Essential Kinematics
for Autonomous Vehicles

Alonzo Kelly

CMU-RI-TR-94-14 - REV 2.0

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

July   18, 2006

Rev 2.0 is issued to correct many typos. Changes from Rev 1.0 are identifed by change bars in the margin

The views and conclusions expressed in this document are those of the author and should not be interpreted as representing the official policies, either express or implied, of the US government.

Abstract

A short tutorial on Homogeneous Transforms is presented covering the triple interpretation of a homogeneous transform as an operator, a coordinate frame, and a coordinate transform. The operator / transform duality is derived and its use in the Denavit Hartenberg convention is explained. Forward, inverse, and differential kinematics are derived for a simple manipulator to illustrate concepts.

A standard set of coordinate frames is proposed for wheeled mobile robots. It is shown that the RPY transform serves the same purpose as the DH matrix in this case. It serves to interface with vehicle position estimation systems of all kinds, to control and model pan/tilt mechanisms and stabilized platforms, and to model the rigid transforms from place to place on the vehicle. Forward and inverse kinematics and the Euler angle rate to the angular velocity transform are derived for the RPY transform.

Projective kinematics for ideal video cameras and laser rangefinders, and the imaging Jacobian relating world space and image space is derived. Finally, the kinematics of the Ackerman steer vehicle is presented for reference purposes.

This report is both a tutorial and a reference for the transforms used in the RANGER vehicle controller. It is both because the models keep evolving and it was necessary to provide the tools, mechanisms, and discipline required to continue the evolution.

Table of Contents

1. Introduction

Kinematic modelling is one of the most essential analytical tools of robotics. It is used for modelling mechanisms, actuators, and sensors for on-line control, off-line programming, and simulation purposes. This document presents a brief survival kit of concepts and techniques that will equip the reader to master a large class of kinematic modelling problems.

Control of autonomous vehicles in 3D can require precise kinematic models of mechanisms, the image formation process, terrain following, steering kinematics, and more. The document provides the tools necessary to solve these problems in one place for reference purposes.

1.1 Acknowledgments

Learning how to do kinematics has been a long process that started with my bachelors thesis done under Andrew Goldenberg at the University of Toronto in 1984. Sid Skull supervised my AI project at York University which was a PROLOG based symbolic mathematics package and an expert system for robot kinematic modelling. At CMU, Martial Hebert and Jay Gowdy started my interest in the peculiarities of modelling the kinematics of rangefinders. Omead Amidi first showed me the bicycle model for Ackerman vehicles. Anthony Stentz first referred me to the need for projection lookup tables for fast perception for off road purposes.

1.2 Commentary

Every year, I have to learn this stuff all over again, so I finally bit the bullet and tried to write down the issues that seem important to me. Somehow, all these matrices with all those strings of trig functions in them have always seemed awkward and hard to figure out. Somewhere in the past there was an obscure book on computer graphics that presented this information from the operator/ transform duality point of view that seemed to me clear and concise. I've never been able to find it again.

This report is both a tutorial and a reference for the transforms used in the RANGER vehicle controller. It is both because the models keep evolving and it was necessary to provide the tools, mechanisms, and discipline required to continue the evolution.

1.3 Notational Conventions

The 3 X 3 matrix R_a^b denotes a rotation matrix which transforms a displacement from its expression in coordinate system 'a' to its expression in coordinate system 'b'. The 4 X 4 matrix T_a^b denotes the homogeneous transform which transforms a vector from its expression in coordinate system 'a' to its expression in coordinate system 'b'. If $\bar{p}^a$ denotes a point expressed in coordinate system 'a', then the notation for conversion of coordinates to coordinate system 'b' is easy to remember by considering that the a's cancel:

$$\bar{p}^b = T_a^b \bar{p}^a$$

The 4 X 4 matrix P_a^b denotes a nonlinear projection operator *represented* as a homogeneous transform. In such notation, the vector normalization step is implicit in the transform. The 3 X 1 vector $\bar{r}_a^b$ represents the translation vector from system 'a' to system 'b' expressed in system 'a' coordinates. The matrix J_a^b denotes the Jacobian of the transformation from system 'a' to system 'b'.

The specification of derivatives will be necessarily loose. If x and y are scalars, $\bar{x}$ and $\bar{y}$ are vectors, and X and Y are matrices, then all of the following derivatives can be defined.

$\dfrac{\partial y}{\partial x}$	a partial derivative	$\dfrac{\partial y}{\partial \bar{x}}$	a gradient vector
$\dfrac{\partial}{\partial x}\bar{y}$	a vector partial derivative	$\dfrac{\partial}{\partial \bar{x}}\bar{y}$	a Jacobian matrix
$\dfrac{\partial Y}{\partial x}$	a matrix partial derivative	$\dfrac{\partial Y}{\partial \bar{x}}$	a Jacobian tensor

The notation ca and sa is shorthand for cos(a) and sin(a) respectively. Sums of angles, for example, will be indicated by cascaded subscripts as follows:

$$c_{123} = cos(\psi_1 + \psi_2 + \psi_3)$$

Bolded italic text is used for emphasis, whereas bolded nonitalic text highlights key words that appear in the index.

