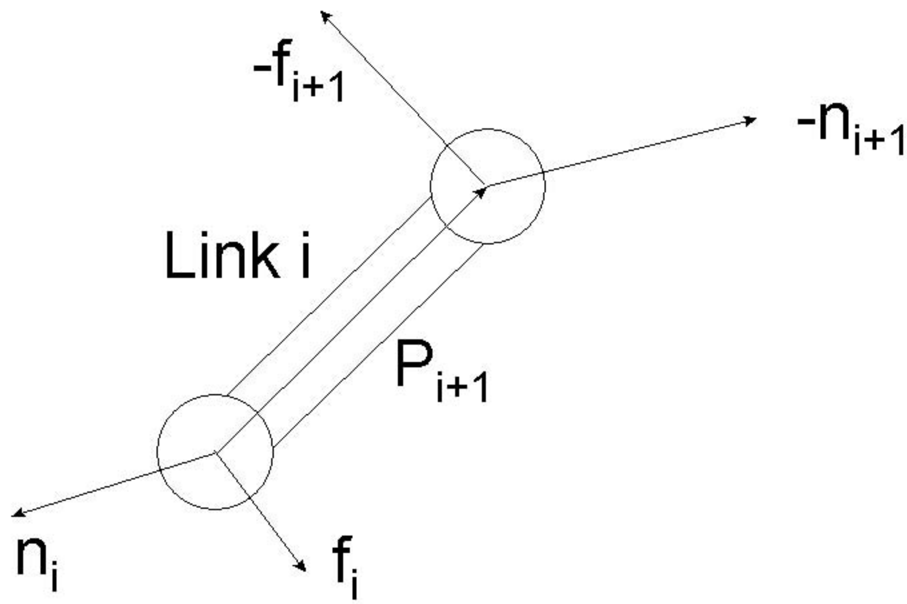

Figure 3.16: Link Equilibrium

$$\mathbf{n}_i = \mathbf{n}_{i+1} + \mathbf{p}_{i+1} \times \mathbf{f}_{i+1} \tag{3.126}$$

We need to guarantee that we eliminate in these equations the components that will be transmitted to the ground through the structure of the mechanism. We will do that by projecting the equations along the joint axis and propagating these relationships along the kinematic chain from the end-effector to the ground. These relationships are as follows: For a prismatic joint $\tau_i = \mathbf{f}_i^T \mathbf{z}_i$ and for a revolute joint $\tau_i = \mathbf{n}_i^T \mathbf{z}_i$.

For link $\{n\}$

$$^n\mathbf{f}_n = {}^n f \tag{3.127}$$

$$^n\mathbf{n}_n = {}^n n + {}^n\mathbf{p}_{n+1} \times {}^n f \tag{3.128}$$

and for link i

$$^{i}\mathbf{f}_i = {}^{i}_{i+1}R^{i+1}\mathbf{f}_{i+1} \tag{3.129}$$

$$^{i}\mathbf{n}_i = {}^{i}_{i+1}R^{i+1}\mathbf{n}_{i+1} + {}^{i}\mathbf{p}_{i+1} \times {}^{i}\mathbf{f}_i \tag{3.130}$$

This iterative process leads to a linear relationship between end-effector forces and moments and joint torques. The analysis of this relationship shows that it is simply the transpose of the Jacobian matrix.

$$\tau = J^T\mathbf{F}$$

where $\mathbf{F}$ is the vector combining end-effector forces and moments. The above relationship is the dual of the relationship we have established earlier between the end-effector linear angular velocities and joint velocities.

Earlier we derived similar equations for propagating angular and linear velocities along the links. These equations are

$$^{i+1}\omega_{i+1} = {}^{i+1}R_i{}^{i}\omega_i + \dot{\theta}_{i+1}{}^{i+1}\mathbf{z}_{i+1} \tag{3.131}$$

$$^{i+1}\mathbf{v}_{i+1} = {}^{i+1}R_{i+1}({}^{i}\mathbf{v}_i + {}^{i}\omega_i \times {}^{i}\mathbf{p}_{i+1}) + \dot{d}_{i+1}{}^{i+1}\mathbf{z}_{i+1} \tag{3.132}$$

Starting from the first fixed link, we can propagate to find the velocities at the end-effector, and then extract the Jacobian matrix.

3.10.2 Example: 3 DOF RRR Arm

Let us illustrate this method on the 3 DOF revolute manipulator we have been using in this Chapter.

For the linear velocity we obtain:

$$\mathbf{v}_{P_1} = 0 \tag{3.133}$$

$$\mathbf{v}_{P_2} = \mathbf{v}_{P_1} + \omega_1 \times P_2 \tag{3.134}$$

$$\mathbf{v}_{P_3} = \mathbf{v}_{P_2} + \omega_2 \times P_3 \tag{3.135}$$

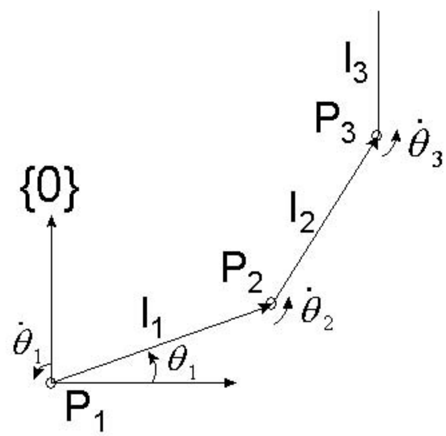

Figure 3.17: Example of Velocity Propagation

or

$$ {}^0\mathbf{v}_{P_2} = 0 + \begin{bmatrix} 0 & -\dot{\theta}_1 & 0 \\ \dot{\theta}_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 \\ l_1 c_1 \\ 0 \end{bmatrix} \dot{\theta}_1 \tag{3.136} $$

and

$$ {}^0\mathbf{v}_{P_3} = \begin{bmatrix} -l_1 s_1 \\ l_1 c_1 \\ 0 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} (\dot{\theta}_1 + \dot{\theta}_2) {}^0\mathbf{p}_3 \tag{3.137} $$

$$ {}^0\mathbf{v}_{P_3} = \begin{bmatrix} -l_1 s_1 \\ l_1 c_1 \\ 0 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} -l_2 s_{12} \\ l_2 c_{12} \\ 0 \end{bmatrix} (\dot{\theta}_1 + \dot{\theta}_2) \tag{3.138} $$

The angular velocities are simple since they are all rotations about the Z axis perpendicular to the plane of the paper.

$$ {}^0\omega_3 = (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) {}^0\mathbf{z}_0 \tag{3.139} $$

In matrix form

$$^0\mathbf{v}_{P_3} = \begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} & 0 \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{3.140}$$

and

$$^0\omega_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \tag{3.141}$$

from which we obtain the Jacobian:

$$\begin{pmatrix} \mathbf{v} \\ \omega \end{pmatrix} = J \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} \tag{3.142}$$

This Jacobian is clearly the same as previously calculated using the explicit method.

3.10.3 Virtual Work

The relationship between end-effector forces and joint torques can be directly established using the *virtual work principle.* This principal states that *at static equilibrium the virtual work of all applied forces is equal to zero.*

The virtual work principal allows to avoid computing and eliminating internal forces. Since internal forces do not produce any work, they are not involved in the analysis.

Joint torques and end-effector forces are the only applied or active forces for this mechanism. Let $\mathbf{F}$ be the vector of applied forces and moments at the end effector,

$$\mathbf{F} = \begin{pmatrix} \mathbf{f} \\ \mathbf{n} \end{pmatrix} \tag{3.143}$$

At static equilibrium, the virtual work is

$$\tau^T \delta \mathbf{q} + (-\mathbf{F})^T \delta \mathbf{x} = 0 \tag{3.144}$$

Note that the minus sign is due to the fact that forces at the end effector are applied by the environment to the end-effector.

Using the relationship

$$\delta \mathbf{x} = J \delta \mathbf{q}$$

yields

$$\tau = J^T \mathbf{F} \tag{3.145}$$

This is an important relationship not only for the the analysis of static forces but also for robot control.

3.11 More on Explicit Form: J_v

We have seen how the linear motion Jacobian, J_v, can be obtained from direct differentiations of the end-effector position vector. We develop here the explicit form for obtaining this matrix. The expression for the linear velocity was found in the form

$$\mathbf{v} = \sum_{i=1}^{n} [\epsilon_i \mathbf{z}_i + \bar{\epsilon}_i (\mathbf{z}_i \times \mathbf{p}_{in})] \dot{q}_i \tag{3.146}$$

$$\mathbf{v} = [\epsilon_1 \mathbf{z}_1 + \bar{\epsilon}_1 (\mathbf{z}_1 \times \mathbf{p}_{1n})] \dot{q}_1 + [\epsilon_2 \mathbf{z}_2 + \bar{\epsilon}_2 (\mathbf{z}_2 \times \mathbf{p}_{2n})] \dot{q}_2 + \cdots + [\epsilon_n \mathbf{z}_n + \bar{\epsilon}_n (\mathbf{z}_n \times \mathbf{p}_{nn})] \dot{q}_n \tag{3.147}$$

and the corresponding Jacobian is

$$J_v = [\, \epsilon_1 \mathbf{z}_1 + \bar{\epsilon}_1 (\mathbf{z}_1 \times \mathbf{p}_{1n}) \quad \cdots \quad \epsilon_n \mathbf{z}_n + \bar{\epsilon}_n (\mathbf{z}_n \times \mathbf{p}_{nn}) \,]\, \dot{\mathbf{q}} \tag{3.148}$$

In this form, J_v is expressed in terms of the $\mathbf{z}_i$ vectors and $\mathbf{p}_{in}$ vectors associated with the various links. Combining the linear and angular parts, the Jacobian J is

$$\begin{pmatrix} (\epsilon_1\mathbf{z}_1 + \bar{\epsilon}_1\mathbf{z}_1 \times \mathbf{p}_{1n}) & \cdots & (\epsilon_{n-1}\mathbf{z}_{n-1} + \bar{\epsilon}_{n-1}\mathbf{z}_{n-1} \times \mathbf{p}_{(n-1)n}) & \epsilon_n\mathbf{z}_n \\ \bar{\epsilon}_1\mathbf{z}_1 & \cdots & \bar{\epsilon}_{n-1}\mathbf{z}_{n-1} & \bar{\epsilon}_n\mathbf{z}_n \end{pmatrix}. \tag{3.149}$$

To express this matrix in a given frame, all vectors should be evaluated in that frame. The cross product $(\mathbf{z}_i \times \mathbf{p}_{in})$ can be evaluated in frame $\{i\}$. Again, since the components in $\{i\}$ of $\mathbf{z}_i$ are independent of frame $\{i\}$, we define

$$\widehat{Z} =^i \widehat{\mathbf{z}}_i = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The components in frame $\{i\}$ of cross product vectors $(\mathbf{z}_i \times \mathbf{p}_{in})$ are simply $(\widehat{Z}\ {}^i\mathbf{p}_{in})$.

The components in the frame $\{i\}$ of the vector $\mathbf{p}_{in}$ are given in the last column of the transformation ${}^i_nT$.

The expression in frame $\{0\}$ of the Jacobian matrix, ${}^0J$ is given by

$$\begin{pmatrix} {}^0_1R(\epsilon_1 Z + \bar{\epsilon}_1\widehat{Z}^1\mathbf{p}_{1n}) & \cdots & {}^0_{n-1}R(\epsilon_{n-1}Z + \bar{\epsilon}_{n-1}\widehat{Z}^{n-1}\mathbf{p}_{(n-1)n}) & {}^0_nR\epsilon_n\mathbf{Z} \\ {}^0_1R\bar{\epsilon}_1 Z & \cdots & {}^0_{n-1}R\bar{\epsilon}_{n-1}Z & {}^0_nR\bar{\epsilon}_n Z \end{pmatrix}. \tag{3.150}$$

3.11.1 Stanford Scheinman arm example

Applying the explicit form of J_v to the Stanford Scheinman arm, we can easily (by setting the numerical values of ϵ_i) write the Jacobian as

$${}^0J = \begin{pmatrix} {}^0(\mathbf{z}_1 \times \mathbf{p}_{13}) & {}^0(\mathbf{z}_2 \times \mathbf{p}_{23}) & {}^0\mathbf{z}_3 & 0 & 0 & 0 \\ {}^0\mathbf{z}_1 & {}^0\mathbf{z}_2 & 0 & {}^0\mathbf{z}_4 & {}^0\mathbf{z}_5 & {}^0\mathbf{z}_6 \end{pmatrix} \tag{3.151}$$

$\mathbf{p}_{13}$ is given in ${}^1_3T$ in frame $\{1\}$

$$ {}^1_3T = \begin{pmatrix} & {}^1_3R & & {}^1\mathbf{p}_{13} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_2 & 0 & s_2 & d_3 s_2 \\ 0 & 1 & 0 & d_2 \\ -s_2 & 0 & c_2 & d_3 c_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.152} $$

To express $(\mathbf{z}_1 \times \mathbf{p}_{13})$ in frame $\{0\}$, we have

$$ {}^0(\mathbf{z}_1 \times \mathbf{p}_{13}) = {}^0_1R.({}^1\mathbf{z}_1 \times {}^1\mathbf{p}_{13}) \tag{3.153} $$

The computation of ${}^i\mathbf{z}_i \times {}^iP_{in}$ in frame $\{i\}$ is simply

$$ ({}^i\mathbf{z}_i \times {}^i\mathbf{p}) = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \end{pmatrix} = \begin{pmatrix} -\mathbf{p}_y \\ \mathbf{p}_x \\ 0 \end{pmatrix} \tag{3.154} $$

For ${}^1\mathbf{z}_1 \times {}^iP_{13}$, this computation is

$$ {}^1\mathbf{p}_{13} = \begin{pmatrix} d_3 s_2 \\ d_2 \\ d_3 c_2 \end{pmatrix} \tag{3.155} $$

and

$$ {}^1\mathbf{z}_1 \times {}^1\mathbf{p}_{13} = \begin{pmatrix} -d_2 \\ d_3 s_2 \\ 0 \end{pmatrix} \tag{3.156} $$

In frame $\{0\}$ this is

$$ {}^0_1R({}^1\mathbf{z}_1 \times {}^1\mathbf{p}_{13}) = \begin{pmatrix} c_1 & s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -d_2 \\ d_3 s_2 \\ 0 \end{pmatrix} = \begin{pmatrix} -c_1 d_2 - s_1 s_2 d_3 \\ -s_1 d_2 + c_1 s_2 d_3 \\ 0 \end{pmatrix} \tag{3.157} $$

For ${}^0(\mathbf{z}_2 \times \mathbf{p}_{23})$, we can similarly obtain

$$ {}^2\mathbf{p}_{23} = \begin{bmatrix} 0 \\ -d_3 \\ 0 \end{bmatrix} \leftarrow {}^2_3T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.158} $$

and

$$ {}^2\mathbf{z}_2 \times {}^2\mathbf{p}_{23} = \begin{pmatrix} d_3 \\ 0 \\ 0 \end{pmatrix} \tag{3.159} $$

Since

$$ {}^0_2R = \begin{pmatrix} c_1c_2 & X & X \\ s_1c_2 & X & X \\ -s_2 & X & X \end{pmatrix} \tag{3.160} $$

we obtain

$$ {}^0_2R({}^2\mathbf{z}_2 \times {}^2\mathbf{p}_{23}) = \begin{pmatrix} c_1c_2d_3 \\ s_1c_2d_3 \\ -s_2d_3 \end{pmatrix} \tag{3.161} $$

Finally $\mathbf{z}_3$ in frame $\{0\}$ is

$$ {}^0\mathbf{z}_3 = \leftarrow {}^0_3R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{pmatrix} c_1s_2 \\ s_1s_2 \\ c_2 \end{pmatrix} \tag{3.162} $$

The Jacobian in frame $\{0\}$ is, as expected, the same as the one derived earlier:

$$\begin{bmatrix} -c_1d_2 - s_1s_2d_3 & c_1c_2d_3 & c_1s_2 & 0 & 0 & 0 \\ -s_1d_2 + c_1s_2d_3 & s_1c_2d_3 & s_1s_2 & 0 & 0 & 0 \\ 0 & -s_2d_3 & c_2 & 0 & 0 & 0 \\ 0 & -s_1 & 0 & c_1s_2 & -c_1c_2s_4 - s_1c_4 & c_1c_2c_4s_5 - s_1s_4s_5 + c_1s_2s_5 \\ 0 & c_1 & 0 & s_1s_2 & -s_1c_2s_4 + c_1c_4 & s_1c_2c_4s_5 + c_1s_4s_5 + s_1s_2s_5 \\ 1 & 0 & 0 & c_2 & s_2s_4 & -s_2c_4s_5 + c_5c_2 \end{bmatrix}$$

There is yet another approach to compute the vectors $\mathbf{p}_{in}$, this is discussed in the next section.

3.11.2 $\mathbf{p}_{in}$ Derivation

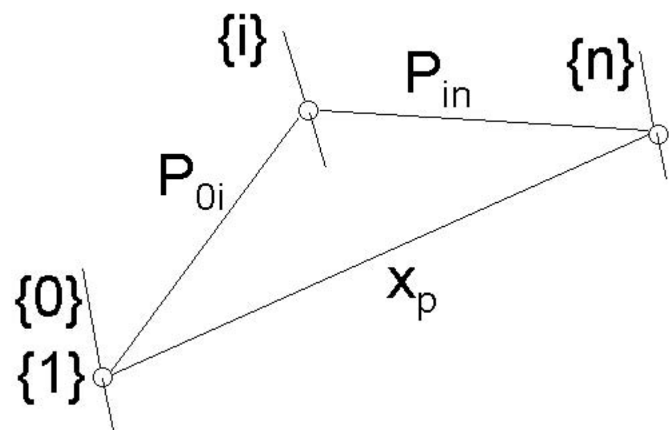

Figure 3.18: Computing $\mathbf{p}_{in}$

The computation in frame $\{i\}$ of the vector $\mathbf{p}_{in}$ requires ${}^i_nT$. However, this transformation is often not explicitly available, as only the matrices: ${}^0_1T$, ${}^0_2T$, ..., ${}^0_iT$, ..., ${}^0_nT$ are computed. In this case, it is more efficient to express $\mathbf{p}_{in}$ as

$$\mathbf{p}_{in} = \mathbf{x}_p - \mathbf{p}_{0i}.$$

The vector $\mathbf{x}_p$ and $\mathbf{p}_{0i}$ are expressed in frame $\{0\}$, ${}^0\mathbf{p}_{0i}$ is given in ${}^0_iT$.

The computation of $({}^0_iR\ \widehat{Z}\ {}^i\mathbf{p}_{in})$ that appears in (3.150) can then be replaced by

$$({}^0_iR\ \widehat{Z}\ {}^i\mathbf{p}_{in}) \quad \Longrightarrow \quad ({}^0_iR\ \widehat{Z}\ {}^0_iR^T)\ \ ({}^0\mathbf{x}_p - {}^0\mathbf{p}_{0i}).$$

Chapter 4

Dynamics

4.1 Introduction

The study of manipulator dynamics is essential for both the analysis of performance and the design of robot control. A manipulator is a multi-link, highly nonlinear and coupled mechanical system. In motion, this system is subjected to inertial, centrifugal, Coriolis, and gravity forces, which can greatly affect its performance in the execution of a task. If ignored, these dynamics may also lead to control instability, especially for tasks that involve contact interactions with the environment. Our goal here is to model the dynamics and establish the manipulator equation of motion in order to develop the appropriate control structures needed to achieve robot's stability and performance.

There are various formulations for modeling the dynamics of manipulators. We will discuss a recursive algorithm based on the Newton-Euler formulation, and present an approach for the explicit model, based on Lagrange's formulation. These two methodologies are similar to the recursive and explicit approaches we presented earlier for the kinematic model and the Jacobian matrix.