2. Homogeneous Coordinates and Transforms

Homogeneous coordinates are a method of representing 3D entities by the projections of 4D entities onto a 3D subspace. This section investigates why such an artificial construct has become the cornerstone of robot kinematic modelling.

2.1 Points

A point is a position in space. Points will be represented by column vectors such as:

$$p_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$$

where the superscript T denotes the matrix transpose and permits writing a column vector as a row vector.

2.2 Operators

An **operator** is any process which maps points onto other points. For the present purpose, operators will be limited to those which can be represented as matrices. The above point p_1 can be altered in many different ways by multiplying it by a general 3 X 3 matrix T:

$$p_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = Tp_1 = \begin{bmatrix} t_{xx} & t_{xy} & t_{xz} \\ t_{yx} & t_{yy} & t_{yz} \\ t_{zx} & t_{zy} & t_{zz} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} t_{xx}x_1 + t_{xy}y_1 + t_{xz}z_1 \\ t_{yx}x_1 + t_{yy}y_1 + t_{yz}z_1 \\ t_{zx}x_1 + t_{zy}y_1 + t_{zz}z_1 \end{bmatrix}$$

Some simple operators are scale, reflection, rotation, projection, and shear. All of these can be generated from the above matrix by a suitable choice of the entries in the matrix.

2.3 Homogeneous Coordinates

Even though the above operator is very general, it is not general enough to be used conveniently in many robotics applications because it cannot represent a translation. That is, there exists no 3 X 3 matrix which adds a constant vector, like p_3 to p_1. Such a translation could be expressed as a vector   addition:

$$p_2 = p_1 + p_3 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} + \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}$$

This cannot be done by a matrix because p_3 cannot be represented, in general, as a linear combination of the elements of p_1 - it is supposed to be independent of p_1.

However, the situation can be fixed with a standard trick. In **homogeneous coordinates,** an extra

element can be added to each point to represent a kind of scale factor:

$$p_1 = \begin{bmatrix} x_1 & y_1 & z_1 & w_1 \end{bmatrix}^T$$

and it is conventional to consider that this 4-vector is projected into 3D by dividing by the scale factor:

$$p_1 = \begin{bmatrix} \dfrac{x_1}{w_1} & \dfrac{y_1}{w_1} & \dfrac{z_1}{w_1} \end{bmatrix}^T$$

Points are typically represented with a scale factor of 1. Thus:

$$p_1 = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \end{bmatrix}^T$$

is a point in homogeneous coordinates. It is also possible to represent a pure direction in terms of a point at infinity by using a scale factor of 0. Thus:

$$q_1 = \begin{bmatrix} x_1 & y_1 & z_1 & 0 \end{bmatrix}^T$$

is a direction in homogeneous coordinates.

2.4 **Homogeneous Transforms**

Using homogeneous coordinates, it is now possible to represent the addition of two vectors as a matrix operation, thus:

$$p_2 = p_1 + p_3 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} + \begin{bmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_3 \\ 0 & 1 & 0 & y_3 \\ 0 & 0 & 1 & z_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \mathrm{Trans}(p_3)p_1$$

where $\mathrm{Trans}(p_3)$ is the homogeneous transform which is equivalent to a translation operator for the translation vector p_3.

2.5 Homogeneous Transforms as Operators

Suppose it is necessary to move a point in some manner and express the result in the *same* coordinate system as the original point. This is the notion of an operator. The basic operators are translation along and rotation about any of the three axes. The following table gives the four elementary operators which are sufficient for the purposes of the report, and for almost any real problems. Operators are identified by an upper case first letter.

$$\text{Trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotx}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Roty}(\phi) = \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotz}(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
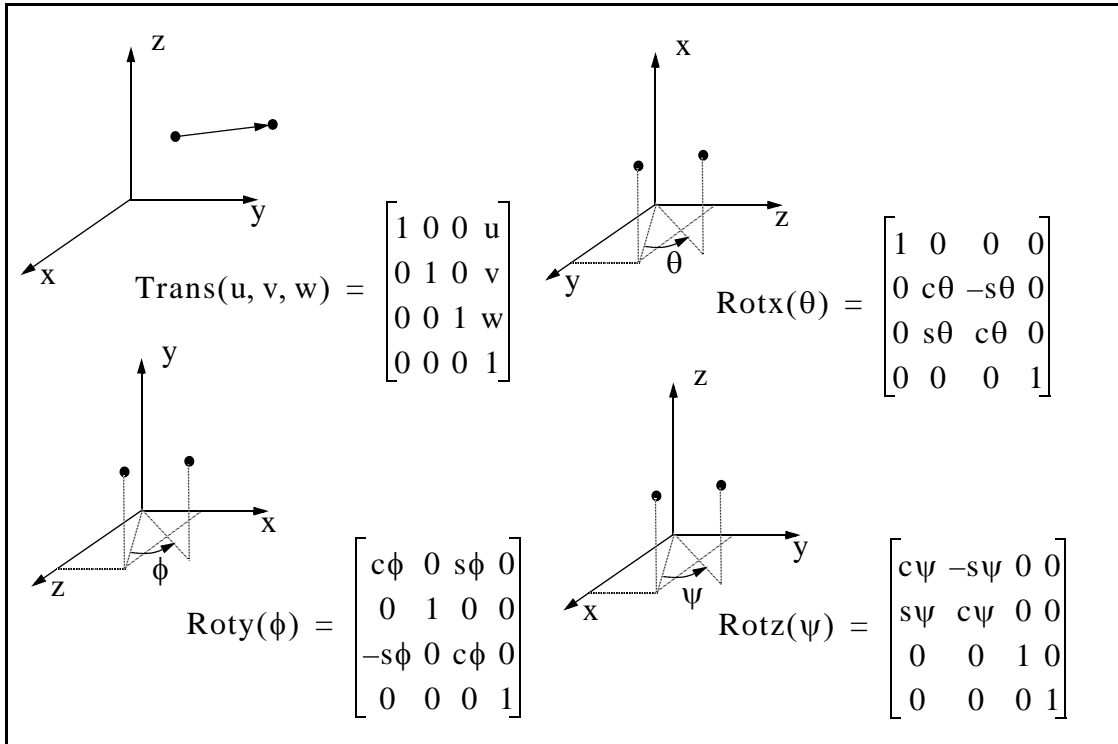
It is important to remember the precise semantics of these operators. They take a point expressed in some coordinate system, operate on it, and supply the result expressed in the same coordinate system.
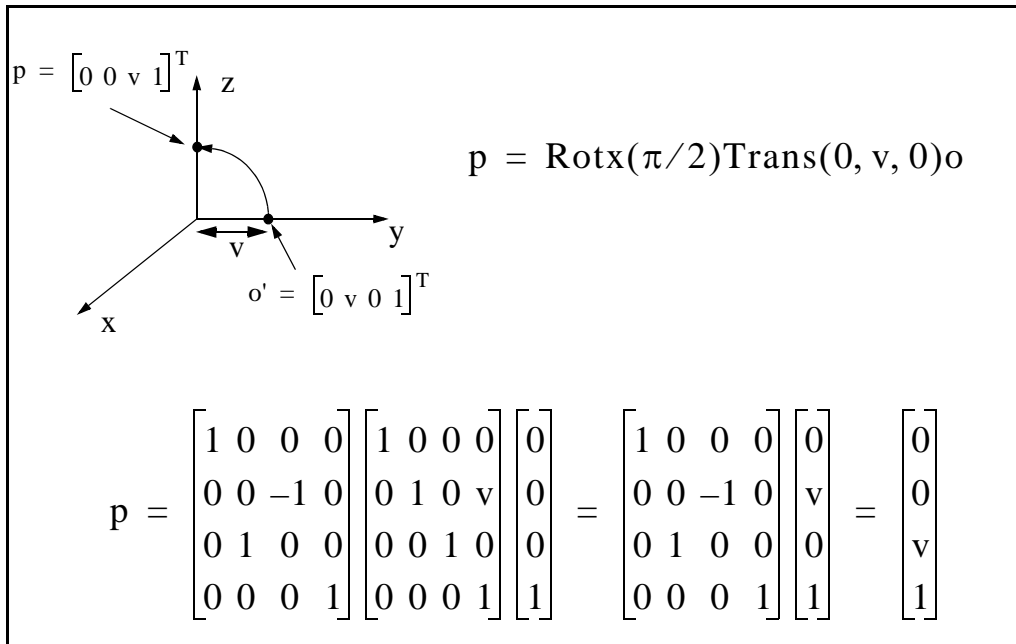
One of the most common operators used in robotics is a rotation followed by a translation. The homogeneous transform can be used to represent this operator as a single matrix. In general, many kinds of operators can be represented by homogeneous transforms. However, the operator interpretation is but one of three possible interpretations of the homogeneous transform.

2.6 Example - Operating on a Point

The homogeneous coordinates of the origin o are clearly:

$$o = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

As an example of applying operators to a point, the following figure indicates the result of translating the origin along the y axis by 'v' units and then rotating the resulting point by 90° around the x axis.

$$p = \mathrm{Rotx}(\pi/2)\mathrm{Trans}(0, v, 0)o$$

$$p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ v \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v \\ 1 \end{bmatrix}$$

Notice that the operators are written in right to left order because this is the order in which they are applied to the original column vector representing the origin. The order is important because matrix multiplication is not commutative. Also notice that the initial point, intermediate point, and final result are all expressed in terms of the axes of the original coordinate system. Operators have *fixed axis semantics*.