In the Newton-Euler method, the analysis is based on isolating each link and considering all the forces acting on it. This analysis is similar to the previous study of static forces, which lead to the relationship between

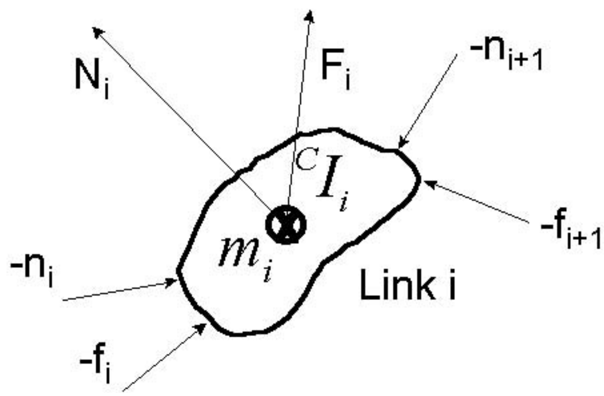

Figure 4.1: Link's Dynamics

end-effector forces and joint torques, i.e. $\tau = J^T\mathbf{F}$. The difference with the previous analysis is that now we must account for the inertial forces acting on the manipulator links.

For link i, we consider the forces $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$, and the moments n_i and n_{i+1} acting at joints i and $i+1$. Because of the motion of the link, we must include the inertial forces associated with this motion. Let F_i and N_i be the inertial forces corresponding to the linear motion and angular motion respectively, expressed at the center of mass of the link. These dynamic forces are given by the equations of Newton (linear motion) and Euler (angular motion),

$$m_i \dot{\mathbf{v}}_{C_i} = F_i \tag{4.1}$$

$$^{C_i}I\dot{\omega}_i + \omega_i \times {}^{C_i}I\omega_i = N_i \tag{4.2}$$

where m_i and ^{C_i}I are the mass and link's tensor of inertia at the center of mass.

Similarly to the static analysis we have seen, recursive force and moment relationships can be developed, and internal forces and moments can be eliminated by projection on the joint axis,

$$\tau_i = \begin{cases} \mathbf{n}_i^T Z_i & \text{for a revolute joint} \\ \mathbf{f}_i^T Z_i & \text{for a prismatic joint} \end{cases} \tag{4.3}$$

The Newton-Euler algorithm consists of two propagation phases. A *forward propagation* of velocities, accelerations, and dynamic forces. *Backward propagation* then eliminates internal forces and moments. Internal forces are transmitted through the structure.

Lagrange's formulation relies on the concept of energy, the kinetic energy K and the potential energy U of the system. The kinetic energy is expressed in terms of the manipulator mass matrix M and the generalized velocities $\dot{\mathbf{q}}$ in the following quadratic form,

$$K = \frac{1}{2}\dot{\mathbf{q}}^T M \dot{\mathbf{q}} \tag{4.4}$$

Given the potential energy V, the Lagrangian is

$$L = K - V \tag{4.5}$$

and Lagrange's equations of motion are

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial L}{\partial q} = \tau \tag{4.6}$$

where τ is the vector of applied generalized torques. Both formalisms, Newton-Euler and Lagrange. leads to the same set of equations, which can be developed in the form

$$M\ddot{\mathbf{q}} + \mathbf{v} + \mathbf{g} = \tau \tag{4.7}$$

where $\mathbf{g}$ is the vector of gravity forces and $\mathbf{v}$ is the vector of centrifugal and Coriolis forces. These equations provides the relationship between torques applied to the manipulator and the resulting accelerations and velocities.

Analysis of Lagrange's equations shows that the coefficients involved in $\mathbf{v}$ can be obtained from M. This reduces the problem to finding M

and $\mathbf{g}$. The mass matrix M can be directly found from the total kinetic energy of the mechanism, and $\mathbf{g}$ can be determined simply from static analysis. This provides an *explicit* form of the equation of motion.

4.2 Newton-Euler Formulation

Newton's equation provides a description of the linear motion. Euler's equation, which describes the angular motion, involves the notion of angular momentum and the link's inertia tensor.

4.2.1 Linear and Angular Momentum

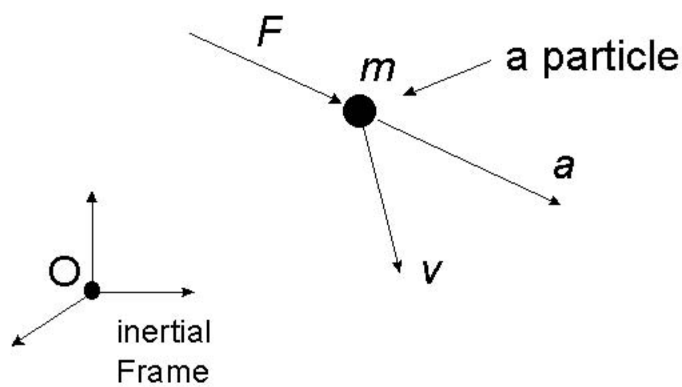

Figure 4.2: A particle's dynamics

Let us start with a simple particle. The kinetic energy of a particle with a velocity $\mathbf{v}$ is $1/2m\mathbf{v}^2$. Newton's law gives us the equation for the acceleration of the particle $\mathbf{a}$ with respect to an inertial frame, given an applied force $\mathbf{F}$

$$m\mathbf{a} = \mathbf{F}$$

This equation can be also written in terms of the linear momentum, $m\mathbf{v}$ of this particle. The rate of change of the linear momentum is equal to the applied forces,

$$\frac{d}{dt}(m\mathbf{v}) = \mathbf{F} \tag{4.8}$$

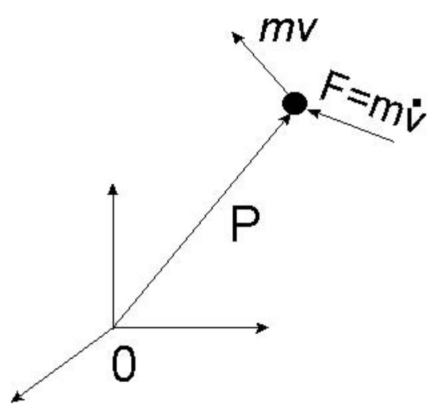

Figure 4.3: Angular Momentum Computation

To introduce the angular momentum, we take the moment of the forces that appear on both sides of the above equation. The moment N of $\mathbf{F}$ with respect to some point O is the cross product of the vector $\mathbf{p}$ locating the particle and the vector F. Taking the moment with respect to the same point of the left hand side of the equation yields

$$\mathbf{p} \times m\dot{\mathbf{v}} = \mathbf{p} \times \mathbf{F} = N \tag{4.9}$$

Let us consider the rate of change of the quantity $\mathbf{p} \times m\mathbf{v}$,

$$\frac{d}{dt}(\mathbf{p} \times m\mathbf{v}) = \mathbf{p} \times m\dot{\mathbf{v}} + \mathbf{v} \times m\mathbf{v} = \mathbf{p} \times m\dot{\mathbf{v}} \tag{4.10}$$

This yields

$$\frac{d}{dt}(\mathbf{p} \times m\mathbf{v}) = N \tag{4.11}$$

The quantity $\mathbf{p} \times m\mathbf{v}$ is the angular momentum with respect to O of the particle. Thus the rate of change of the angular momentum is equal to the applied moment. This equation complements the one above for the rate of change of the linear momentum and the applied forces.

4.2.2 Euler Equation

To develop Euler's equations, we must extend our previous result to the rigid body case. A rigid body can be treated as a large set of particles, and the previous analysis can be extended to the sum over this set.

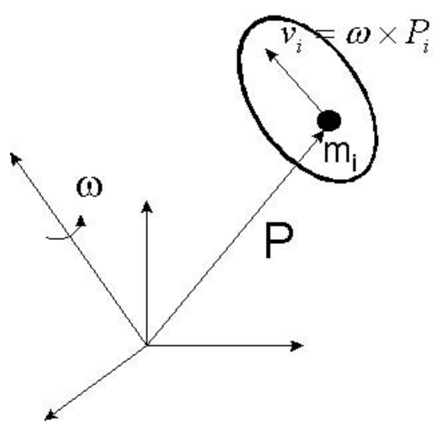

Figure 4.4: Rigid body rotational motion

Let us consider the angular motion of a rigid body rotating with respect to some fixed point O at an angular velocity ω. The linear velocity, $\mathbf{v}_i$, of a particle i of this rigid body is $\omega \times \mathbf{p}_i$. where $\mathbf{p}_i$ is the position vector for the particle with respect to O. The angular momentum, Φ, of the rigid body – the sum of the angular momentums of all particles – is

$$\Phi = \sum_i m_i \mathbf{p}_i \times (\omega \times \mathbf{p}_i) \tag{4.12}$$

Let us assume that the mass density of the rigid body is ρ. The mass m_i can be approximated by the product of the density of the rigid body ρ by a small volume dv occupied by a particle. Integrating over the rigid body's volume, V, we obtain

$$\Phi = \int_V \mathbf{p} \times (\omega \times \mathbf{p}) \rho dv \tag{4.13}$$

Observing that ω is independent of the variable in this integral, and replacing $\mathbf{p}\times$ by the cross product operator $\hat{\mathbf{p}}$, yields

$$\Phi = [\int_V -\hat{\mathbf{p}}\hat{\mathbf{p}} \rho dv]\omega \tag{4.14}$$

The quantity in brackets is called the inertia tensor of the rigid body, I, hence

$$I = [\int_V -\hat{\mathbf{p}}\hat{\mathbf{p}} \rho dv]$$

Finally, the angular momentum of this rigid body is

$$\Phi = I\omega \tag{4.15}$$

Euler's equations for the rotational motion with respect to some point O state that the rate of change of the angular momentum of the rigid body is equal to the applied moments

$$\dot{\Phi} = I\dot{\omega} + \omega \times I\omega = N \tag{4.16}$$

Together with Newton's law, these equations provide the description of the linear and angular motions for a manipulator, subjected to external forces.

4.2.3 Inertia Tensor

The inertia tensor I is defined by

$$I = \int_V -\hat{\mathbf{p}}\hat{\mathbf{p}}\rho dv \tag{4.17}$$

The quantity $-\hat{\mathbf{p}}\hat{\mathbf{p}}$ can be computed as

$$(-\hat{\mathbf{p}}\hat{\mathbf{p}}) = (\mathbf{p}^T\mathbf{p})I_3 - \mathbf{p}\mathbf{p}^T \tag{4.18}$$

The inertia tensor is therefore

$$I = \int_V [(\mathbf{p}^T\mathbf{p})I_3 - \mathbf{p}\mathbf{p}^T]\rho dv \tag{4.19}$$

Let us consider a Cartesian representation for the position vector $\mathbf{p}$,

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.20}$$

The term in the integral is

$$[(\mathbf{p}^T\mathbf{p})I_3 - \mathbf{p}\mathbf{p}^T] = \begin{bmatrix} y^2+z^2 & -xy & -xz \\ -xy & z^2+x^2 & -yz \\ -xz & -yz & x^2+y^2 \end{bmatrix} \tag{4.21}$$

The inertia tensor I is represented by the matrix

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{4.22}$$

where

$$I_{xx} = \iiint (y^2 + z^2)\rho dx dy dz \tag{4.23}$$

$$I_{yy} = \iiint (z^2 + x^2)\rho dx dy dz \tag{4.24}$$

$$I_{zz} = \iiint (x^2 + y^2)\rho dx dy dz \tag{4.25}$$

$$I_{xy} = \iiint xy\rho dx dy dz \tag{4.26}$$

$$I_{xz} = \iiint xz\rho dx dy dz \tag{4.27}$$

$$I_{yz} = \iiint yz\rho dx dy dz \tag{4.28}$$

I_{xx}, I_{yy}, and I_{zz} are called the moments of inertia and I_{xy}, I_{yz} and I_{zx} are called products of inertia. When the matrix I is diagonal, the diagonal moments of inertia are called the *principal moments of inertia.*

Parallel Axis Theorem

Because of the symmetries generally found in rigid bodies, it is more efficient to compute the body's inertia tensor with respect to its center of mass. If needed with respect to another point and axes, the inertia tensor can be obtained from the tensor computed at the center of mass through a translation and rotation transformation, determined by the parallel axis theorem.

Assuming the the inertia tensor has been computed with respect to the frame {C} (at the body's center of mass), to find the inertia tensor with respect to another frame {A}, whose axes are parallel to those of {C}, we can proceed as follows.

Let $\mathbf{p}_C$ be the vector locating point C in frame {A}. The parallel axis theorem states:

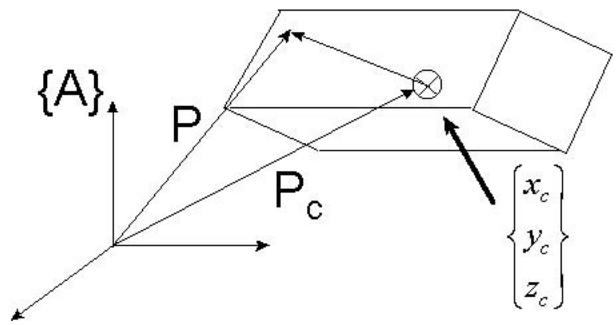

Figure 4.5: Parallel Axis Theorem

$$^{A}I = {}^{C}I + m[(\mathbf{p}_C^T\mathbf{p}_C)I_3 - \mathbf{p}_C\mathbf{p}_C^T] \tag{4.29}$$

If (x_c, y_c, z_c) are the Cartesian coordinates of point C in frame {A}, the relationships between the two tensors are

$$^{A}I_{zz} = {}^{C}I_{zz} + m(x_C^2 + y_C^2) \tag{4.30}$$

$$^{A}I_{xy} = {}^{C}I_{xy} + m x_C y_C \tag{4.31}$$

Rotation Transformation Let us consider the case where we wish to express the inertia tensor with respect to another frame rotated with respect to the rigid body frame. The angular momentum expressed in frame {A} is

$$^{A}\Phi = {}^{A}I\, {}^{A}\omega$$

Let's express this quantity in a frame {B}, having the same origin as {A} and obtained by a rotation ${}^{B}_{A}R$,

$$^{B}\Phi = {}^{B}_{A}R\, {}^{A}\Phi = {}^{B}_{A}R\, {}^{A}I\, {}^{A}\omega$$

where

$$^{A}\omega = {}^{A}_{B}R^{B}\omega = {}^{B}_{A}R^{T\,B}\omega$$

thus

$$^{B}\Phi = {}^{B}_{A}R^{A}I({}^{B}_{A}R^{T}\ {}^{B}\omega)$$

also

$$^{B}\Phi = {}^{B}I\ {}^{B}\omega$$

finally

$$^{B}I = [{}^{B}_{A}R^{A}I^{B}_{A}R^{T}]$$

The relationship described above is a similarity transformation. For a general frame transformation involving both translation and rotation. We first proceed with a translation using the parallel axis theorem, and then apply the similarity transformation for the rotation.

4.3 Lagrange Formulation

Given a set of *generalized coordinates,* **q**, describing the configuration of a mechanism, there is a set of corresponding *generalized forces,* τ, acting along (or about) each of these coordinates. If the coordinate q_i represents the rotation of a revolute joint, the corresponding force τ_i would be a torque acting about the joint axis. For a prismatic joint, τ_i is a force acting along the axis of the joint.

Lagrange's equations involve a scalar quantity L, the Lagrangian, which represents the difference between the two scalars corresponding to the kinetic energy K and the potential energy V of the mechanism,

$$L = K - V \tag{4.32}$$

The Lagrangian, L is a function, of the generalized coordinates $\mathbf{q}$ and the generalized velocities, $\dot{\mathbf{q}}$.

$$L = L(\mathbf{q}, \dot{\mathbf{q}})$$

. For an n DOF mechanism, the Lagrange formulation provides the n equations of motion in the following from

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\mathbf{q}}}) - \frac{\partial L}{\partial \mathbf{q}} = \tau \tag{4.33}$$

Since the potential energy (due to the gravity) is only dependent on the configuration, these equations can be written as

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) - \frac{\partial K}{\partial \mathbf{q}} + \frac{\partial U}{\partial \mathbf{q}} = \tau \tag{4.34}$$

The first two terms define the inertial forces associated with the motion of the mechanism, and the third term represents the gradient of the gravity potential acting on it. This gradient is the gravity force vector.

For a single mass m with a velocity v, the kinetic energy is $1/2(v^T m v)$. In the case of a multi-link manipulator with a mass matrix M and generalized velocities, $\dot{\mathbf{q}}$, the kinetic energy is the scalar given by the quadratic form

$$K = \frac{1}{2}\dot{\mathbf{q}}^T M \dot{\mathbf{q}} \tag{4.35}$$

Using this expression of K we can write

$$\frac{\partial K}{\partial \dot{\mathbf{q}}} = \frac{\partial}{\partial \dot{\mathbf{q}}}(\frac{1}{2}\dot{\mathbf{q}}^T M(q)\dot{\mathbf{q}}) = M\dot{\mathbf{q}} \tag{4.36}$$

Differentiating with respect to time we obtain:

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) = M\ddot{\mathbf{q}} + \dot{M}\dot{\mathbf{q}} \tag{4.37}$$

The inertial forces in the equation of Lagrange can be expressed as

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) - \frac{\partial K}{\partial \mathbf{q}} = M\ddot{\mathbf{q}} + \dot{M}\dot{\mathbf{q}} - \frac{1}{2}\begin{bmatrix} \dot{\mathbf{q}}^T \frac{\partial M}{\partial q_1}\dot{\mathbf{q}} \\ \vdots \\ \dot{\mathbf{q}}^T \frac{\partial M}{\partial q_n}\dot{\mathbf{q}} \end{bmatrix} \tag{4.38}$$

This equation can be developed in the form

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) - \frac{\partial K}{\partial \mathbf{q}} = M\ddot{\mathbf{q}} + \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) \tag{4.39}$$

where $\mathbf{v}$ is the vector of centrifugal and Coriolis forces given by

$$\mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{M}\dot{\mathbf{q}} - \frac{1}{2}\begin{bmatrix} \dot{\mathbf{q}}^T \frac{\partial M}{\partial q_1}\dot{\mathbf{q}} \\ \vdots \\ \dot{\mathbf{q}}^T \frac{\partial M}{\partial q_n}\dot{\mathbf{q}} \end{bmatrix} \tag{4.40}$$

Finally adding the inertial and gravity terms in the Lagrange equations, yields

$$M(q)\ddot{\mathbf{q}} + \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \tau \tag{4.41}$$

The vector of centrifugal and Coriolis forces can be expressed as

$$\mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) = C(\mathbf{q})[\dot{\mathbf{q}}^2] + B(\mathbf{q})[\dot{\mathbf{q}}\dot{\mathbf{q}}] \tag{4.42}$$

4.3.1 Explicit Form of the Mass Matrix

The mass matrix M plays a central role in the dynamics of manipulator. In particular, the elements of the matrices B and C can be completely determined from this matrix.

Because of its additive property, the kinetic energy of the total system is the sum of the kinetic energies associated with its links.

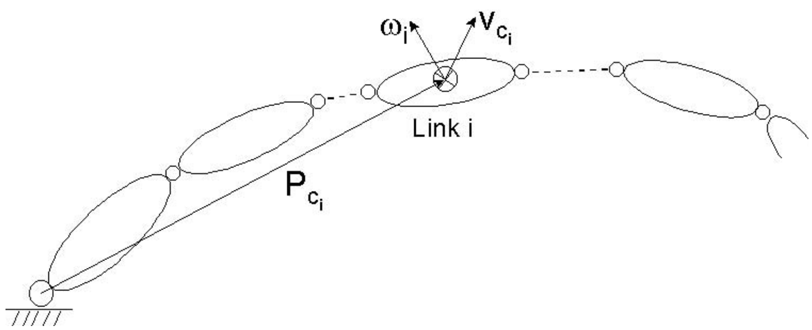

Figure 4.6: Explicit Form

$$K = \sum_{i=1}^{n} K_i \tag{4.43}$$

The kinetic energy of a link has two components: one that is due to its linear motion, and the second due to its rotational motion. If the linear velocity of the center of mass of a link is $\mathbf{v}_{C_i}$, and if the angular velocity of the link is ω_i, the kinetic energy, K_i of this link is

$$K_i = \frac{1}{2}(m_i \mathbf{v}_{C_i}^T \mathbf{v}_{C_i} + \omega_i^{T C} I_i \omega_i) \tag{4.44}$$

where ${}^C I_i$ is the inertia tensor of link i computed with respect to the link's center of mass, C_i. The linear velocity at the center v_{C_i} can be expressed as a linear combination of the joint velocities, $\dot{\mathbf{q}}$. Introducing a Jacobian matrix, J_{v_i}, corresponding to the linear motion of the center-of-mass of link i, the velocity vector v_{C_i} can be written as

$$\mathbf{v}_{C_i} = J_{v_i} \dot{\mathbf{q}} \tag{4.45}$$

where

$$J_{v_i} = [\, \frac{\partial \mathbf{p}_{C_i}}{\partial q_1} \quad \frac{\partial \mathbf{p}_{C_i}}{\partial q_2} \quad \cdots \quad \frac{\partial \mathbf{p}_{C_i}}{\partial q_i} \quad 0 \quad 0 \quad \cdots \quad 0 \,] \tag{4.46}$$

In this matrix, the columns $i+1$ to n in J_{v_i} are zero columns, since the velocity $\mathbf{v}_{C_i}$ at the center of mass of link i is independent of the velocities of joint $i+1$ to joint n. Similarly the angular velocity can be expressed as

$$\omega_{C_i} = J_{\omega_i}\dot{\mathbf{q}} \tag{4.47}$$

where

$$J_{\omega_i} = [\,\bar{\varepsilon}_1\mathbf{z}_1 \quad \bar{\varepsilon}_2\mathbf{z}_2 \quad \cdots \quad \bar{\varepsilon}_i\mathbf{z}_i \quad 0 \quad 0 \quad \cdots \quad 0\,] \tag{4.48}$$

Using these expressions, the kinetic energy becomes

$$K = \frac{1}{2}\sum_{i=1}^{n}(m_i\dot{\mathbf{q}}^T J_{v_i}^T J_{v_i}\dot{\mathbf{q}} + \dot{\mathbf{q}}^T J_{\omega_i}^{T\,C} I_i J_{\omega_i}\dot{\mathbf{q}}) \tag{4.49}$$

Factoring out the generalized velocities, the kinetic energy can be expressed as

$$K = \frac{1}{2}\dot{\mathbf{q}}^T[\sum_{i=1}^{n}(m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^{T\,C} I_i J_{\omega_i})]\dot{\mathbf{q}} \tag{4.50}$$

Equating this expression to the quadratic form of the kinetic energy leads to the following explicit form of the mass matrix$M$,

$$M = \sum_{i=1}^{n}(m_i J_{v_i}^T J_{v_i} + J_{\omega_i}^{T\,C} I_{C_i} J_{\omega_i}) \tag{4.51}$$

The mass matrix M is a symmetric positive definite matrix, i.e. $m_{ij} = m_{ji}$ and $\dot{\mathbf{q}}^T M\dot{\mathbf{q}} > 0$ for $\dot{\mathbf{q}} \neq 0$

4.3.2 Centrifugal and Coriolis Forces

We now consider the relationships between the matrices B and C with the matrix M. These relationships can be obtained from the development of equation 4.40 defining the vector of centrifugal and Coriolis forces

$$\mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{M}\dot{\mathbf{q}} - \frac{1}{2}\begin{bmatrix} \dot{\mathbf{q}}^T \frac{\partial M}{\partial q_1}\dot{\mathbf{q}} \\ \vdots \\ \dot{\mathbf{q}}^T \frac{\partial M}{\partial q_n}\dot{\mathbf{q}} \end{bmatrix}$$

This equation involves time derivatives and partial derivatives of the elements m_{ij} of the matrix M. We denote by m_{ijk} the partial derivatives

$$m_{ijk} \equiv \frac{\partial m_{ij}}{\partial q_k} \tag{4.52}$$

The time derivative of an element m_{ij} is

$$\frac{dm_{ij}}{dt} = \sum_{k=1}^{n} m_{ijk}\dot{q}_k$$

To simplify the development, let us consider the case of a 2 DOF manipulator. The mass matrix is

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{pmatrix} \tag{4.53}$$

The vector $\mathbf{v}$ of centrifugal and Coriolis forces is

$$\mathbf{v} = \dot{M}\dot{\mathbf{q}} - \frac{1}{2}\begin{bmatrix} \dot{\mathbf{q}}^T M_{q_1}\dot{\mathbf{q}} \\ \dot{\mathbf{q}}^T M_{q_2}\dot{\mathbf{q}} \end{bmatrix} = \begin{pmatrix} \dot{m}_{11} & \dot{m}_{12} \\ \dot{m}_{12} & \dot{m}_{22} \end{pmatrix}\dot{\mathbf{q}} - \frac{1}{2}\begin{bmatrix} \dot{\mathbf{q}}^T \begin{pmatrix} m_{111} & m_{121} \\ m_{121} & m_{221} \end{pmatrix}\dot{\mathbf{q}} \\ \dot{\mathbf{q}}^T \begin{pmatrix} m_{112} & m_{122} \\ m_{122} & m_{222} \end{pmatrix}\dot{\mathbf{q}} \end{bmatrix}$$

These expressions can be developed in the form

$$\mathbf{v}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} \frac{1}{2}(m_{111}+m_{111}-m_{111}) & \frac{1}{2}(m_{122}+m_{122}-m_{221}) \\ \frac{1}{2}(m_{211}+m_{211}-m_{112}) & \frac{1}{2}(m_{222}+m_{222}-m_{222}) \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} m_{112}+m_{121}-m_{121} \\ m_{212}+m_{221}-m_{122} \end{bmatrix} [\dot{q}_1\dot{q}_2] \tag{4.54}$$

Expansion in this form shows a pattern of grouping of coefficients that leads to the following representation of Christoffel symbols,

$$b_{ijk} = \frac{1}{2}(m_{ijk} + m_{ikj} - m_{jki}) \tag{4.55}$$

Using these symbols the equation above can be written as:

$$\mathbf{v} = \begin{bmatrix} b_{111} & b_{122} \\ b_{211} & b_{222} \end{bmatrix} \begin{bmatrix} \dot{q}_1^2 \\ \dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} 2b_{112} \\ 2b_{212} \end{bmatrix} [\dot{q}_1\dot{q}_2] \tag{4.56}$$

In this equation, the first matrix corresponds to the matrix C of the coefficients associated with centrifugal forces, and the second matrix represents the matrix B corresponding to the the coefficients of Coriolis forces. In this case of 2 DOF, the matrix C is of dimension (2×2) and B is (2×1).

In the general case of n DOF, C is an $(n \times n)$ matrix, while B is of dimensions $(n \times \frac{(n-1)n}{2})$. Using these matrices, the vector $\mathbf{v}$ is

$$\mathbf{v}(\mathbf{q},\dot{\mathbf{q}}) = C(\mathbf{q})[\dot{\mathbf{q}}^2] + B(\mathbf{q})[\dot{\mathbf{q}}\dot{\mathbf{q}}] \tag{4.57}$$

$[\dot{\mathbf{q}}^2]$ is the symbolic representation of the $n \times 1$ vector of components $\dot{q}_i^2$ (square joint velocities),

$$[\dot{\mathbf{q}}^2]^T = [\dot{q}_1^2 \; \dot{q}_2^2 \; \dot{q}_3^2 \ldots \dot{q}_n^2]^T$$

$[\dot{\mathbf{q}}\dot{\mathbf{q}}]$ is the $(\frac{(n-1)n}{2} \times 1)$ vector of product of joint velocities

$$[\dot{\mathbf{q}}\dot{\mathbf{q}}]^T = [\dot{q}_1\dot{q}_2 \; \dot{q}_1\dot{q}_3 \ldots \dot{q}_1\dot{q}_n \; \dot{q}_2\dot{q}_3 \; \dot{q}_2\dot{q}_4 \ldots \dot{q}_2\dot{q}_n \ldots \dot{q}_{(n-1)} \; \dot{q}_n]^T$$

The general forms of the matrices B and C are

$$C(\mathbf{q}) = \begin{bmatrix} b_{1,11} & b_{1,22} & \cdots & b_{1,nn} \\ b_{2,11} & b_{2,22} & \cdots & b_{2,nn} \\ \vdots & \vdots & \vdots & \vdots \\ b_{n,11} & b_{n,22} & \cdots & b_{n,nn} \end{bmatrix} \tag{4.58}$$

and

$$B(\mathbf{q}) = \begin{bmatrix} 2b_{1,12} & 2b_{1,13} & \cdots & 2b_{1,1n} & 2b_{1,23} & \cdots & 2b_{1,2n} & \cdots & 2b_{1,(n-1)n} \\ 2b_{2,12} & 2b_{2,13} & \cdots & 2b_{2,1n} & 2b_{2,23} & \cdots & 2b_{2,2n} & \cdots & 2b_{2,(n-1)n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2b_{n,12} & 2b_{n,13} & \cdots & 2b_{n,1n} & 2b_{n,23} & \cdots & 2b_{n,2n} & \cdots & 2b_{n,(n-1)n} \end{bmatrix} \tag{4.59}$$

Because of the properties of the mass matrix, many of the elements b_{ijk} are zero. This symmetric, positive definite matrix represents the inertial properties of the manipulator with respect to joint motion. For instance, if joint 1 was revolute, m_{11} would represent the inertia (mass if it were prismatic) of the whole manipulator as it rotates about the joint axis 1. m_{11} is independent of the first joint, but varies with the configuration of the links following in the chain $(q_2, q_3, \ldots, q_n)$. Similarly m_{22} depends only on $q_3, \ldots, q_n$, and $m_{(n-1)(n-1)}$ depends only on q_n. Finally m_{nn} is a constant element. These properties result in a number of zero partial derivatives of the elements of the mass matrix, and leads to significant simplification of the elements involved in B and C.

4.3.3 Gravity Forces

The gravity forces are the gradient of the potential energy of the mechanism. The potential energy of link i increases with the elevation of its center of mass. This energy is proportional to the mass, the gravity constant, and to the height of the center of mass.

$$V_i = m_i g_0 h_i + V_0 \tag{4.60}$$

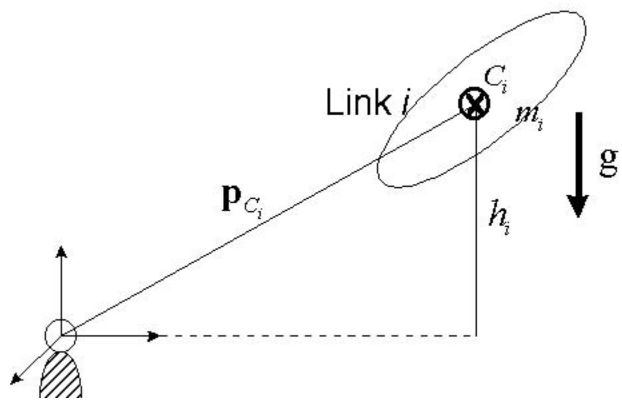

Figure 4.7: Potential Energy

Where V_0 represents the potential energy at some reference level. The height is given as the projection of the position vector $\mathbf{p}_{C_i}$ along the gravity direction,

$$V_i = m_i(-g^T p_{C_i}) \tag{4.61}$$

The potential energy of the whole manipulator is

$$V = \sum_i V_i \tag{4.62}$$

Using the matrix J_{v_i}, the gradient of the potential energy is

$$\mathbf{g} = -\begin{pmatrix} J_{v_1}^T & J_{v_2}^T & \cdots & J_{v_n}^T \end{pmatrix} \begin{pmatrix} m_1 g \\ m_2 g \\ \vdots \\ m_n g \end{pmatrix} \tag{4.63}$$

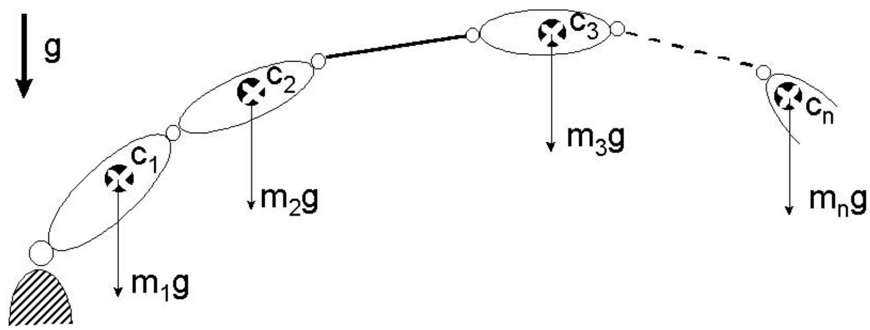

Figure 4.8: Gravity Vector

Direct Computation of g The gravity forces can be directly also by considering the gravity at the link's as weights acting at each link's center of mass. The gravity forces can then be directly computed as the torques needed to compensate for these weights. This leads

$$\mathbf{G} = -(J_{v_1}^T(m_1\mathbf{g}) + J_{v_2}^T(m_2\mathbf{g}) + \cdots + J_{v_n}^T(m_n\mathbf{g})) \tag{4.64}$$

We will use g (without boldface) as the scalar gravity constant, $g \approx 9.8\frac{m}{s^2}$. We will use $\mathbf{g}$ (with boldface) as the gravity acceleration vectory; the (3×1) vector $\mathbf{g}$ has a magnitude of g. Finally, we will use the vector $\mathbf{G}$ to indicate the $(n \times 1)$ vector of gravity-induced generalized torques in the Lagrange equations of motion.

4.3.4 Example: 2-DOF RP Manipulator

The links of the RP manipulator shown in Figure 4.9 have total masses of m_1 and m_2. The center of mass of link 1 is located at a distance l_1 from the joint axis 1, and the center of mass of link 2 is located at the distance d_2 from the joint axis 1. The inertia tensors of these links are

$$^{C}I_1 = \begin{pmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{pmatrix}; \quad \text{and} \quad ^{C}I_2 = \begin{pmatrix} I_{xx2} & 0 & 0 \\ 0 & I_{yy2} & 0 \\ 0 & 0 & I_{zz2} \end{pmatrix}.$$

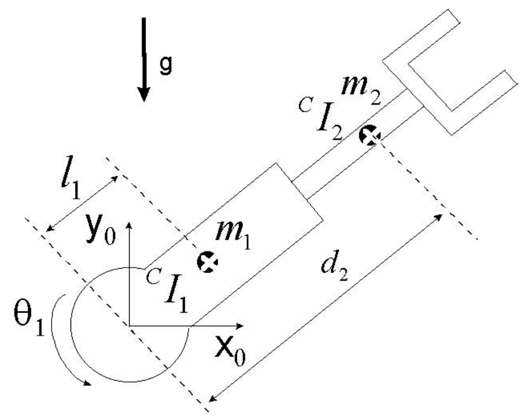

Figure 4.9: 2 DOF RP Manipulator

The Mass Matrix M The mass matrix M can be obtained by applying equation 4.51 to this 2 DOF manipulator:

$$M = m_1 J_{v1}^T J_{v1} + J_{\omega 1}^T {}^{C}I_1 J_{\omega 1} + m_2 J_{v2}^T J_{v2} + J_{\omega 2}^T {}^{C}I_2 J_{\omega 2}.$$

J_{v1} and J_{v2} are obtained by direct differentiation of the vectors:

$$^{0}\mathbf{p}_{C1} = \begin{bmatrix} l_1 C1 \\ l_1 S1 \end{bmatrix}; \quad \text{and} \quad ^{0}\mathbf{p}_{C2} = \begin{bmatrix} d_2 C1 \\ d_2 S1 \end{bmatrix}.$$

In frame $\{0\}$, these matrices are:

$$^{0}J_{v1} = \begin{bmatrix} -l_1 S1 & 0 \\ l_1 C1 & 0 \end{bmatrix}; \qquad ^{0}J_{v2} = \begin{bmatrix} -d_2 S1 & C1 \\ d_2 C1 & S1 \end{bmatrix}.$$

This yields

$$m_1({}^0J_{v1}^T\,{}^0J_{v1}) = \begin{bmatrix} m_1 l_1^2 & 0 \\ 0 & 0 \end{bmatrix}; \qquad (m_2\,{}^0J_{v2}^T\,{}^0J_{v2}) = \begin{bmatrix} m_2 d_2^2 & 0 \\ 0 & m_2 \end{bmatrix}.$$

The matrices $J_{\omega 1}$ and $J_{\omega 2}$ are given by

$$J_{\omega 1} = [\,\bar{\epsilon}_1 \mathbf{z}_1 \quad \mathbf{0}\,] = \quad \text{and} \quad J_{\omega 2} = [\,\bar{\epsilon}_1 \mathbf{z}_1 \quad \bar{\epsilon}_2 \mathbf{z}_2\,].$$

Joint 1 is revolute and joint 2 is prismatic. In frame $\{0\}$, these matrices are:

$${}^0J_{\omega 1} = {}^0J_{\omega 2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

and

$$({}^0J_{\omega 1}^T\,{}^C I_1\,{}^0J_{\omega 1}) = \begin{bmatrix} I_{zz1} & 0 \\ 0 & 0 \end{bmatrix}; \qquad ({}^0J_{\omega 2}^T\,{}^C I_2\,{}^0J_{\omega 2}) = \begin{bmatrix} I_{zz2} & 0 \\ 0 & 0 \end{bmatrix}.$$

Finally, the matrix M is

$$M = \begin{bmatrix} m_1 l_1^2 + I_{zz1} + m_2 d_2^2 + I_{zz2} & 0 \\ 0 & m_2 \end{bmatrix}.$$

Centrifugal and Coriolis Vector v The Christoffel Symbols are defined as

$$b_{i,jk} = \frac{1}{2}(m_{ijk} + m_{ikj} - m_{jki}); \text{ where } m_{ijk} = \frac{\partial m_{ij}}{\partial q_k}; \text{ with } b_{iii} = b_{iji} = 0.$$

For this manipulator, only m_{11} (see matrix M) is configuration dependent – a function of d_2. This implies that only m_{112} is non-zero,

$$m_{112} = 2 m_2 d_2.$$

Matrix B

$$B = \begin{bmatrix} 2b_{112} \\ 0 \end{bmatrix} = \begin{bmatrix} 2m_2 d_2 \\ 0 \end{bmatrix}.$$

Matrix C

$$C = \begin{bmatrix} 0 & b_{122} \\ b_{211} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -m_2 d_2 & 0 \end{bmatrix}.$$

Vector V

$$V = \begin{bmatrix} 2m_2 d_2 \\ 0 \end{bmatrix} [\,\dot{\theta}_1 \dot{d}_2\,] + \begin{bmatrix} 0 & 0 \\ -m_2 d_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{d}_2^2 \end{bmatrix}.$$

The Gravity Vector g

$$\mathbf{G} = -[J_{v1}^T m_1 \mathbf{g} + J_{v2}^T m_2 \mathbf{g}].$$

In frame $\{0\}$, the gravity vector is

$${}^0G = -\begin{bmatrix} -l_1 S1 & l_1 C1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -m_1 g \end{bmatrix} - \begin{bmatrix} -d_2 S1 & d_2 C1 \\ C1 & S1 \end{bmatrix} \begin{bmatrix} 0 \\ -m_2 g \end{bmatrix};.$$

and

$${}^0G = \begin{bmatrix} (m_1 l_1 + m_2 d_2) g C1 \\ m_2 g S1 \end{bmatrix}.$$

Equations of Motion

$$\begin{bmatrix} m_1 l_1^2 + I_{zz1} + m_2 d_2^2 + I_{zz2} & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} +$$
$$\begin{bmatrix} 2m_2 d_2 \\ 0 \end{bmatrix} [\,\dot{\theta}_1 \dot{d}_2\,] + \begin{bmatrix} 0 & 0 \\ -m_2 d_2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{d}_2^2 \end{bmatrix} + \begin{bmatrix} (m_1 l_1 + m_2 d_2) g c_1 \\ m_2 g s_1 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Example: 2-DOF RR Equations of Motion

The masses of the links are m_1 and m_2. The center of mass for the first link is located on the second joint axis at a distance l_1 from the fixed origin. The distance from the second joint axis to the center of mass of link 2 is denoted by l_2. The inertia tensors of the links are ${}^{C_1}I_1$ and ${}^{C_2}I_2$.