2.7 Homogeneous Transforms as Coordinate Frames

Recall that homogeneous coordinates can be used to represent directions as well as points. This is done by using a zero scale factor. The unit vectors of a cartesian coordinate system can be considered to be directions. The x, y, and z unit vectors can be written as:

$$i = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \qquad j = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T \qquad k = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$$

Therefore, these three columns can be grouped together with the homogeneous coordinates of the
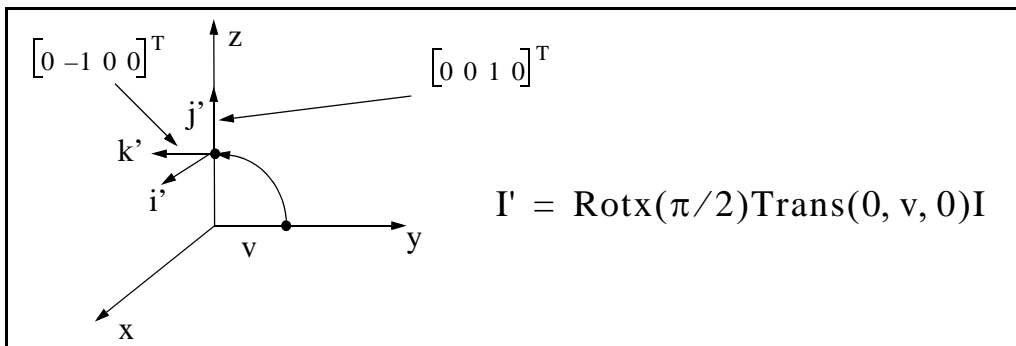
origin to form an identity matrix:

$$\begin{bmatrix} i & j & k & o \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

Clearly, the columns of the identity matrix can be interpreted as the unit vectors and the origin of a coordinate system. The result of applying an operator to these four vectors has a similar interpretation that will be shown by example.

2.8 Example - Interpreting an Operator as a Frame

These four vectors can be transformed simultaneously by applying the same two operators to the identity matrix:

$$I' = \mathrm{Rotx}(\pi/2)\mathrm{Trans}(0, v, 0)I$$

$$I' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This result is just the product of the original two operators. Each column of this result is the transformation of the corresponding column in the original identity matrix. Thus the transformed vectors are:

$$i' = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T \qquad j' = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$k' = \begin{bmatrix} 0 & -1 & 0 & 0 \end{bmatrix}^T \qquad o' = \begin{bmatrix} 0 & 0 & v & 1 \end{bmatrix}^T$$

It has been shown that the column vectors of any compound fundamental operator are just the

homogeneous coordinates of the transformed unit vectors and the origin. This is because the operators apply to any vectors - including the origin and the unit vectors. The result can be visualized by drawing the new transformed axes at the new position.

2.9 The Coordinate Frame

So far in the discussion, there are two complementary interpretations of exactly the same 4 X 4 matrix. It can be an operator which moves points and vectors around, or it can be a representation of a cartesian coordinate system positioned somewhere in space relative to another one. Cartesian coordinate systems positioned in space are a central concept in 3D kinematics. They encode both a position and an attitude. With an encoded position and attitude available, it becomes possible to talk about the *state of motion* of the origin in terms of translation and orientation and all of their associated time derivatives. Therefore, this entity embodies the properties of a *frame of reference*.

With a set of three orthogonal unit vectors, it is possible to represent an arbitrary vector quantity in terms of its projections onto these axes. Therefore, this entity also embodies the properties of a cartesian *coordinate system*. This moving set of unit vectors is often called a **coordinate frame** or simply a **frame**. Often, imaginary frames are embedded in the links of a mechanism in order to keep track of its configuration. These *embedded frames* are central to the study of manipulators, legs and other mechanisms.

2.10 Homogeneous Transforms as Coordinate Transforms

There is a third interpretation of a homogeneous transform. It was shown that they move points and move and orient frames and that they directly represent frames. However, if they move frames, one can think about the original frame and the transformed frame as two different frames and then ask about the relationship between the coordinates of any point in each frame. Let the original frame be called 'a' and the transformed one be called 'b'.

Let a general point p be expressed in the coordinates of frame 'b'. This will be represented by a superscript b. Its coordinates in this frame are:

$$p^b = \begin{bmatrix} x^b & y^b & z^b & 1 \end{bmatrix}$$

Alternately, it can be written as a true physical vector thus:

$$\vec{p}^b = x^b\hat{i}_b + y^b\hat{j}_b + z^b\hat{k}_b$$

This vector can be expressed in the coordinates of frame 'a' by expressing the unit vectors of frame 'b' in the coordinates of frame 'a' and adding the position of the origin, thus:

$$\vec{p}^a = x^b i' + y^b j' + z^b k' + o'$$

but this is just the same as if a matrix multiplication were used:

$$p^a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^b \\ y^b \\ z^b \\ 1 \end{bmatrix}$$

It has been shown that the same homogeneous transform that moves frame 'a' into coincidence with frame 'b' also converts the coordinates of points *in the opposite direction* - from frame 'b' to frame 'a'. The reason for the opposite sense in the interpretation is that moving a point "forward" in a coordinate system is completely equivalent to moving the coordinate system "backward".

It is easy to show that the inverse of a rotation matrix is equal to its transpose. This is called the property of **orthogonality**. This leads to a second set of 4 matrices, the fundamental transforms, which are the inverses of the operators and which therefore convert coordinates from 'a' to 'b'. These are given in the table below. In the report, the transforms will be denoted with lowercase first letters in their names.

$$\text{trans}(u, v, w) = \begin{bmatrix} 1 & 0 & 0 & -u \\ 0 & 1 & 0 & -v \\ 0 & 0 & 1 & -w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rotx}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & s\theta & 0 \\ 0 & -s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{roty}(\phi) = \begin{bmatrix} c\phi & 0 & -s\phi & 0 \\ 0 & 1 & 0 & 0 \\ s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
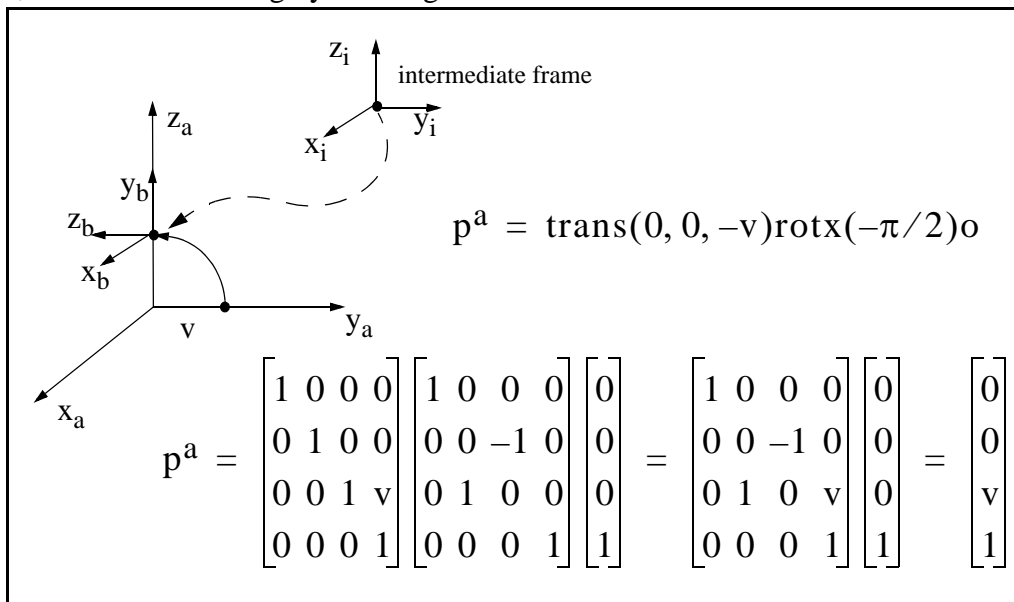
$$\text{rotz}(\psi) = \begin{bmatrix} c\psi & s\psi & 0 & 0 \\ -s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.11 Example - Transforming the Coordinates of a Point

Again, the homogeneous coordinates of the origin o are:

$$o = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

As an example of applying transforms to a point, the following figure indicates the result of transforming the origin of frame 'b' into the coordinates of frame 'a' where the two frames are related by the same sequence of operators used earlier. Frame 'b' is moved into coincidence with frame 'a' by first rotating frame 'b' by -90° around its x axis to produce the intermediate frame in the figure, and then translating by -v along the z axis of the intermediate frame.

$$p^a = \text{trans}(0, 0, -v)\text{rotx}(-\pi/2)o$$

$$p^a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & v \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v \\ 1 \end{bmatrix}$$

Notice that the transforms are written in right to left order because this is the order in which they are applied to the original column vector representing the origin. The order is important because matrix multiplication is not commutative. Also notice that, for transforms, each new transform is interpreted as if it were applied to the last frame in the sequence of frames that ultimately bring the first into coincidence with the last. Transforms have ***moving axis semantics***.

2.12 Inverse of a Homogeneous Transform

The homogeneous transforms that will be used in this report will all be structured according to the following template:

The scale factor will almost always be 1 and the perspective part will be all zeros except when modelling cameras. Under these conditions, it is easy to show by multiplying the inverse by the original matrix, that the inverse is:

$$\left[\begin{array}{ccc|c} & R & & p \\ \hline 0 & 0 & 0 & 1 \end{array} \right]^{-1} = \left[\begin{array}{ccc|c} & R^T & & -R^Tp \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

This is very useful for converting from a matrix that converts coordinates in one direction to one that converts coordinates in the opposite direction. Throughout the report, remember that it is trivial to reverse the sense of a coordinate transform.