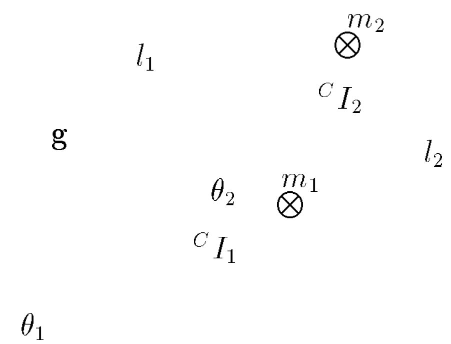

Figure 4.10: 2-DOF RR Equations of Motion

$$ {}^{C_1}I_1 = \begin{pmatrix} I_{xx1} & 0 & 0 \\ 0 & I_{yy1} & 0 \\ 0 & 0 & I_{zz1} \end{pmatrix}; \quad \text{and} \quad {}^{C_2}I_2 = \begin{pmatrix} I_{xx2} & 0 & 0 \\ 0 & I_{yy2} & 0 \\ 0 & 0 & I_{zz2} \end{pmatrix}. $$

Matrix M The mass matrix M is obtained by applying equation 4.51.

$$ M = m_1 J_{v1}^T J_{v1} + J_{\omega 1}^T {}^C I_1 J_{\omega 1} + m_2 J_{v2}^T J_{v2} + J_{\omega 2}^T {}^C I_2 J_{\omega 2}. $$

We compute J_{v_1} and J_{v_2} by direct differentiation of P_{C_1} and P_{C_2}.

$$ {}^0\mathbf{p}_{C_1} = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \end{bmatrix}; \quad \text{and} \quad {}^0\mathbf{p}_{C_2} = \begin{bmatrix} l_1 c_1 + l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} \end{bmatrix}. $$

In frame $\{0\}$, these matrices are:

$$ {}^0J_{v1} = \begin{bmatrix} -l_1 s_1 & 0 \\ l_1 c_1 & 0 \end{bmatrix}; \qquad {}^0J_{v2} = \begin{bmatrix} -l_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}. $$

This yields

$$m_1({}^0J_{v1}^T\,{}^0J_{v1}) = \begin{bmatrix} m_1 l_1^2 & 0 \\ 0 & 0 \end{bmatrix}; \quad (m_2\,{}^0J_{v2}^T\,{}^0J_{v2}) = \begin{bmatrix} m_2(l_1^2+l_2^2+2l_1l_2c_2) & m_2(l_2^2+l_1l_2c_2) \\ m_2(l_2^2+l_1l_2c_2) & m_2l_2^2 \end{bmatrix}.$$

The matrices $J_{\omega 1}$ and $J_{\omega 2}$ are given by

$$J_{\omega 1} = [\,\bar{\epsilon}_1\mathbf{z}_1 \quad \mathbf{0}\,] = \quad \text{and} \quad J_{\omega 2} = [\,\bar{\epsilon}_1\mathbf{z}_1 \quad \bar{\epsilon}_2\mathbf{z}_2\,].$$

Both joints are revolute. In frame $\{0\}$, these matrices are:

$${}^0J_{\omega 1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}; \quad {}^0J_{\omega 2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix};$$

and

$$({}^0J_{\omega 1}^T\,{}^CI_1\,{}^0J_{\omega 1}) = \begin{bmatrix} I_{zz1} & 0 \\ 0 & 0 \end{bmatrix}; \quad ({}^0J_{\omega 2}^T\,{}^CI_2\,{}^0J_{\omega 2}) = \begin{bmatrix} I_{zz2} & I_{zz2} \\ I_{zz2} & I_{zz2} \end{bmatrix}.$$

Finally, the matrix M is

$$M = \begin{bmatrix} m_1l_1^2 + I_{zz1} + m_2(l_1^2+l_2^2+2l_1l_2C2) + I_{zz2} & m_2(l_2^2+l_1l_2C2) + I_{zz2} \\ m_2(l_2^2+l_1l_2C2) + I_{zz2} & l_2^2m_2 + I_{zz2} \end{bmatrix}.$$

Centrifugal and Coriolis Vector v The Christoffel Symbols are defined as

$$b_{i,jk} = \frac{1}{2}(m_{ijk}+m_{ikj}-m_{jki}); \text{ where } m_{ijk} = \frac{\partial m_{ij}}{\partial q_k}; \text{ with } b_{iii} = b_{iji} = 0.$$

<u>Matrix B</u>

$$B = \begin{bmatrix} 2b_{112} \\ 0 \end{bmatrix}; \quad b_{112} = \frac{1}{2}m_{112};$$

$$B = \begin{bmatrix} -2l_1l_2m_2S2 \\ 0 \end{bmatrix}.$$

Matrix C

$$C = \begin{bmatrix} 0 & b_{122} \\ b_{211} & 0 \end{bmatrix};\ b_{211} = \frac{1}{2}(-m_{112});\quad \text{and} b_{122} = m_{122};$$

$$C = \begin{bmatrix} 0 & -l_1 l_2 m_2 S2 \\ l_1 l_2 m_2 S2 & 0 \end{bmatrix}.$$

Vector v

$$V = \begin{bmatrix} -2l_1 l_2 m_2 S2 \\ 0 \end{bmatrix} [\,\dot{\theta}_1 \dot{\theta}_2\,] + \begin{bmatrix} 0 & -l_1 l_2 m_2 S2 \\ l_1 l_2 m_2 S2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix}.$$

The Gravity Vector g

$$\mathbf{G} = -[J_{v1}^T m_1 \mathbf{g} + J_{v2}^T m_2 \mathbf{g}].$$

In frame $\{0\}$, the gravity vector is

$${}^0G = -\begin{bmatrix} -l_1 S1 & l_1 C1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -m_1 g \end{bmatrix} - \begin{bmatrix} -l_1 S1 - l_2 S12 & l_1 C1 + l_2 C12 \\ -l_2 S12 & l_2 C12 \end{bmatrix} \begin{bmatrix} 0 \\ -m_2 g \end{bmatrix};.$$

and

$${}^0G = \begin{bmatrix} [(m_1 + m_2) l_1 C1 + m_2 l_2 C12] g \\ m_2 l_2 C12 g \end{bmatrix}.$$

Equations of Motion

$$\begin{bmatrix} m_1 l_1^2 + I_{zz1} + m_2(l_1^2 + l_2^2 + 2l_1 l_2 C2) + I_{zz2} & m_2(l_2^2 + l_1 l_2 C2) + I_{zz2} \\ m_2(l_2^2 + l_1 l_2 C2) + I_{zz2} & l_2^2 m_2 + I_{zz2} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix}$$

$$+ \begin{bmatrix} -2l_1 l_2 m_2 S2 \\ 0 \end{bmatrix} [\,\dot{\theta}_1 \dot{\theta}_2\,] + \begin{bmatrix} 0 & -l_1 l_2 m_2 S2 \\ l_1 l_2 m_2 S2 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} +$$

$$\begin{bmatrix} [(m_1 + m_2) l_1 C1 + m_2 l_2 C12] g \\ m_2 l_2 C12 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}.$$

Chapter 5

Trajectory Generation

5.1 Introduction

The majority of robots in industry perform pick and place operations. That amounts to positioning a manipulator at a certain point and orientation, grasping an object, moving it over to some other position and orientation, and ungrasping it. In that motion, the end-effector of the manipulator traverses some trajectory in such a way so that the rest of the structure of the manipulator does not collide with the objects in the workspace.

We can consider several variations of this problem. If we do not know anything about the environment, we can design it in such a way that the robot can move in it and perform its tasks as fast as possible. If the environment is already given (industrial workstation for example) but we have the freedom to design the robot, we might want to do it in an optimal manner with respect to some workspace requirements. Finally, if both the environment and the robot are given (e.g. a PUMA robot in an industrial workcell) we would like to calculate the best trajectory for the robot's end-effector to follow, in order to perform the tasks in question. In this Chapter we will consider this third problem.

In that framework our goal is: Move the manipulator arm from some initial position (frame $\{T_A\}$) to some desired final position (frame $\{T_C\}$)

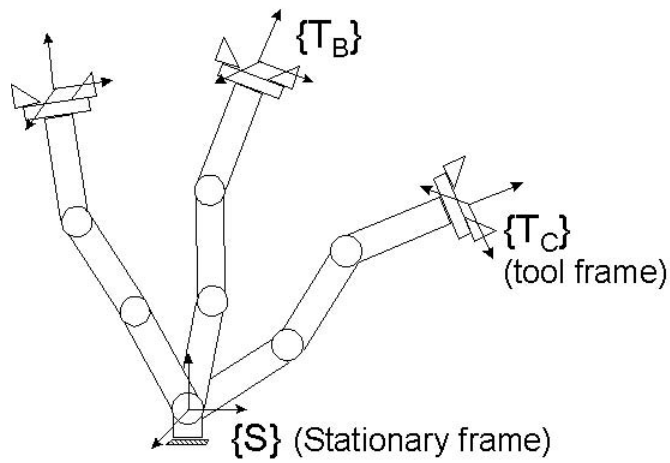

Figure 5.1: Path points for a manipulator

going through some intermediate via points (e.g. frame $\{T_B\}$). The frames shown in Figure 5.1 represent the position and orientation of the end-effector with respect to some fixed reference frame in the environment (for example the workcell for the robot). In terms of terminology we will call these points (initial, via and final) **path points**.

We will find a trajectory for the robot's end-effector to move through these path points in the desired order. The trajectory is the time history of the position, velocity and acceleration for the end-effector (in Cartesian space) or for each degree of freedom of the robot (in Joint space).

During its motion the manipulator will be subjected to different constraints. Those constraints can be spatial (the manipulator should be within the workspace and should not collide with obstacles in the workspace), time (certain operations or motions might need to be performed within certain time frame - particularly for industrial robots), smoothness (we do not want discontinuous motions, particularly if we are performing insertion tasks where the objects that are manipulated can be damaged). All these constraints can be expressed mathematically in terms of the positions, velocities and accelerations of the end-effector considered.

5.2 Joint Space vs. Cartesian Space planning

The planning of the motion of the manipulator can be performed in Cartesian or in Joint space. In either of these spaces we can calculate the necessary trajectories that achieve the required motion. Working in each space has its advantages and disadvantages.

Consider first **Joint space**. Since we control the torques, any particular position in the workspace can be accessed. Thus we can go exactly through the intermediate path points. However if we want to follow a particular trajectory or track a shape in the workspace, this is not that easy in joint space. There is no guarrantee that we will be able to solve the inverse kinematics for all points along the required trajectory uniquely and continuously.

In **Cartesian space**, on the other side, we can track shapes exactly (for example we can follow a straight line). However in order to actually command torques to the joints to achieve this motion along the straight line, we need to solve the inverse kinematics at all points along the trajectory. This needs to be done at every update at the servo rate which can be very computationally expensive. In joint space there are generally less calculations.

Another difference between the two spaces is with respect to singularities. In joint space we do not have problems with singularities because we are actually planning in the space of the mechanism and all that needs to be done is to solve the forward kinematics (which has an unique solution). In Cartesian space, the trajectory that are plannned might pass through path points corresponding to singularities in joint space. That will make the motion impossible or in the best case very expensive.

We will illustrate this discussion with a few examples. In Figure 5.2 the task is to move from the initial point to the goal point along a straight line with a 2D two revolute link manipulator (RR). The two links are of different lengths and the workspace is the donut inside the large circle and outside the small one. Clearly the straight line trajectory

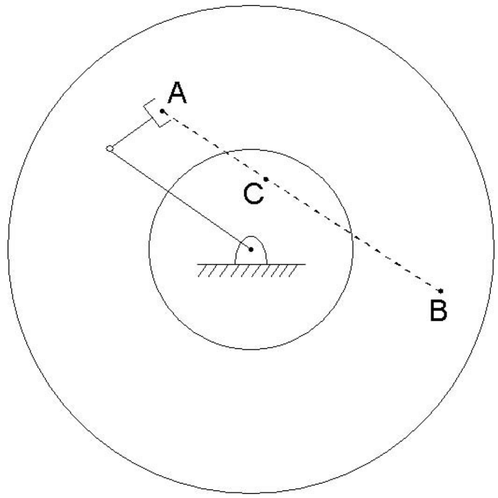

Figure 5.2: Unreachable intermediate points

passes through a region outside of the workspace of the manipulator. Thus even though the initial and final path points are reachable by the manipulator, some of the intermediate path points are not. Here the joint limits of the manipulator create the problem.

In the next example in Figure 5.3 our task is the same, but this time both links of the manipulator are of the same length. Thus the workspace is the entire inside of the circle and all points along the straight line path are reachable by the manipulator. However we can not perform that straight line motion continuosly because toward the middle of the trajectory we are approaching a singularity. At the singularity the resulting velocity is infinitely high, thus it is virtually impossible to control the robot and follow a particular trajectory. Here the problem is due to singularities of the mechanism.

In the third example in Figure 5.4 all the points along the path are reachable, the singularity can be dealt with, but the robot still can not perform the straight line motion. The reason is that the initial and the goal path points are reachable in different joint space solutions. Thus we can not continuosly move from one point to the other. Some of the intermediate points along the path are reachable from below and the only way to move continuosly from the start to the goal is from below

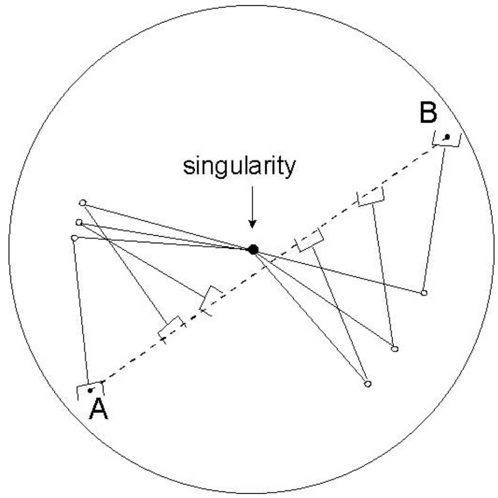

Figure 5.3: Approaching singularities of the manipulator

in which case the possible trajectory is not a straight line.

In order to perform the path planning in practice we need to account for all of these problems. Often the planning is done in a hybrid fashion, where part of the plan is done in Cartesian space and part in joint space, in order to avoid some of the problems outlined.

For generality of the discussion for the rest of the chapter we will consider a generic parameter u and do the actual planning for that parameter u. We have to remember that if we are planning in joint space we can substitute for u any of the generalized coordinates q_i. In fact we need to plan for each and every one of them in turn and then put the entire trajectory together in multidimensional vectors.

If we do the planning in Cartesian space, we can substitute for u the parameters we use for the position or orientation representation. If we use Cartesian position we will substitute x, y and z for u one by one. For spherical coordinates we will substitute ρ, θ and ϕ, for cylindrical - ρ, d and θ. For orientation trajectories we can use the Euler or fixed angles α, β and γ instead of u, or Euler parameters, etc.

Independent of which space and which representation we use, the actual trajectories are just mathematical curves and we will describe them

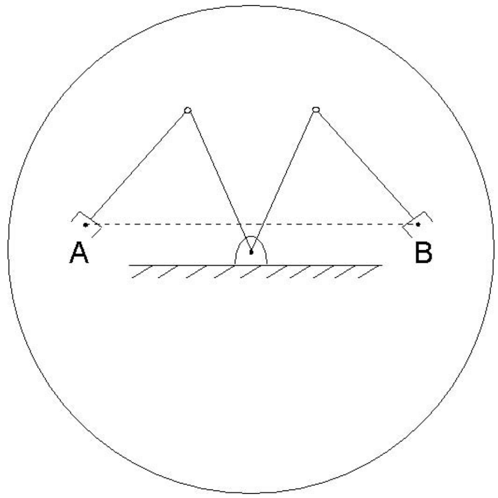

Figure 5.4: Different joint space solutions

here as functions of the generic variable u. These trajectories are actually multi-parameter ones (say $q_1, q_2, \ldots, q_n$ in joint space) but we will assume that all parameters are independent and plan for each of them as functions of one scalar variable. In practice we typically choose the joint or the Cartesian variables to be independent, thus this is a reasonable approximation.

5.3 Path Planning with Polynomial Trajectories

Having performed these abstractions, the problem we face is to fit a curve that starts from a given point u_0, goes through some intermediate points $(u_1, u_2, \ldots)$ and ends in some goal point u_n subject to some constraints. The constraints are on the position (the curve itself), the velocity (the first derivative of the curve) or the acceleration (the second derivative of the curve). We can use a variety of curves, starting with the simplest one - a straight line- and going to second, third or higher degree polynomials if necessary. The advantage of using polynomials is that they, as well as their derivatives, are linear with respect to the coefficients in front of the function variable.

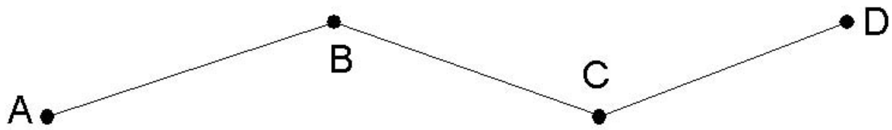

Figure 5.5: Straight lines trajectory

Let us consider the simplest polynomial that can result in a reasonable analysis - a straight line. If we have a trajectory going through some intermediate points, we can connect all path points with straight line segments in the corresponding order as in Figure 5.5. However there is a potential problem when using straight segments. Usually the intermediate points are there so that the robot avoids obstacles and we want to be able to move along the path through these intermediate points in a continuous fashion. That requirement can not be guarranteed by straight line trajectory.

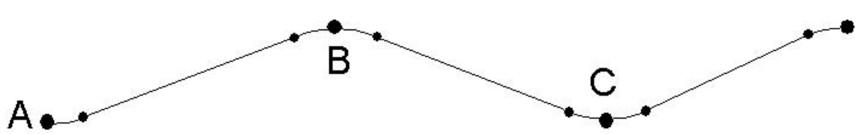

Figure 5.6: Straight lines with blends trajectory

We can take care of this problem by introducing blend regions around each via point so that the trajectory is smooth and differentiable as depicted in Figure 5.6. The simplest possible blends that we can use are parabolic blends. They depend on one parameter only. The motion starts with half a blend that blends into a straight line segment and then blends around the first via point, transfers into a straight line, etc. until the last path point at which the path blends down to zero velocity and acceleration. A potential problem with this approach is that as we will see, the formulas for calculating the velocities and accelerations are quite involved and we have to constantly keep track of the type of region we are moving through and use the corresponding formulas.