2.13 A Duality Theorem

The careful reader has probably noticed that operators and transforms are inverses. Whether a particular matrix is considered to be operator or transform is a matter of taste. This has an important implication for kinematic modelling that will be demonstrated by example. Using the previous example, the moving axis operations which bring frame 'a' into coincidence with frame 'b' are:

- translate v units along the z axis
- rotate 90° around the new x axis

The complete transform that converts coordinates from frame 'a' to frame 'b' can be written as:

$$T_a^b = \text{rotx}(\pi/2)\text{trans}(0, 0, v)$$

This can be rewritten in terms of operators as:

$$T_a^b = \text{Rotx}(-\pi/2)\text{Trans}(0, 0, -v)$$

The inverse of this matrix will convert coordinates from frame 'b' to frame 'a', and as was shown earlier, also represents the position and attitude of frame 'b' with respect to frame 'a'. Using the well known method for inverting a matrix product, this inverse matrix can be expressed as follows:

$$T_b^a = \text{Trans}(0, 0, -v)^{-1}\text{Rotx}(-\pi/2)^{-1}$$

$$T_b^a = \text{Trans}(0, 0, v)\text{Rotx}(\pi/2)$$

So that the transform from 'b' to 'a' can be written in terms of transforms from right to left or in terms of operators *in the reverse order*. The latter view is traditional in robotics and it will be used in the next section.[1]

---

1. One implication of this is that any set of rotations about fixed axes is equivalent to the same set executed in the reverse order about moving axes.

3. Forward Kinematics

Forward kinematics is the relatively straightforward process of chaining homogeneous transforms together in order to represent the articulations of a mechanism or simply to represent the fixed transformation between two frames. In this process, the joint variables are given, and the problem is to find the transform.

3.1 Nonlinear Mapping

It is clear that it is possible to transform coordinates between coordinate systems which are fixed in position and attitude with respect to each other. It is also possible to perform elementary operations with homogeneous transforms. Ultimately, this is possible because such operations are *linear*. That is, the result is always a weighted sum of the original point or vector.

Most mechanisms, however, are not linear. They are composed of one or more rotary degrees of freedom. It turns out that the homogeneous transform can still be used to model such complex devices if they are viewed in a different way. In the study of mechanisms, the parameters of the rotations and translations between the frames are considered to be variables. In this sense, the problem is one of understanding the variation in a homogeneous transform when it is considered to be a function of one or more variables.

In the study of mechanisms, the real world space in which a mechanism moves is often called **task space**, and points and vectors are normally expressed in terms of cartesian coordinates. However, the mechanism articulations are most easily expressed in terms of angles and displacements along axes. This space is called **configuration space**. So far, we have studied linear transformations within task space. This section and the following sections will consider the more difficult problem of expressing the relationship between task space and configuration space. This problem is one of multidimensional nonlinear mapping.

3.2 Mechanism Models

It is natural to think about the operation of a mechanism in a moving axes sense - because most mechanisms are built that way. That is, the position and orientation of any link in a kinematic chain is dependent on all other joints that come before it in the sequence.

Conventionally, one thinks in terms of moving axis operations applied to frames embedded in the mechanism which move the first frame sequentially into coincidence with all of the others in the mechanism. Then, a sequence of operators is written to represent the mechanism kinematics. The conventional rules for modelling a sequence of connected joints are as follows:

- assign embedded frames to the links in sequence such that the operations which move each frame into coincidence with the next are a function of the appropriate joint variable
- write the operator matrices which correspond to these operations in *left to right order*

This process will generate the matrix that represents the position of the last embedded frame with respect to the first, or equivalently, which converts the coordinates of a point from the last to the first. This matrix will be called the **mechanism model**.

3.3 Modelling a Mechanism

In 1955, J. Denavit and R. S. Hartenberg [3] first proposed the use of homogeneous transforms to represent the articulations of a mechanism, and this form of model has been used almost universally since. A mechanism is considered to be any collection of joints, either linear or rotary, joined together by links. The total number of movable joints is called the number of **degrees of freedom**.
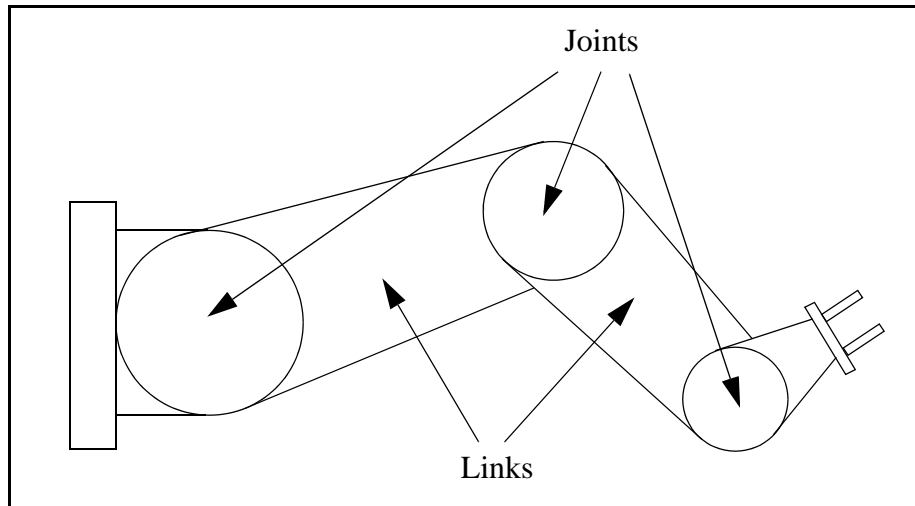

3.4 Denavit Hartenberg Convention for Mechanisms

It is conventional in many aspects of robotics to use a special product of four fundamental operators as a basic conceptual unit. The resulting single matrix can represent the most general spatial relationship between two frames. This convention with a small number of associated rules for assignment of embedded frames has come to be called the Denavit Hartenberg (DH) convention.