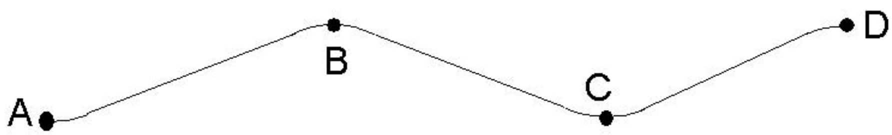

Figure 5.7: Higher order polynomials trajectory

In the curve depicted in Figure 5.7 we are using higher order polynomials, in particular cubics. Here the curve itself has higher degree of smoothness and the given cubic can seemlessly blend into the next cubic on the chain without the need of specific blend curves. In this trajectory all curves are of second degree and it is relatively easy to deal with them.

If we need to satisfy more constraints (e.g. on accelerations and velocities) we might need to use higher degree polynomials or perhaps trigonometric or transcendental curves. In what follows we will develop the formulas for the most common cases outlined so far.

Let us start with a cubic spline. The general formula for a cubic spline is

$$u(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{5.1}$$

with its derivative and acceleration being respectively:

$$\dot{u}(t) = a_1 + 2a_2 t + 3a_3 t^2 \tag{5.2}$$

and

$$\ddot{u}(t) = 2a_2 + 6a_3 t \tag{5.3}$$

In these formulas u is the generic parameter of the functions, the argument t is the time, and a_0, a_1, a_2 and a_3 are the unknown coefficients that define the trajectory.

The constraints on the trajectory expressed in mathematical form will define the equations that we will use to calculate the coefficients. The most imporatnt type of constraints are the initial conditions for the trajectory. In particular we might be given the initial and final positions for the function, e.g. $f(0) = u_0$ and $f(t_f) = u_f$. Graphically this general cubic function is depicted in Figure 5.8.

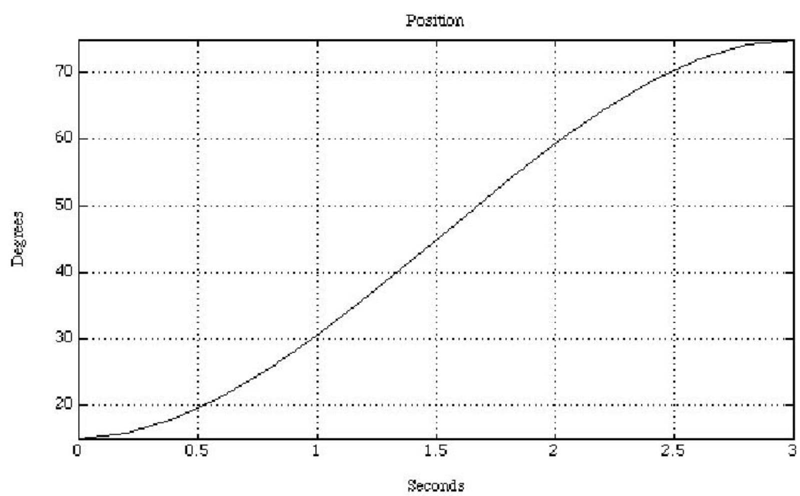

Figure 5.8: A cubic polynomial

If we assume that the manipulator starts from rest and ends at rest, this is equivalent to writing $\dot{f}(0) = 0$ and $\dot{f}(t_f) = 0$. This velocity is graphically depicted in Figure 5.9.

Finally the acceleration is the straight line in Figure 5.10 (the second derivative of the function).

If we need to satisfy the four conditions above (two for the position and two for the velocity) they can be written as the following linear system:

$$a_0 = u_0 \tag{5.4}$$

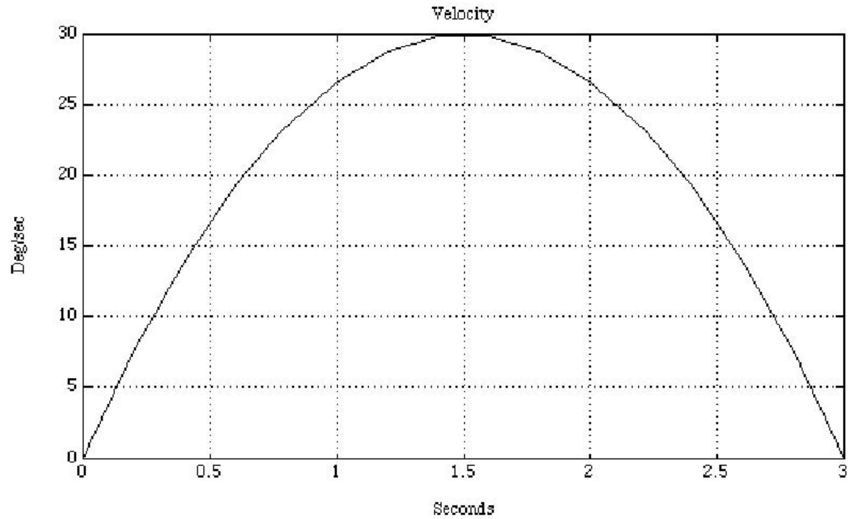

Figure 5.9: The derivative of a cubic polynomial

$$a_0 + a_1 t_f + a_2 {t_f}^2 + a_3 {t_f}^3 = u_f \tag{5.5}$$

$$a_1 = 0 \tag{5.6}$$

$$a_1 + 2a_2 t_f + 3a_3 {t_f}^2 = 0 \tag{5.7}$$

The system has four equations and four unknowns (the parameters a_0, a_1, a_2 and a_3). Since we are using polynomials, this is a linear system of equations. We can solve for the parameters and that will determine uniquely the function f. Clearly the acceleration in that case will be predetermined by those just found parameters and would be beyond our control, namely

$$\dddot{u(t)} = 6a_3 \tag{5.8}$$

The linear system above is a very simple one to solve (we do not even have to build the entire 4×4 matrix and solve it). The solution is the following

$$a_0 = u_0 \tag{5.9}$$

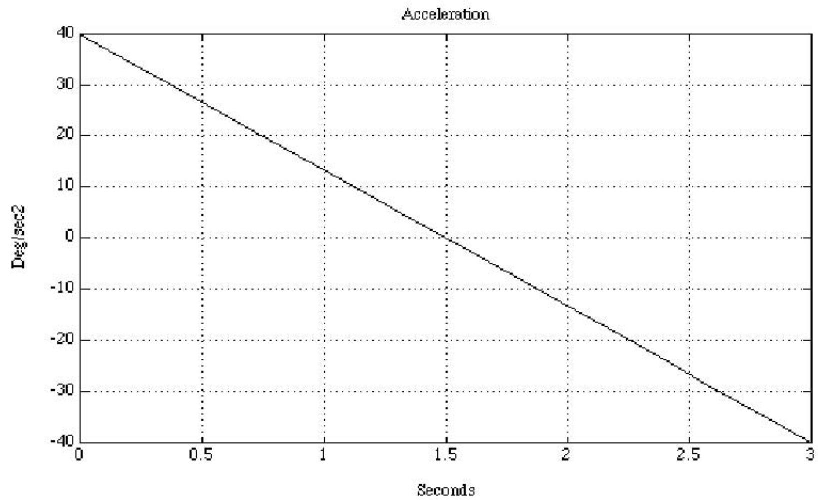

Figure 5.10: The second derivative of a cubic polynomial

$$a_1 = 0 \tag{5.10}$$
$$a_2 = (3/{t_f}^2)(u_f - u_0) \tag{5.11}$$
$$a_3 = (-2/{t_f}^3)(u_f - u_0) \tag{5.12}$$

and the resulting function f is:

$$u(t) = u_0 + \frac{3}{t_f^2}(u_f - u_0)t^2 + (-\frac{2}{t_f^3})(u_f - u_0)t^3 \tag{5.13}$$

In the general case for n equations and n unknowns we will build an $n \times n$ linear system. If the determinant of the linear system is zero, that means that the conditions we are trying to solve are not independant and we can satisfy additional conditions. That also means that we can control all these conditions independently.

5.4 Planning with Intermediate Points

If the targeted motion requires the manipulator to go through some intermediate points, several cubic splines need to be jointed together at via points. If we were to stop (i.e have velocity zero) at each via point, we can directly use the formulas derived above. However in general a smooth motion requires continuous velocities, i.e. the velocity at the end of each segment is the same (non-zero) velocity as the one at the beginning of the next segment. In that case the two conditions for the velocity will be:

$$\dot{u}(0) = \dot{u}_0 \tag{5.14}$$
$$\dot{u}(t_f) = \dot{u}_f \tag{5.15}$$

The new linear system of equations is

$$a_0 = u_0 \tag{5.16}$$
$$a_0 + a_1 t_f + a_2 (t_f)^2 + a_3 (t_f)^3 = u_f \tag{5.17}$$
$$a_1 = \dot{u}_0 \tag{5.18}$$
$$a_1 + 2a_2 t_f + 3a_3 (t_f)^2 = \dot{u}_f \tag{5.19}$$

and its solution is

$$a_0 = u_0 \tag{5.20}$$
$$a_1 = \dot{u}_0 \tag{5.21}$$
$$a_2 = \frac{3}{t_f^2}(u_f - u_0) - \frac{2}{t_f}\dot{u}_0 - \frac{1}{t_f}\dot{u}_f \tag{5.22}$$
$$a_3 = -\frac{2}{t_f^3}(u_f - u_0) + \frac{1}{t_f^2}(\dot{u}_f + \dot{u}_0) \tag{5.23}$$

These formulas are slightly more complicated than the single cubic case.

The values that are used above for the initial conditions can be derived from different requirements. The positions come from the workspace of the manipulator (the Cartesian coordinates of the points on the path). The velocities can be due to the linear and angular velocities of the manipulator (using the Jacobian for finding the joint velocities). Alternatively we can let the system choose some reasonable velocities based on heuristic or deterministic considerations. We can also choose the velocities' values to achieve continuous velocities ($\dot{u}_1(t_f) = \dot{u}_2(0)$) or accelerations ($\ddot{u}_1(t_f) = \ddot{u}_2(0)$).

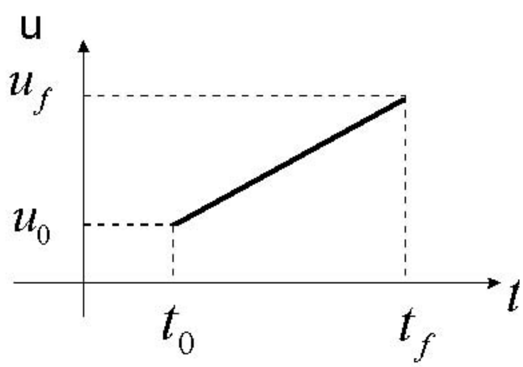

Figure 5.11: A straight line segment

5.5 Straight paths with Parabolic blends

The next trajectory segments that we will consider are the linear ones. The general equation in this case is

$$u(t) = a_0 + a_1 t \tag{5.24}$$

where the motion starts at time t_0, ends at time t_f with corresponding values for $u : u_0$ and u_f as depicted in Figure 5.11.

$$u(t_0) = u_0 \tag{5.25}$$
$$u(t_f) = u_f \tag{5.26}$$

There are only two parameters a_0 and a_1 that define a single line segment. Thus we can satisfy at most two constraints, in particular the position in the beginning and at the end. The result is a linear system of two equations with two unknowns). Obviously we have no control over the velocity and that is why it is not appropriate to chain line segments for trajectory through intermediate path points. The velocity is discontinuous and constant for each segment. To deal with this problem we introduce parabolic blends between the line segments shown in Figure 5.12.

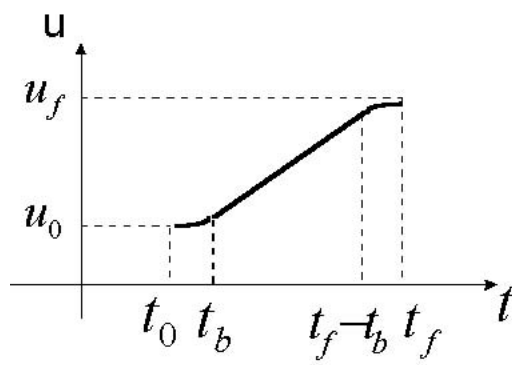

Figure 5.12: Linear segment with a parabolic blend

The general equation for a parabolic blend is

$$u(t) = \frac{1}{2}at^2 \tag{5.27}$$

where a is the parameter that we can use to satisfy constraints. The velocity in the blend is a linear function of time ($\dot{u}(t) = at$), while the acceleration is the constant a ($\ddot{u}(t) = a$). Thus during a parabolic blend $u(t) = \frac{1}{2}\ddot{u}t^2$. To smooth the motion in the beginning and at the end of a linear segment, we add a parabolic blend (half blends per segment) at both ends. If we denote with t_b the time when the first blend ends and t_f as the time at the end of that segment, we can derive:

$$t_b = \frac{t}{2} - \frac{\sqrt{\ddot{u}t^2 - 4\ddot{u}(u_f - u_0)}}{2\ddot{u}} \tag{5.28}$$

where

$$t = t_f - t_0 \tag{5.29}$$

is the desired duration of the motion.

In other words, if we know the acceleration we can define the blend region, as well as the time for the blend. These are all the parameters that are necessary for a motion described by a straight segment with parabolic blends.

We can now put several of these trajectory segments together. The goal is to move from the initial path point to the final path point via several intermediate points using straight segments and parabolic blends. The actual motion will start with a half blend that turns into a line segment until we reach close to the first intermediate point. At that point we include a full blend that connects the line segment between initial point and first via point, and the line segment between the first and the second via path points. The trajectory continues similarly as illustrated in Figure 5.13. The last line segment connects via a half blend to the goal path point for a smooth and continuous motion.

As denoted in the figure, there are several time parameters: t_{dij} is the duration for the motion from via point i to via point j, t_i is the time for the blend region around path point i and t_{ij} is the time for the linear segment between path points i and j. The parameters include the velocities $\dot{u_{ij}}$ for the part between path points i and j as well as the acceleration $\ddot{u}_i$ of the blend region for point i. The following quantities can be considered as given in the problem:

- the positions of the path points along the trajectory u_i, u_j, u_k, u_l, u_m
- the desired time durations for each segment $t_{dij}, t_{djk}, t_{dkl}, t_{dlm}$;

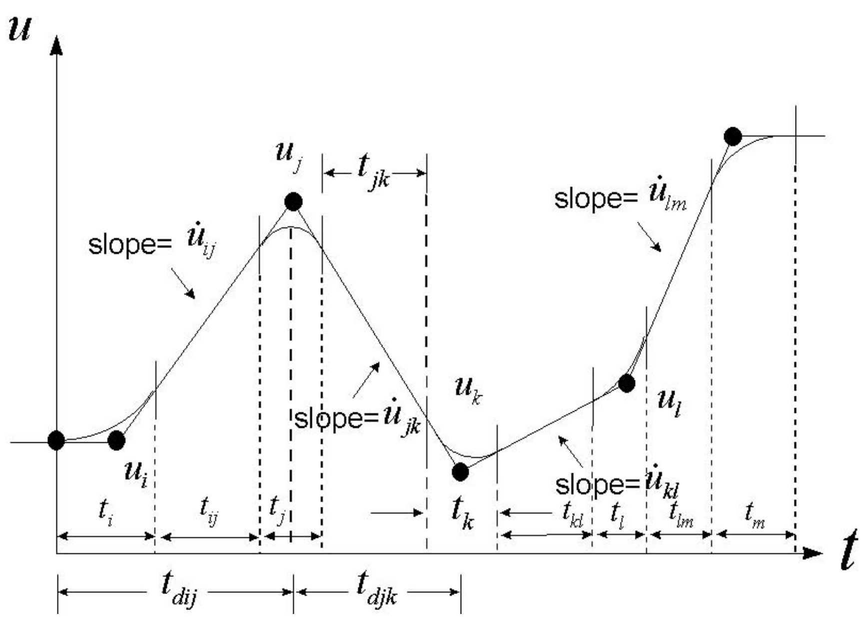

Figure 5.13: Linear interpolation with parabolic blends

- in some cases we also have some information about the magnitude of the accelerations in the blends $|\ddot{u}_i|, |\ddot{u}_j|, |\ddot{u}_k|, |\ddot{u}_l|$.

What needs to be computed is (in this order):

- the blend times at each path point t_i, t_j, t_k, t_l, t_m;
- the straight segments times $t_{ij}, t_{jk}, t_{kl}, t_{lm}$;
- the velocities of each segment $\dot{u}_{ij}, \dot{u}_{jk}, \dot{u}_{kl}, \dot{u}_{lm}$;
- the signs of the accelerations (are we accelerating or deccelerating) at each via point.

From the parameters that need to be given, the magnitudes of the accelerations are usually determined by the system and the configuration of the robot. Using them, as well as the information about the problems that need to be solved, we can compute the desired time durations t_{dij}.

The parameters above allow us to build the entire trajectory by computing all necessary entities by the formulas below. For the first segment the formulas are:

$$\ddot{u}_1 = sign(u_2 - u_1)\,|\ddot{u}_1| \tag{5.30}$$

$$t_1 = t_{d12} - \sqrt{t_{d12}^2 - \frac{2(u_2 - u_1)}{\ddot{u}_1}} \tag{5.31}$$

$$\dot{u}_{12} = \frac{u_2 - u_1}{t_{d12} - \frac{1}{2}t_1} \tag{5.32}$$

$$t_{12} = t_{d12} - t_1 - \tfrac{1}{2}t_2 \tag{5.33}$$

The inside segments can be computed using:

$$\dot{u}_{jk} = \frac{u_k - u_j}{t_{djk}} \tag{5.34}$$

$$\ddot{u}_k = sign(\dot{u}_{kl} - \dot{u}_{jk})\,|\ddot{u}_k| \tag{5.35}$$

$$t_k = \frac{\dot{u}_{kl} - \dot{u}_{jk}}{\ddot{u}_k} \tag{5.36}$$

$$t_{jk} = t_{djk} - \frac{1}{2}t_j - \frac{1}{2}t_k \tag{5.37}$$

The last segment yields:

$$\ddot{u}_n = sign(u_{n-1} - u_n)\,|\ddot{u}_n| \tag{5.38}$$

$$t_n = t_{d(n-1)n} - \sqrt{t_{d(n-1)n}^2 + \frac{2(u_n - u_{n-1})}{\ddot{u}_n}} \tag{5.39}$$

$$\dot{u}_{(n-1)n} = \frac{u_n - u_{n-1}}{t_{d(n-1)n} - \frac{1}{2}t_n} \tag{5.40}$$

$$t_{(n-1)n} = t_{d(n-1)n} - t_n - \frac{1}{2}t_{n-1} \tag{5.41}$$

Please note that the initial and the final position sets of formulas are slightly different than the others. They reflect the different treatment that the first and the last half blends have. The formulas above are derived in the Appendix. The formulas are arranged in the order needed to calculate the motion.

5.6 Generalized Path planning

There are a few interesting features of the computed trajectory.

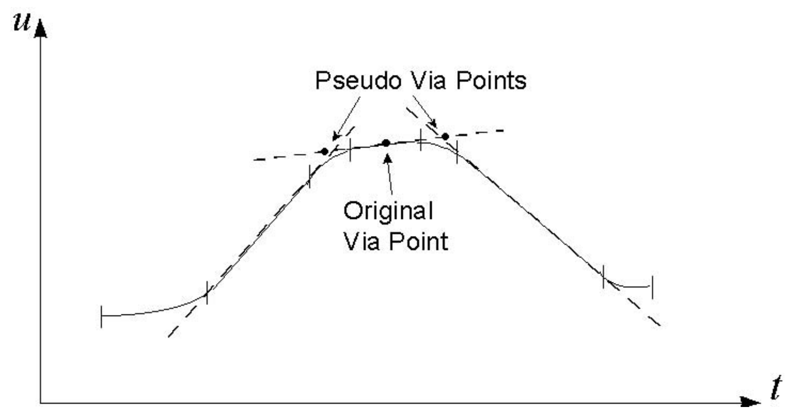

Figure 5.14: "Pseudo" via points in the trajectory

As seen in Figure 5.14, this trajectory does not pass exactly through the via points because of the blends. If we do need to pass through these via points, there are several possible approaches. We can double the intermediate points and thus force the trajectory to go through them (it will create a straight segment through the points) . We can also introduce two artificial path points close by on different sides of the via points infinitesimally close to them. Finally we can use sufficiently high acceleration to go through the actual via points exactly.

In some applications there is a need to satisfy more conditions for a single segment. In that case we will need more parameters and thus higher degree polynomials. For example if two values are used for the position, two for the velocities and two for the accelerations, we will need a polynomial of degree 5 (quintic polynomial $u(t) = a_0 + a_1 t +$

$a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5$). We can also fit algebraic series, sine and cosine waves and other higher-dimensional functions.

The discussion so far in the chapter was for functions $f(u)$ of one variable. For joint or Cartesian space there are 6 to 12 parameters that describe the complete position and orientation of the end-effector or the joint variables. The planning needs to be done for each of the parameters individually and then integrated in the general plan for manipulation.

The trajectories (defined in terms of the path parametres u, $\dot{u}$ and $\ddot{u}$) often need to be computed at run-time. If we are working in joint space directly, the calculations are similar to above. In Cartesian space, as mentioned earlier in the chapter, we need to solve the inverse kniematics quite often while moving along. At each update we will solve for the sequence of the parameters at a frequency equivalent to the ones commonly used.

In real world applications there are a number of objects in the workspace of the manipulator, which for consistancy we will call obstacles (they can be moving or fixed obstacles , parts of the environment, etc.). The plan for the motion of the manipulator needs to go around these obstacles in an optimal way. This area of research is called "Motion Planning" and is extensively discussed in a variety of publications. We will refer the reader to some of these books. That planning needs to consider gross (for the main structure) and fine motion planning (for the end-effctor movement). We can also talk about global (in the whole space) vs. local (for a mini- manipulator at the end of the structure) planning. Other techniques are using configuration space or potential field approach. C-space (or configuration space) obstacles are the obstacles "grown" by the dimensions of the moving robot, so that the robot is represented by a point.

As mentioned in the beginning of this chapter, we can extend our consideration to include the cases when we are trying to design a robot that is optimized for a particular environment, or create both the environment and the robot simultaneously. Interesting situations include cases where more than one robot share a workspace, or there are non-holonomic constraints for the robot motion. For more discussions on these topics please refer to the bibliography.

Chapter 6

Manipulator Control

6.1 Introduction

In this chapter, we first review the basics of PID controllers. Next, we present the general control structure for dynamic decoupling and control of joint motions. Finally, we present the basics of the task-oriented operational space control, which provides dynamic decoupling and direct control of end-effector motions.

Let us consider the task of controlling the motion of an n-DOF manipulator for some goal configuration defined by a set of desired joint positions. This task can be accomplished by selecting n independent *proportional-derivative*, PD, controllers that affect each joint to move from its current position to the goal position. Each of these controllers can be viewed as a spring-damper system attached to the joint. The spring's neutral position corresponds to the goal position of the joint, as illustrated in Figure 6.1. Any disturbance from that position would result in a restoring force that moves the joint back to its goal position. During motion, the damper contributes to the stability of the system.

These simple controllers are widely used in industrial robots to execute point-to-point motion tasks. However, these controllers are limited in their ability to perform motion tracking or any task that involves interaction with the environment. In motion, the manipulator is subjected

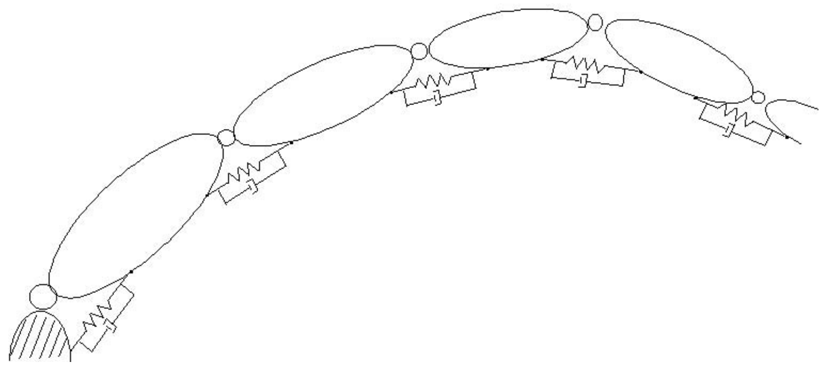

Figure 6.1: Manipulator with springs and dampers

to the dynamic forces acting on its links. By ignoring these forces, PD controllers are limited in their performance for these tasks. The control structures needed to address the dynamics of manipulator systems are presented in section 6.6.2. Our discussion here focuses on independent PD controllers.

When the goal position is specified in terms of the end effector configuration, the manipulator can be directly controlled at the end effector. We could imagine placing a 3D spring-damper system at the end-effector itself, instead of the once we placed at the joints. The end effector will be then attracted to move to its goal position by the stiffness of the spring, and the stability of the motion will be provided by the damper.

A more general way to think about this approach is to imagine the application of a force at the end-effector, as the gradient of some attractive potential field, whose minimum is at the goal position. We will need to add some damping proportional to the velocity to stabilize the system at the goal position. To produce this type of control, it is necessary to be able to create a force at the end-effector, which must be produced by the actuators at the joints. This can be accomplished with the transpose of the Jacobian, which relates forces at the end effector to corresponding torques at the joints. This allows direct control of the effector without requiring any inverse kinematics.

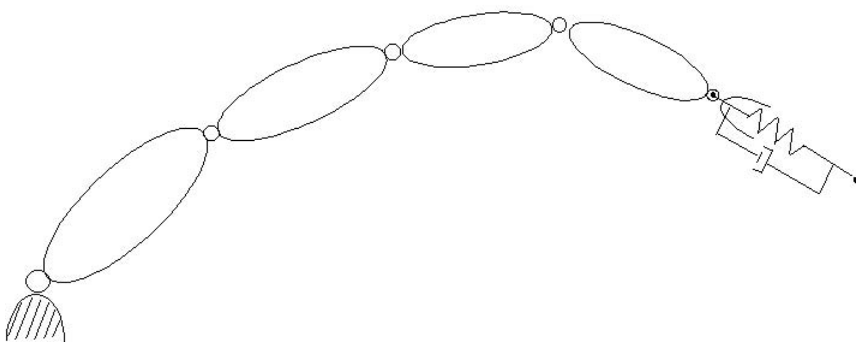

Figure 6.2: Potential Field for a manipulator

The difficulty in practice with such controllers is their limited performance for motion control, since the dynamics of the manipulator are ignored. The incorporation of the dynamics for the end-effector control is discussed in section 6.7.1.

Control with Inverse Kinematics Typically robot control has the following structure

Let $\mathbf{x}_d$ be the desired position and orientation of the end effector. The inverse kinematics are used to find the corresponding desired joint position $\mathbf{q}_d$. This is a vector of desired positions for each joint that is transmitted to a set of independent controllers each of which is trying to minimize the error between the desired joint position and actual joint position. These controllers are simple PD or PID controllers. Because of the computational complexity of inverse kinematics, this approach is difficult to use for tasks involving real-time modifications of the end-effector desired position and orientation.

Control with Linearized Kinematics Another approach to the task transformation problem relies on the linearized kinematics and

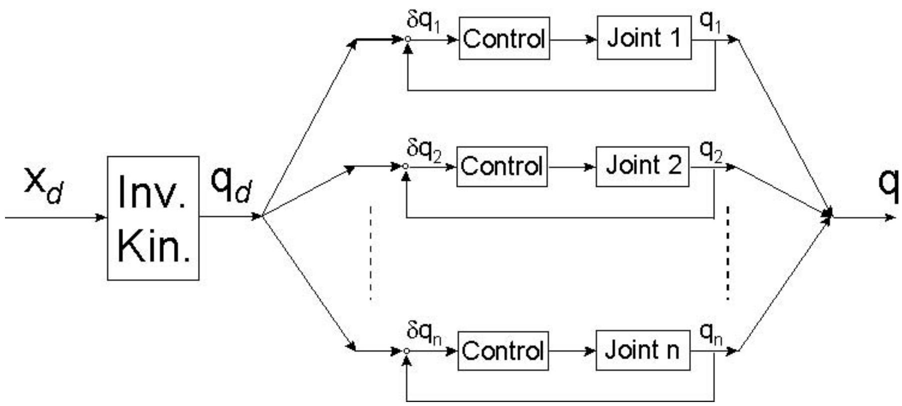

Figure 6.3: Inverse kinematics robot control

the use of the Jacobian and its inverse. This approach called *resolved motion rate control* was first proposed by Whitney in 1972.

To a small joint space displacement $\delta\mathbf{q}$ corresponds a small end-effector displacement $\delta\mathbf{x}$. Given $\delta\mathbf{q}$, the corresponding displacement $\delta\mathbf{x}$ is given by

$$\delta\mathbf{x} = J(\mathbf{q})\delta\mathbf{q} \tag{6.1}$$

When it exists, the inverse of the Jacobian allows us to compute the displacement $\delta\mathbf{q}$ that corresponds to a desired displacement $\delta\mathbf{x}$,

$$\delta\mathbf{q} = J^{-1}(\mathbf{q})\delta\mathbf{x} \tag{6.2}$$

If the manipulator task consisted of following a path of the end effector, this relationship can be used to continuously increment the joint position in accordance with small displacements along the end-effector path. At a given configuration $\mathbf{q}$, the end-effector position and orientation is determined by the forward kinematics, $\mathbf{x} = f(\mathbf{q})$. Selecting a neighboring desired end-effector configuration $\mathbf{x}_d$ results in a small end-effector displacement.

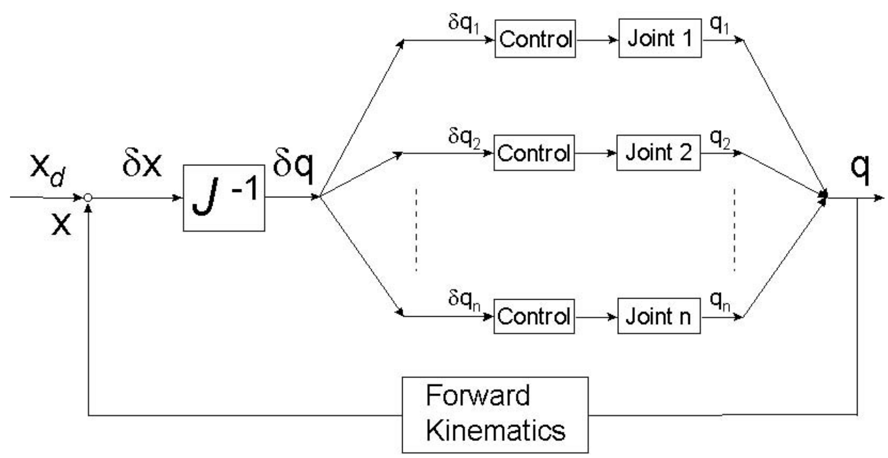

Figure 6.4: Linearized kinematics robot control

$$\delta\mathbf{x} = \mathbf{x}_d - \mathbf{x}$$

Using the inverse of the Jacobian matrix, we can establish the corresponding joint displacement δq

$$\delta\mathbf{q} = J^{-1}(\mathbf{q})\delta\mathbf{x}$$

From the current joint configuration $\mathbf{q}$, this allows to compute the desired joint configuration $\mathbf{q}_d$ as,

$$\mathbf{q}_d = \mathbf{q} + \delta\mathbf{q}$$

In general, the task transformation problem involves, in addition, transformations of the end-effector desired velocities and accelerations into joint descriptions. Task transformations are computationally demanding and are difficult to carry out in real-time.

6.2 Passive Natural Systems

The behavior of a PD controlled mechanism has common characteristics with passive spring-damper systems. In this section, we will consider the mass-spring-damper system shown in Figure 6.5. The study of this natural 1 DOF system will provide the basis for the development of PD controllers.

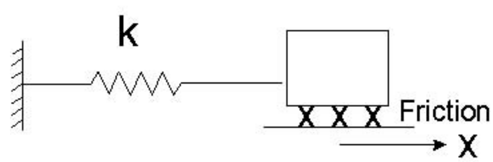

Figure 6.5: Spring-mass system

6.2.1 Conservative Systems

Consider a mass m connected to a spring of stiffness k. The position of the mass is determined by the coordinate x, and the neutral position of the spring is assumed at $x = 0$. The kinetic energy of this system is

$$K = \frac{1}{2} m \dot{x}^2$$

The potential energy of this system is due to the spring. The mass is subjected to the force

$$f = -kx$$

which is the gradient of the spring potential energy

$$V = \frac{1}{2} k x^2$$

The Lagrangian equation for this system is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0 \tag{6.3}$$

This system is conservative, since the only force acting on it is a *conservative* force due to a potential energy. On the right hand side of Lagrange's equation, the external force is zero. The total energy of the system is therefore constant. Thus this system is stable, but oscillatory. The equation of motion is

$$m\ddot{x} + kx = 0 \tag{6.4}$$

We can initially set the potential energy of the system by pulling on the mass. The system will start with zero kinetic energy, its velocity will increase and the potential energy will be transfered into kinetic energy. When the potential energy becomes zero, the kinetic energy will be transferred back to the potential energy.

As illustrated in Figure 6.6, the potential is set to some level. After its release, the mass oscillates between two positions with a frequency that depends on both k and m – higher frequency with higher stiffness and smaller mass. The *natural frequency*, ω_n of this system is

$$\omega_n = \sqrt{\frac{k}{m}} \tag{6.5}$$

The equation of motion can be written in the form

$$\ddot{x} + \omega_n^2 x = 0 \tag{6.6}$$

and the time response, $x(t)$, of this system is

$$x(t) = c\cos(\omega_n t + \phi) \tag{6.7}$$

where c and ϕ are constants depending on the initial conditions.

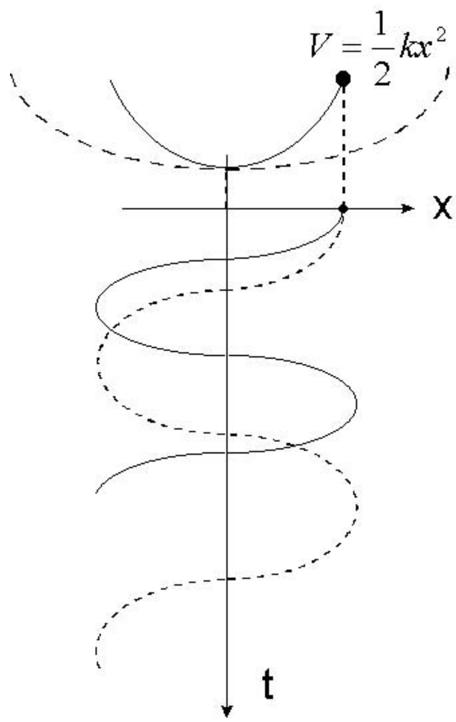

Figure 6.6: System's response

6.2.2 Dissipative Systems

In a real setting there is always some amount of friction acting on a mechanical system. Let us assume that the friction acting on our mass-spring is simply viscous,

$$f_{\text{friction}} = -b\dot{x}$$

With the friction, the Lagrange equation is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = f_{\text{friction}} \tag{6.8}$$

The *dissipative* force f_{friction} appears now as an external force on the right hand side of the equation. This system is dissipative, and is described by the second order equation

$$m\ddot{x} + b\dot{x} + kx = 0 \tag{6.9}$$

To analyze the characteristics of this system, we divide the equation by m. $\sqrt{k/m}$ represents the natural frequency of the system and b/m represents the damping of the system.

$$\ddot{x} + \frac{b}{m}\dot{x} + \frac{k}{m}x = 0 \tag{6.10}$$

The friction results in an *oscillatory-damped* behavior. As the friction coefficient b increases, the the magnitude of oscillations decreases at a faster rate. If b was very large, the system will be *over damped*. It will never cross the zero axis, slowly moving toward the goal position. Between these two states, there is a *critically-damped* behavior of the system. As we will see, this behavior is quite desirable in the development of control systems. The analysis of the time-response of the system shows that the critically-damped state is reached when

$$\frac{b}{m} = 2\omega_n \tag{6.11}$$

Treating the critically-damped state as a reference sate, we introduce a ratio describing the damping b/m with respect to its critical value when $b/m = 2\omega_n$. The *natural damping ratio* is defined as

$$\xi_n = \frac{b}{2\omega_n m} = \frac{b}{2\sqrt{km}} \tag{6.12}$$

The system is critically damped when $\xi = 1$, It's over damped if $\xi > 1$, and oscillatory when $\xi < 1$. With ω_n and ξ_n, the equation of motion is

$$\ddot{x} + 2\xi_n\omega_n\dot{x} + \omega_n^2 x = 0 \tag{6.13}$$

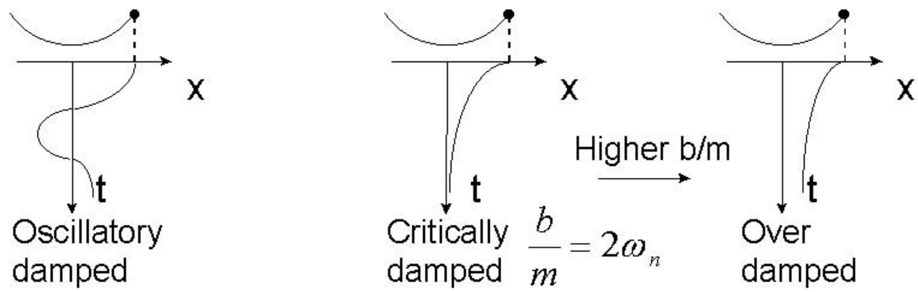

Figure 6.7: Dissipative Systems

The time-response of this system is

$$x(t) = ce^{-\xi_n t} \cos(\omega_n \sqrt{1 - \xi_n^2} t + \phi) \tag{6.14}$$

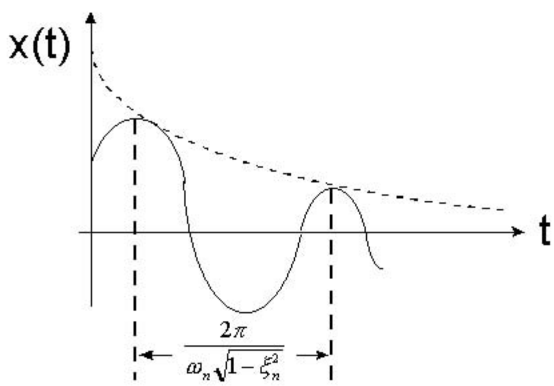

Figure 6.8: Dissipative Systems Response

The exponential decreases with the damping ratio, and the frequency of oscillation is affected by $\omega_n \sqrt{1 - \xi_n^2}$. This quantity is defined as the *damped natural frequency*,

$$\omega = \omega_n \sqrt{1 - \xi_n^2}$$

When $\xi_n = 1$, the system is critically damped and there are no oscillations. The smaller ξ_n, the closer is the damped natural frequency is to the natural frequency. The period of the oscillations is $\frac{2\Pi}{\omega_n \sqrt{1-\xi_n^2}}$.

Example Let us consider the simple example when $m = 2.0; b = 4.8; k = 8.0$. Since $\omega = \omega_n \sqrt{1 - \xi_n^2}$ we can obtain $\omega_n = \sqrt{\frac{k}{m}} = 2$, $\xi_n = \frac{b}{2\sqrt{km}} = 0.6$ and $\omega = 2.\sqrt{1 - 0.36} = 1.6$.

6.3 Passive-Behavior Control

We are going to reproduce the passive behavior of a natural system in the design of the robot controller. Let us consider a robot with 1-DOF prismatic joint, with mass m. The task is to move the robot from its current configuration, x_0, to a desired position x_d. The actuation of this robot involves one force, f, acting on the prismatic link. The robot equation of motion is

$$m\ddot{x} = f$$

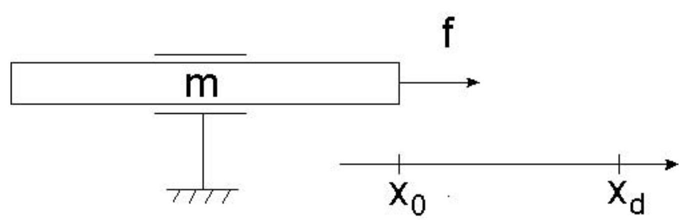

Figure 6.9: Simple goal control

The goal is to find the force f that accomplishes the task, while providing passive behavior. To achieve this behavior, we select f to be conservative – the gradient of a positive potential function. For this task, the potential function will be designed to have a zero minimum at the desired position x_d,

$$V(x) = \begin{cases} 0, & x = x_d \\ V(x) > 0, & x \neq x_d \end{cases}$$

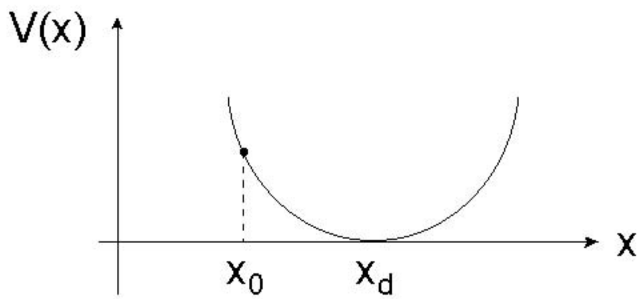

Figure 6.10: Potential function

The simplest such function is the potential

$$V(x) = \frac{1}{2}k_p(x - x_d)^2$$

The gradient of this potential is the conservative force

$$f = -\nabla V(x) = -\frac{\partial V}{\partial x}$$

Applying this control to the robot, leads to

$$m\ddot{x} = -\frac{\partial}{\partial x}[\frac{1}{2}k_p(x - x_d)^2] \tag{6.15}$$

The behavior of the controlled closed-loop system is

$$m\ddot{x} + k_p(x - x_d) = 0 \tag{6.16}$$

This behavior is identical to that we have already seen for the conservative mass-spring system. In this controlled system, however, the natural springiness, is reproduced by the parameter k_p, which controls the artificial stiffness of the closed loop system. Under this control, the link will oscillate around the desired position x_d, in similar fashion to the mass-spring system. The frequency of this oscillation is determined by $\sqrt{k_p/m}$. This frequency represents the *closed loop frequency*, ω of the controlled system.

With its conservative behavior, this system is stable. However, it is not asymptotically stable. Asymptotic stability can be achieved by the application of a dissipative force. This non-conservative force will appear on the right-hand side of Lagrange equation,

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}}) - \frac{\partial L}{\partial x} = \tau_{dissipative} \tag{6.17}$$

The general condition for asymptotic stability is

$$\dot{x}^T \tau_{dissipative} < 0, \quad \text{for } \dot{\text{x}} \neq 0 \tag{6.18}$$

Intuitively, the above condition implies that the dissipative forces are always acting to oppose the velocity. If we choose the dissipative force as $f_d = -k_v \dot{x}$, the asymptotic stability condition for this system becomes

$$\dot{x}^T(-k_v \dot{x}) = -k_v \dot{x}^2 < 0, \quad \text{for } \dot{\text{x}} \neq 0 \tag{6.19}$$

This condition is satisfied if $k_v > 0$. The total control force becomes

$$f = -k_p(x - x_d) - k_v \dot{x} \tag{6.20}$$

where $k_p > 0$ and $k_v > 0$. This is simply the conventional *proportional-derivative* control. k_p is *position gain* and k_v is the *velocity gain*. The closed-loop system corresponding to this control is described by

$$m\ddot{x} + k_v \dot{x} + k_p x = k_p x_d \tag{6.21}$$

To determine the characteristics of this 2nd order system, we divide the above equation by m and introduce the frequency and damping ratio, as we proceeded with the mass-spring system. This leads to

$$\ddot{x} + 2\xi\omega\dot{x} + \omega^2 x = \omega^2 x_d \tag{6.22}$$

with

$$\omega^2 = \frac{k_p}{m} \tag{6.23}$$

$$\xi = \frac{k_v}{2\sqrt{k_p m}} \tag{6.24}$$

where ω is the *closed-loop frequency* and ξ is the *closed-loop damping ratio.* Since these two parameters determines the response of the controlled system, we will first set ω and ξ to achieve the desired response. The position gain k_p and the velocity gain k_v can be then selected in accordance with the desired behavior. These are

$$k_p = m\omega^2 \tag{6.25}$$

$$k_v = m(2\xi\omega) \tag{6.26}$$

The above expressions for k_p and k_v show that both of these gains are proportional to the mass m, given a selection of ω and ξ. For a system with unit mass, these gains would be given by

$$k'_p = \omega^2 \tag{6.27}$$

$$k'_v = 2\xi\omega \tag{6.28}$$

k'_p and k'_v represent the position and velocity gains that correspond to a desired dynamic response for a unit-mass system.

$$1.\ddot{x} = f' \quad \text{with} \quad f' = -k'_v\dot{x} - k'_p(x - x_d)$$

and the closed-loop behavior for the single unit-mass system is

$$1.\ddot{x} + k'_v\dot{x} + k'_p x = k'_p x_d$$

Given the control gains for a desired behavior of a unit-mass system, the gains that provide the same behavior for an m-mass system are given by

$$\begin{aligned} k_p &= mk'_p \\ k_v &= mk'_v \end{aligned}$$

and the control f of the m-mass system is

$$m\ddot{x} = f \quad \text{with} \quad f = mf'$$

These relationships play an important role in extending our analysis to systems with nonlinearities and larger numbers of degrees of freedom.

6.3.1 Nonlinear Systems

Let us consider again the 1-DOF prismatic arm. A more realistic model of this system will include some amount of friction, which will be approximated by a nonlinear function of the position and velocity, i.e. $b(x, \dot{x})$. The system is now described by the equation

$$m\ddot{x} + b(x, \dot{x}) = f \tag{6.29}$$

If we were able to model the friction b, we could then use this model in the control of the system to compensate for this friction, and to control the resulting linearized system as before. The general structure for implementing this type of control is

$$f = \alpha f' + \beta \tag{6.30}$$

where β represents the portion of the control that compensates for the nonlinear forces acting on the system, and α is the mass of the system allowing the use of unit-mass system's control design f'. Since both the mass and the nonlinearities in the system most be identified, α and β will only be estimates of these quantities. In the case of our system, α and β are

$$\alpha = \widehat{m} \tag{6.31}$$
$$\beta = \widehat{b}(x, \dot{x}) \tag{6.32}$$

$\widehat{m}$ and $\widehat{b}(x, \dot{x})$ are the estimate for the mass and friction of the system.

$$m\ddot{x} + b(x, \dot{x}) = \widehat{m}f' + \widehat{b}(x, \dot{x}) \tag{6.33}$$

With perfect estimates, ($\widehat{m} = m$, and $\widehat{b} = b$), the closed-loop behavior of this system will be described by the the unit-mass system controlled by f'

$$1.\ddot{x} = f'$$

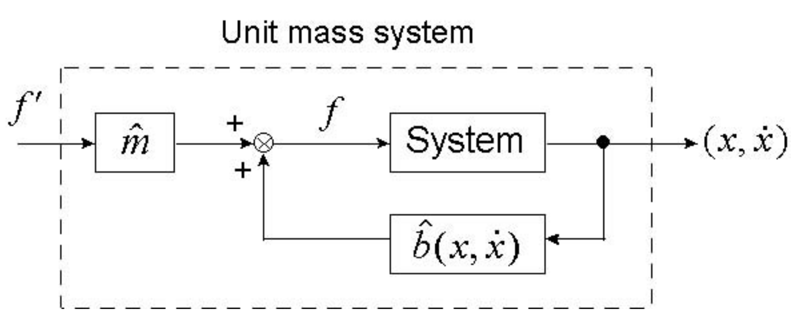

Figure 6.11: Non-linear control system

f', the control input of the linearized unit-mass system, will be designed with k_p' and k_v' to achieve the desired behavior. This control structure is shown in Figure 6.11. The dotted block in this Figure represents the unit-mass system being controlled by f' with outputs x and $\dot{x}$.

6.3.2 Motion Control

The task discussed above concerned placing the robot at a desired position, x_d. The system robot system

$$m\ddot{x} + b(x, \dot{x}) = f$$

is controlled by selecting

$$f = \widehat{m} f' + \widehat{b}(x, \dot{x})$$

with

$$f' = -k_v'\dot{x} - k_p'(x - x_d)$$

where k_p' and k_v' are the PD control gains. With perfect estimates, the closed-loop behavior is described by

$$1.\ddot{x} + k_v'\dot{x} + k_p'(x - x_d) = 0 \qquad (6.34)$$

A robot task may involve the tracking of a desired trajectory $x_d(t)$. In addition to the time-varying desired position, a trajectory tracking task generally involves the desired velocities and accelerations, i.e. $\dot{x}_d(t)$ and $\ddot{x}_d$. The robot control for this task will have an identical structure to the controller described above, with a new unit-mass control input, f', designed for trajectory tracking

$$f' = \ddot{x}_d - k_v'(\dot{x} - \dot{x}_d) - k_p'(x - x_d) \qquad (6.35)$$

The closed-loop system is described by

$$(\ddot{x} - \ddot{x}_d) + k'_v(\dot{x} - \dot{x}_d) + k'_p(x - x_d) = 0 \tag{6.36}$$

This is a second order system in the error $e = x - x_d$,

$$\ddot{e} + k'_v\dot{e} + k'_p e = 0 \tag{6.37}$$

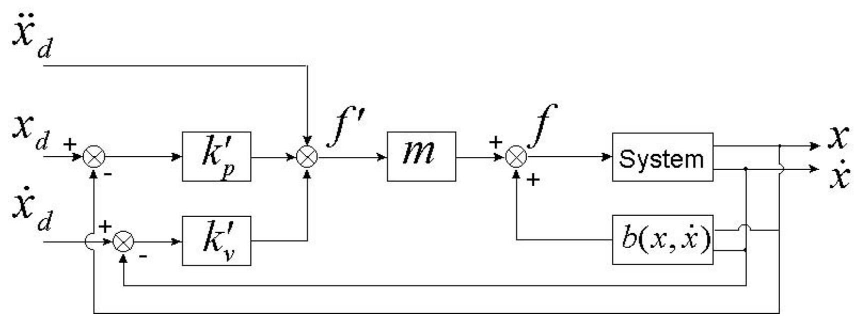

Figure 6.12: Error correction control

The trajectory tracking control system is shown in Figure 6.12. In addition to position and velocity feedback, trajectory tracking control involves an acceleration feedforward component that allows the system to better anticipate changes on the desired trajectory.

6.4 Disturbance Rejection

The linear closed-loop behavior we have achieved relies on the assumption of exact estimates of the system's parameters. The errors in these estimates, in addition to other unmodeled aspects of the system, all contribute to disturbance forces that could affect the performance and stability of the system. The effects of these disturbances can be minimized by appropriate selection of gains, k'_p and k'_v, involved in the unit-mass controller. The larger these gains are the better the disturbance rejection of the system becomes. There are, however, various factors that limit the gains, as we will see in section 6.4.1.

Let us assume that all disturbances acting on the system can be represented by a single disturbance force f_{dist}, that is directly acting at the input of system, as illustrated in Figure 6.13. In addition, this disturbance force will be assumed to be constant. The system's equation of motion becomes

$$m\ddot{x} + b(x, \dot{x}) = f + f_{\text{dist}} \tag{6.38}$$

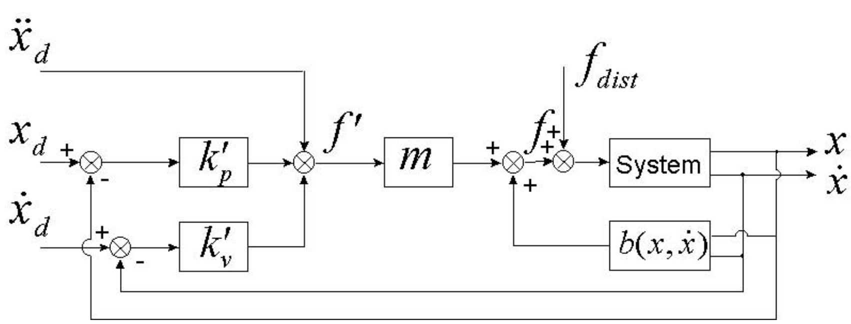

Figure 6.13: Disturbance rejection control

Using the control structure

$$f = mf' + b(x, \dot{x})$$

and the unit-mass control for trajectory tracking

$$f' = \ddot{x}_d - k'_v(\dot{x} - \dot{x}_d) - k'_p(x - x_d)$$

The closed-loop behavior of the unit-mass system is

$$\ddot{e} + k'_v\dot{e} + k'_p e = \frac{f_{dist}}{m} \tag{6.39}$$

For a desired position task,

$$f' = -k'_v \dot{x} - k'_p (x - x_d)$$

The closed-loop behavior of the system is

$$1.\ddot{x} + k'_v \dot{x} + k'_p (x - x_d) = \frac{f_{dist}}{m}$$

Steady-State Error The steady-state error is determined by analysis of the closed-loop system at rest, i.e. when all derivatives are set to zero. This leads to the steady-state equation

$$k'_p e = \frac{f_{dist}}{m} \tag{6.40}$$

Then the steady-state error is

$$e = \frac{f_{dist}}{m k'_p} = \frac{f_{dist}}{k_p} \tag{6.41}$$

Thus high values of k'_p reduce the steady-state error. This also shows that heavy systems (with large masses) are less sensitive to disturbances.

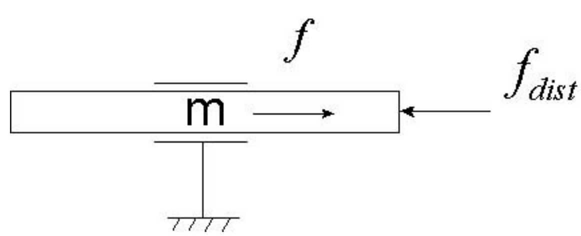

Figure 6.14: 1 DOF prismatic manipulator

Example Let us consider the 1-DOF manipulator shown in Figure 6.14.

The manipulator is controlled to the desired position x_d. The closed-loop behavior of this system is

$$m\ddot{x} + k_v\dot{x} + k_p(x - x_d) = 0 \tag{6.42}$$

Let us consider the application of a disturbance force, f_{dist}, to this system, and let us find the new position of the manipulator. At rest, the steady-state equation is

$$k_p(x - x_d) = f_{dist}$$

and the manipulator is positioned at

$$x = x_d + \frac{f_{dist}}{k_p}$$

6.4.1 Control Gain Limitations

The higher k_p is, the better the disturbance rejection becomes. However, control gains are limited by various factors involving structural flexibilities in the mechanism, time-delays in actuators and sensing, and sampling rates. An increase of k_p results in an increase of the closed-loop frequency ω. As this frequency approaches the the first unmodeled resonant frequency, $\omega_{\text{low-resonant}}$, the corresponding mode can be excited. It is thus important to keep ω well below this frequency. In addition, ω must be remain below the frequency corresponding to the largest time delay, $\omega_{\text{large-delay}}$. The frequency associated with the sampling rate, $\omega_{\text{sampling-rate}}$ also imposes a limitation on ω. Typically ω is selected as

$$\begin{aligned}
\omega &< \frac{1}{2}\omega_{\text{low-resonant}} \\
\omega &< \frac{1}{3}\omega_{\text{large-delay}} \\
\omega &< \frac{1}{5}\omega_{\text{sampling-rate}}
\end{aligned}$$

6.4.2 Integral Control

We have analyzed the performance of PD controllers. The addition of integral action to a PD controller allows us to further reduce disturbance errors. For a trajectory tracking task, A PID controller (proportional-integral-derivative) involves, in addition to k'_p, and k'_v, the integral gain k'_i. A PID controller for a trajectory tracking task is

$$f' = \ddot{x}_d - k'_v(\dot{x} - \dot{x}_d) - k'_p(x - x_d) - k'_i \int (x - x_d)dt \tag{6.43}$$

The closed-loop behavior in the presence of a disturbance force is

$$\ddot{e} + k'_v\dot{e} + k'_p e + k'_i \int e dt = \frac{f_{dist}}{m} \tag{6.44}$$

The disturbance force is assumed constant. Taking the the derivative of the equation above yields

$$\dddot{e} + k'_v\ddot{e} + k'_p\dot{e} + k'_i e = 0 \tag{6.45}$$

The steady-state error equation (all derivatives set to zero) is

$$e = 0$$

6.5 Actuation System

Let us consider the following illustration of gain selection for effective inertia optimization. Figure 6.15 depicts a gear reduction system.

The gear ratio is $\eta = \frac{R}{r}$, and the relationships for angles and torques at the motor and links are

$$\dot{\theta}_L = (\frac{1}{\eta})\dot{\theta}_m \tag{6.46}$$

$$\tau_L = \eta\tau_m \tag{6.47}$$

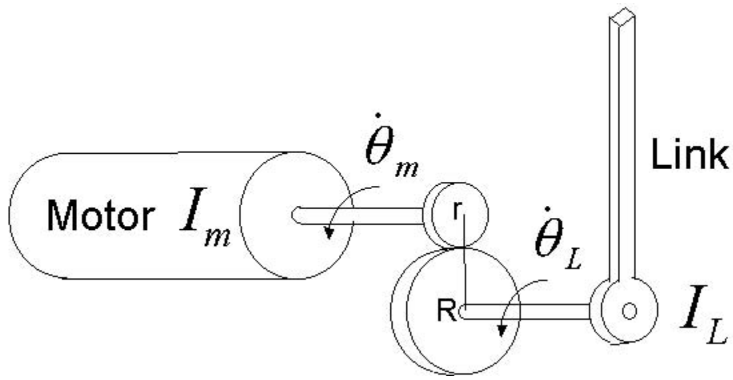

Figure 6.15: Gear Reduction

The corresponding equations of motion are

$$\tau_m = I_m \ddot{\theta}_m + \frac{1}{\eta}(I_L \ddot{\theta}_L) + b_m \dot{\theta}_m + \frac{1}{\eta} b_L \dot{\theta}_L \tag{6.48}$$

Since

$$\ddot{\theta}_L = \frac{1}{\eta} \ddot{\theta}_m$$

we can write

$$\tau_m = (I_m + \frac{I_L}{\eta^2}) \ddot{\theta}_m + (b_m + \frac{b_L}{\eta^2}) \dot{\theta}_m \tag{6.49}$$

and

$$\tau_L = (I_L + \eta^2 I_m) \ddot{\theta}_L + (b_L + \eta^2 b_m) \dot{\theta}_L \tag{6.50}$$

$(I_L + \eta^2 I_m)$ represents the *effective inertia*, and $(b_L + \eta^2 b_m)$ represents the *effective damping* both perceived at the link.

Finally, the position and velocity gains can be selected as

$$k_p = (I_L + \eta^2 I_m)k'_p \tag{6.51}$$
$$k_v = (I_L + \eta^2 I_m)k'_v \tag{6.52}$$

6.6 PD Control for Multi-link Systems

The control of multi-link manipulator system can be accomplished with a set of PD controllers designed independently for each link. While sufficient for pick-and-place tasks, this type of control is limited in its performance for tasks involving trajectory tracking and interactions with the environment. To analyze the limitations of PD controllers, let us consider the example of two revolute-joint manipulator shown Figure 6.16.

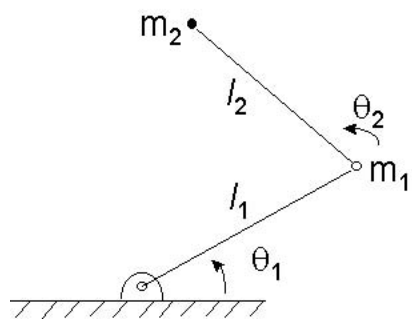

Figure 6.16: 2 DOF manipulator example

The dynamic equation of motion for this manipulator is

$$\begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} + \begin{pmatrix} m_{112} \\ 0 \end{pmatrix} (\, \dot{\theta}_1 \dot{\theta}_2 \,) + \begin{pmatrix} 0 & m_{122} \\ -\frac{m_{112}}{2} & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{pmatrix} + \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} \tag{6.53}$$

The two scalar equations corresponding to the behavior of joint 1 and joint 2 are

$$m_{11}\ddot{\theta}_1 + m_{12}\ddot{\theta}_2 + m_{112}\dot{\theta}_1\dot{\theta}_2 + m_{122}\dot{\theta}_2^2 + G_1 = \tau_1 \quad (6.54)$$
$$m_{22}\ddot{\theta}_2 + m_{21}\ddot{\theta}_1 - \frac{m_{112}}{2}\dot{\theta}_1^2 + G_2 = \tau_2 \quad (6.55)$$

In the design of two independent PD controllers for this robot, the dynamic coupling between the two links is ignored, and these links are treated as two decoupled systems described by

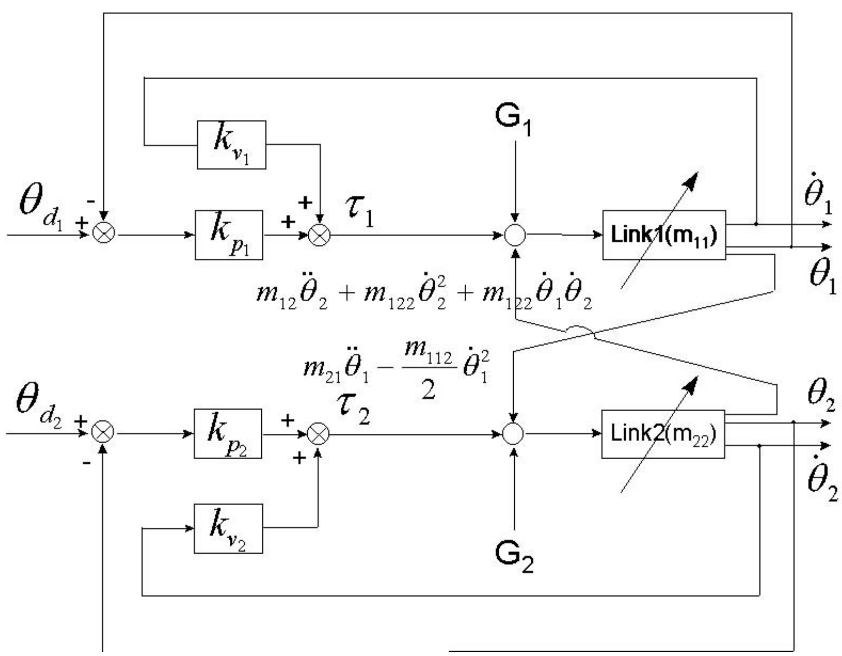

Figure 6.17: Controller for 2 DOF example

$$m_{11}\ddot{\theta}_1 = \tau_1 \quad (6.56)$$
$$m_{22}\ddot{\theta}_2 = \tau_2 \quad (6.57)$$

These equations neglect the dynamic forces acting on the joints, and ignore the configuration dependency of the link inertias. The actual system is nonlinear and highly coupled, as illustrated in Figure 6.18. The dynamic disturbances acting on an n-DOF manipulator controlled by n independent PD systems is illustrated in Figure 6.18.

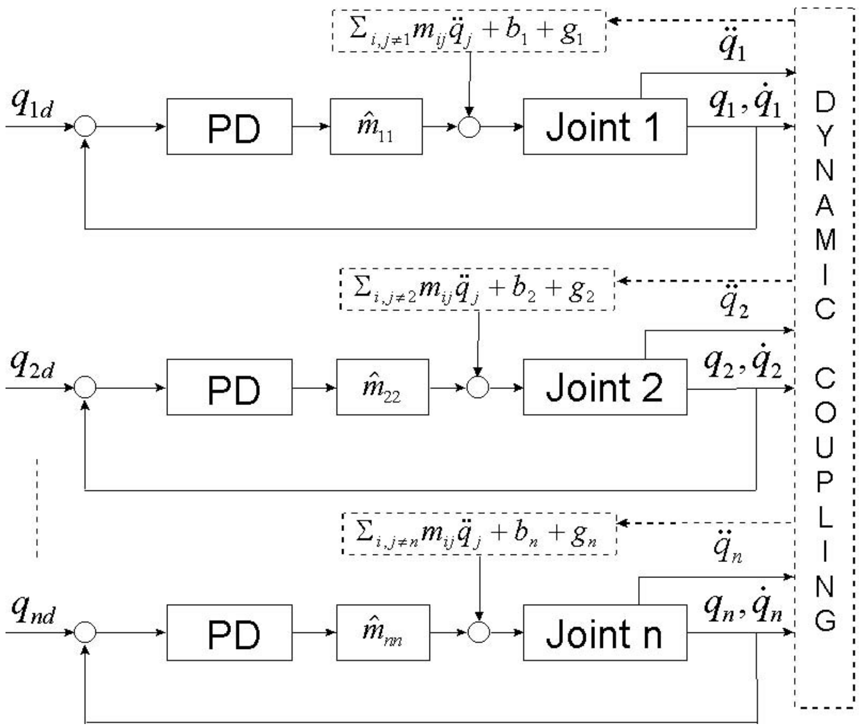

Figure 6.18: Coupled n DOF control

6.6.1 Stability of PD Control

The dynamics of an n DOF manipulator are described by

$$M\ddot{\mathbf{q}} + B(\mathbf{q})[\dot{\mathbf{q}}\dot{\mathbf{q}}] + C(\mathbf{q})[\dot{\mathbf{q}}^2] + \mathbf{g}(\mathbf{q}) = \tau \qquad (6.58)$$

With PD control, the joint torques are selected as

$$\tau = -k_p(\mathbf{q} - \mathbf{q}_d) - k_v\dot{\mathbf{q}} \tag{6.59}$$

The stability of this system can be easily concluded, as all external forces acting on this system are either conservative ($-k_p(\mathbf{q} - \mathbf{q}_d)$, gradient of a potential) or dissipative ($-k_v\dot{\mathbf{q}}$). That is

$$\tau = -\nabla_q V_d - k_v\dot{\mathbf{q}} \tag{6.60}$$

where

$$(V_d = \frac{1}{2}k_p(\mathbf{q} - \mathbf{q}_d)^T(\mathbf{q} - \mathbf{q}_d)$$

To further analyze the stability, let us consider again Lagrange's equation, from which equation 6.58 was obtained.

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) - \frac{\partial(K - V_{\text{gravity}})}{\partial \mathbf{q}} = -\nabla V_d - k_v\dot{\mathbf{q}} \tag{6.61}$$

where V_{gravity} represents the system's natural potential energy due to the gravity. Applying the control torque of equation 6.60, the controlled system becomes

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) - \frac{\partial(K - (V_{\text{gravity}} + V_d))}{\partial \mathbf{q}} = \tau_{dissipative} \tag{6.62}$$

where

$$\tau_{dissipative} = -k_v\dot{\mathbf{q}}$$

This shows how the conservative portion of the control modifies the potential energy (V_{gravity} to $V_{\text{gravity}} + V_d$). Without dissipative forces, this system is oscillatory, but stable. The addition of dissipative force provides asymptotic stability, under the condition

$$\dot{\mathbf{q}}^T\tau_{dissipative} < 0$$

which is verified for the dissipative force $-k_v\dot{\mathbf{q}}$ if $k_v > 0$.

A manipulator controlled with a set of independent PD controllers is stable, as the effect of these controllers is only to modify the manipulator's potential energy, while providing the dissipation needed for asymptotic stability.

6.6.2 Joint Space Dynamic Control

While providing stability, a PD controller is limited in its performance as it ignores the dynamic coupling forces. High gains provide better disturbance rejection, but as we mentioned earlier, control gains are limited by the system's flexibilities, time delays and sampling rate. Dynamic decoupling and motion control of the robot system can be accomplished by a control structure that uses the manipulator dynamic model. The manipulator dynamics are described by

$$M(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{v}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \tau \tag{6.63}$$

Based on this model, the control structure for dynamic decoupling and control is

$$\tau = \widehat{M}(\mathbf{q})\tau' + \widehat{\mathbf{v}}(\mathbf{q}, \dot{\mathbf{q}}) + \widehat{\mathbf{g}}(\mathbf{q}) \tag{6.64}$$

where $\widehat{.}$ represents an estimate. If we apply this control to the robot, the closed-loop behavior will be described by

$$1.\ddot{\mathbf{q}} = (M^{-1}\widehat{M})\tau' + M^{-1}[(\mathbf{v} - \widehat{\mathbf{v}}) + (\mathbf{g} - \widehat{\mathbf{g}})] \tag{6.65}$$

With prefect estimates, the system is described by the unit-mass system

$$1.\ddot{\mathbf{q}} = \tau' \tag{6.66}$$

With a PD design, the control input for the unit-mass systems, τ', is

$$\tau' = \ddot{\mathbf{q}}_d - k_v'(\dot{\mathbf{q}} - \dot{\mathbf{q}}_d) - k_p'(\mathbf{q} - \mathbf{q}_d) \tag{6.67}$$

and the closed-loop system is

$$\ddot{\mathbf{e}} + k_v'\dot{\mathbf{e}} + k_p'\mathbf{e} = 0 \tag{6.68}$$

with

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d$$

The overall control system is shown in Figure 6.19. This structure provides decoupling and linearization of the robot system, rendering it as set of unit-mass systems, controlled by τ'. The control input to the decoupled system was selected as a set of simple PD controllers, but obviously other control designs can be used.

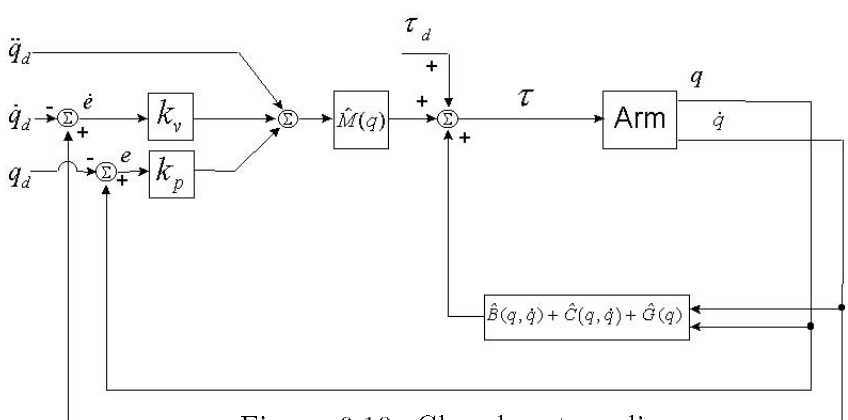

Figure 6.19: Closed system diagram

The input to the control system shown in Figure 6.19 is the joint trajectory, q_d, $\dot{q}_d$ and $\ddot{q}_d$. However, robot tasks are generally specified in terms of end-effector descriptions. In which case, the end-effector task must be first transformed into joint specifications.

6.7 Operational Space control

The operational space framework provides direct control of the end-effector motions, eliminating the need for task transformation. The problem of task transformation and joint coordination is yet more difficult for manipulation involving redundant mechanisms or multiple robots. The manipulation of an object, for example, by two robots, as illustrated in Figure 6.20, requires complex real-time coordination. This becomes unnecessary with the direct control of the manipulated object provided in the operational space approach.

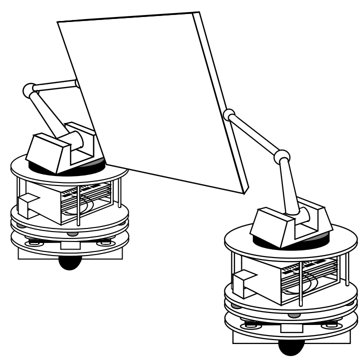

Figure 6.20: Multiple arm manipulation

The basic idea in the operational space approach is to control the end effector by a potential function whose minimum is at the end-effector goal position.

$$V_{\text{goal}}(\mathbf{x}) = \frac{1}{2}k_p(\mathbf{x} - \mathbf{x}_{\text{goal}})^T(\mathbf{x} - \mathbf{x}_{\text{goal}}) \tag{6.69}$$

The corresponding force, $\mathbf{F}$, that must be created at the end is given by the gradient of this potential,

$$\mathbf{F} = -\nabla_x V_{\text{goal}}(\mathbf{x})$$

This force will be produced by a torque vector at the joint of the robot, given simply by

$$\tau = J^T \mathbf{F}$$

Other more complex behaviors can be simply created by the design of artificial potential functions to avoid joint limits, kinematic singularities, or obstacles.

To analyze the stability of this type of control, we again resort to Lagrange's equation

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial (K - V_{\text{gravity}})}{\partial \mathbf{q}} = \tau \tag{6.70}$$

The control forces applied to the system are

$$\tau = J^T(-\nabla_x V_{goal}) \tag{6.71}$$

which can be rewritten as

$$\tau = -\nabla_q V_{goal} \tag{6.72}$$

and the controlled system becomes

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial (K - (V_{\text{gravity}} + V_{\text{goal}}))}{\partial \mathbf{q}} = 0 \tag{6.73}$$

This system is stable. The asymptotic stability requires the addition of damping forces, for instance

$$\mathbf{F}_d = -k_v \dot{\mathbf{x}}$$

The corresponding torques are

$$\tau_d = J^T(-k_v\dot{\mathbf{x}})$$

and the Lagrange's equation becomes

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{q}}}) - \frac{\partial(K - (V_{\text{gravity}} + V_{\text{goal}}))}{\partial \mathbf{q}} = \tau_d$$

The condition for asymptotic stability is

$$\dot{\mathbf{q}}^T\tau_d < 0 \tag{6.74}$$

Replacing $\dot{\mathbf{x}}$ by $J\dot{\mathbf{q}}$ yields

$$\dot{\mathbf{q}}^T\tau_d = -k_v[\dot{\mathbf{q}}^T(J^TJ)\dot{\mathbf{q}})] < 0$$

For a non-redundant manipulator and outside of singularities, J^TJ, is a positive definite matrix, and the quantity $[\dot{\mathbf{q}}^T(J^TJ)\dot{\mathbf{q}})]$ is positive. The system is then asymptotically stable if $k_v > 0$.

6.7.1 Operational Space Dynamics

The description of the dynamics at the end-effector requires first to select a set of generalized coordinates, $\mathbf{x}$, that represent the end-effector position and orientation, e.g. $(x, y, z, \alpha, \beta, \gamma)$. The kinetic energy of the system can then be expressed as a quadratic form of the generalized velocities, $\dot{\mathbf{x}}$,

$$K_x = \frac{1}{2}\dot{\mathbf{x}}^T M_x \dot{\mathbf{x}} \tag{6.75}$$

where M_x represents the mass matrix associated with the inertial properties at the end effector. Let $\mathbf{F}$ be the vector of generalized forces corresponding to the generalized coordinates x. The end-effector equations of motion are

$$\frac{d}{dt}(\frac{\partial K}{\partial \dot{\mathbf{x}}}) - \frac{\partial (K - V)}{\partial \mathbf{x}} = \mathbf{F} \tag{6.76}$$

which can be developed in the form

$$M_x \ddot{\mathbf{x}} + \mathbf{v}_x(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_x(\mathbf{q}) = \mathbf{F}$$

This equation is similar to the one we have obtained for joint space dynamics. In fact, joint-space and operational-space dynamics are related by simple relationships. First let us examine the kinetic energy. In terms of joint velocities, the kinetic energy of the system is

$$K_q = \frac{1}{2}\dot{\mathbf{q}}^T M \dot{\mathbf{q}} \tag{6.77}$$

where M is the joint space mass matrix. Expressing the fact that $K_x = K_q$, we can establish the relationship

$$J^T M_x J = M \tag{6.78}$$

which leads to

$$M_x = J^{-T} M J^{-1} \tag{6.79}$$

The relationship between the gravity force vectors $\mathbf{g}$ and $\mathbf{g}_x$ is simply given by the transpose of the Jacobian matrix.

$$\mathbf{g} = J^T \mathbf{g}_x$$

The relationship between $\mathbf{v}$ and $\mathbf{v}_x$ involves the time derivatives of the Jacobian matrix ($\ddot{\mathbf{x}} = J\ddot{\mathbf{q}} + \dot{J}\dot{\mathbf{x}}$). In summary these relationships are

$$\begin{aligned} M_x &= J^{-T} M J^{-1} && (6.80) \\ \mathbf{v}_x &= J^{-T}\mathbf{v} - M_x \dot{J}\dot{q} && (6.81) \\ \mathbf{g}_x &= J^{-T}\mathbf{g} && (6.82) \end{aligned}$$

6.7.2 Operational Space Dynamic Control

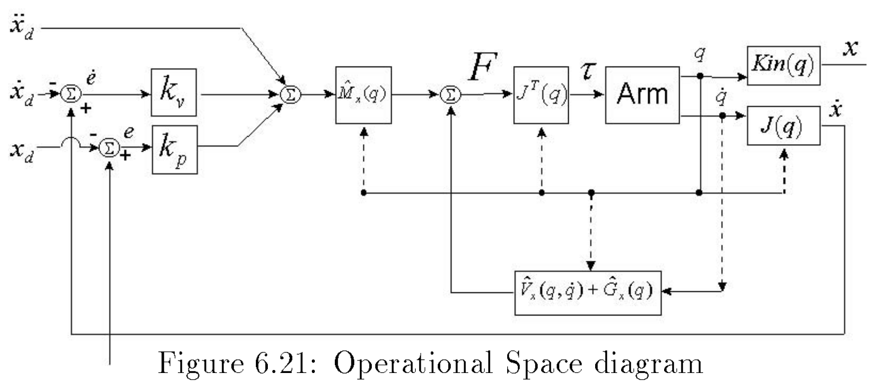

Figure 6.21: Operational Space diagram

The end-effector dynamics is described by the equation

$$M_x \ddot{\mathbf{x}} + \mathbf{v}_x(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}_x(\mathbf{q}) = \mathbf{F}$$

The control structure for dynamic decoupling and motion control is

$$\mathbf{F} = \widehat{M}_x \mathbf{F}' + \hat{\mathbf{v}}_x + \widehat{\mathbf{g}}_x \tag{6.83}$$

where $\hat{}$ represents an estimate. $\mathbf{F}'$ is the input of the unit-mass system,

$$1.\ddot{\mathbf{x}} = \mathbf{F}' \tag{6.84}$$

For an end-effector trajectory following, $\mathbf{F}'$ is

$$\mathbf{F}' = \ddot{\mathbf{x}}_d - k_v(\dot{\mathbf{x}} - \dot{\mathbf{x}}_d) - k_p(\mathbf{x} - \mathbf{x}_d) \tag{6.85}$$

The operational space control structure is shown in Figure 6.21. Here, the trajectory is directly specified in terms of end-effector motion, x_d, $\dot{x_d}$ and $\ddot{x_d}$.