For two frames positioned in space, the directions of, say, the z axes of two frames define two lines in 3D space. For nonparallel axes, there is a well defined measure of the distance between them given by the length of their mutual perpendicular. Let the first frame be called frame i-1 and the second frame i. The first can be moved into coincidence with the second by a sequence of 4 operations:

- rotate around the x_{i-1} axis by an angle θ_i
- translate along the x_{i-1} axis by a distance u_i
- rotate around the new z axis by an angle ψ_i
- translate along the new z axis by a distance w_i

The following figure indicates this sequence graphically.
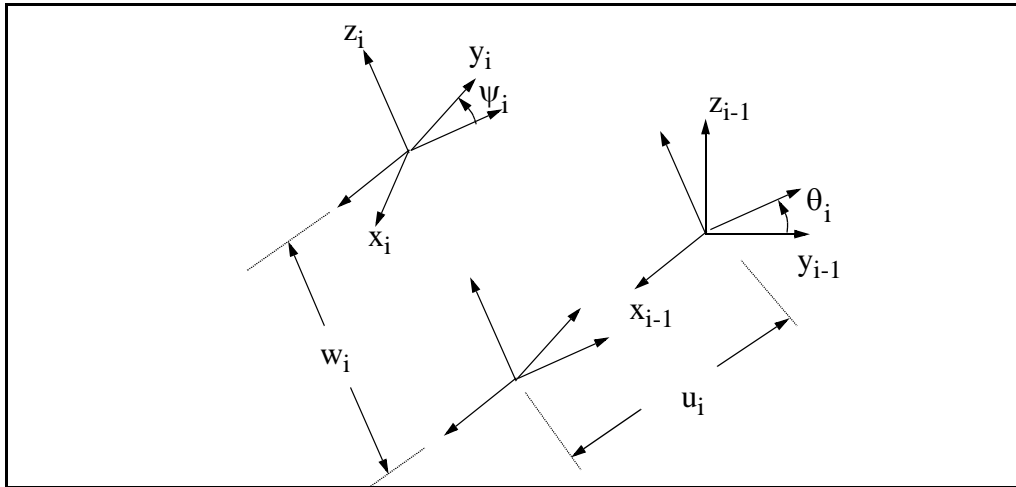
The matrix, conventionally called A_i, that converts coordinates from frame i to frame i-1 can be written using the rules for forward kinematic modelling given previously.

$$A_i = T_i^{i-1} = \mathrm{Rotx}(\theta_i)\mathrm{Trans}(u_i, 0, 0)\mathrm{Rotz}(\psi_i)\mathrm{Trans}(0, 0, w_i)$$

$$A_i = T_i^{i-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta_i & -s\theta_i & 0 \\ 0 & s\theta_i & c\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & u_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\psi_i & -s\psi_i & 0 & 0 \\ s\psi_i & c\psi_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & w_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i = T_i^{i-1} = \begin{bmatrix} c\psi_i & -s\psi_i & 0 & u_i \\ c\theta_i s\psi_i & c\theta_i c\psi_i & -s\theta_i & -s\theta_i w_i \\ s\theta_i s\psi_i & s\theta_i c\psi_i & c\theta_i & c\theta_i w_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix has the following interpretations:

- its columns represent the axes and origin of frame i expressed in frame i-1 coordinates
- it converts coordinates from frame i to frame i-1

3.5 Example - 3 Link Planar Manipulator

Although the homogeneous transform matrices can represent any complex 3D mechanism, the process of writing forward kinematic equations will be illustrated for a simple 2D manipulator composed of three rotary joints, three links and an end effector. Coordinate frames are assigned to the center of each rotary joint and the datum angle for each is set to zero as illustrated in the figure below.

It is conventional to assign the z axis of a frame so that the associated degree of freedom, linear or rotary, coincides with it. Also, the x axis is normally chosen so that it points along the mutual perpendicular. In this case, this means that frame 1 is a rotated version of frame 0:
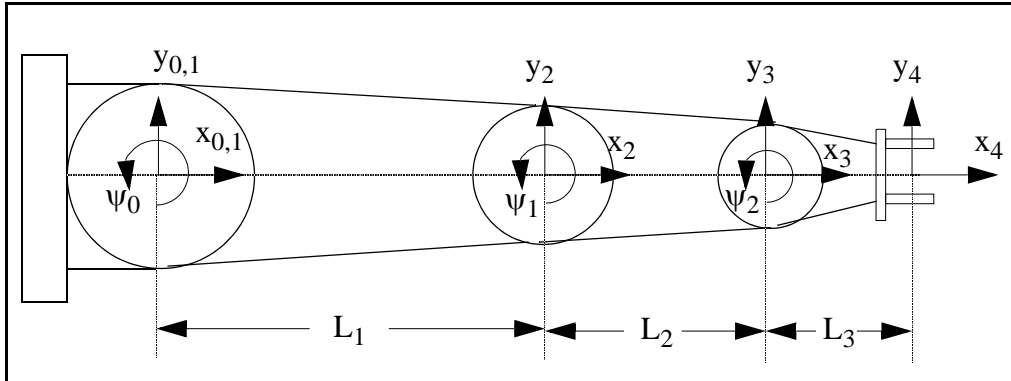
The joint parameters are indicated in the table below:

Table 1: 3 Link Planar Manipulator

Link	θ	u	ψ	w
0	0	0	ψ_1	0
1	0	L_1	ψ_2	0
2	0	L_2	ψ_3	0
3	0	L_3	0	0

This gives the following Denavit Hartenberg matrices. Their inverses are computed from the

inverse formula because they will be useful later:

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_1^{-1} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_2^{-1} = \begin{bmatrix} c_2 & s_2 & 0 & -c_2 L_1 \\ -s_2 & c_2 & 0 & s_2 L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_3^{-1} = \begin{bmatrix} c_3 & s_3 & 0 & -c_3 L_2 \\ -s_3 & c_3 & 0 & s_3 L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_4^{-1} = \begin{bmatrix} 1 & 0 & 0 & -L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The position and orientation of the end effector with respect to the base is given by:

$$T_4^0 \,=\, T_1^0 T_2^1 T_3^2 T_4^3 \,=\, A_1 A_2 A_3 A_4$$

$$T_4^0 \,=\, \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 & L_1 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_3 & -s_3 & 0 & L_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^0 \,=\, \begin{bmatrix} c_{123} & -s_{123} & 0 & (c_{123}L_3 + c_{12}L_2 + c_1 L_1) \\ s_{123} & c_{123} & 0 & (s_{123}L_3 + s_{12}L_2 + s_1 L_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Inverse Kinematics

Inverse kinematics is the problem of finding the joint parameters given only the numerical values of the homogeneous transforms which model the mechanism. In practice, this is useful because it solves the problem of where to drive the joints in order to get the hand of an arm or the foot of a leg in the right place, at the right orientation.

Proficiency with inverse kinematics requires a degree of skill and practice and there are few general guidelines that can be given. This section will discuss this difficult problem and continue with the simple example.

4.1 Existence and Uniqueness

The inverse kinematic problem is considerably more difficult than the forward one because it involves the solution of nonlinear equations. In general, there is no guarantee that nonlinear equations have a solution, and if they do, it may not be unique. It has been shown theoretically by Pieper [13], that a six degree of freedom manipulator for which the last three joints intersect at a point is always solvable. Most manipulators are constructed in this way, so most are solvable.

Any real mechanism has a finite reach, so it can only achieve positions in a region of space known as the **workspace**. Unless the initial transform to be solved is in the workspace, the inverse kinematic problem will have no solution. This manifests itself, for example, as inverse sines and cosines of arguments greater than one. Further, mechanisms with many rotary joints will often have more than one solution, which will exhibit various forms of symmetry. The latter case is known as **redundancy**.

4.2 Technique

This problem would be very difficult to solve without the discipline afforded by the use of homogeneous transforms. Using the DH convention, a mechanism can be "solved" more or less one joint at a time by a process of rewriting the forward kinematics equations in several different ways.

Any DH matrix has only six degrees of freedom. The rotation matrix part is constrained to be orthonormal. That is, the rows or columns must be of unit length and mutually perpendicular. The bottom row is constrained to contain zeros and ones. The position vector contains three independent values.

The inverse kinematic problem is solved by rewriting the forward transform in many different ways in an attempt to isolate unknowns. Although there are only 6 independent relationships, they can be rewritten in many different ways.

Using a 3 degree of freedom mechanism for example, the forward kinematics can be written in all

of these ways by premultiplying or postmultiplying by the inverse of each link transform in turn:

$$T_4^0 = A_1 A_2 A_3 A_4 \qquad\qquad T_4^0 = A_1 A_2 A_3 A_4$$

$$A_1^{-1} T_4^0 = A_2 A_3 A_4 \qquad\qquad T_4^0 A_4^{-1} = A_1 A_2 A_3$$

$$A_2^{-1} A_1^{-1} T_4^0 = A_3 A_4 \qquad\qquad T_4^0 A_4^{-1} A_3^{-1} = A_1 A_2$$

$$A_3^{-1} A_2^{-1} A_1^{-1} T_4^0 = A_4 \qquad\qquad T_4^0 A_4^{-1} A_3^{-1} A_2^{-1} = A_1$$

$$A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} T_4^0 = I \qquad\qquad T_4^0 A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1} = I$$

These are, of course, redundant specifications of the same set of equations, so they contain no new information. However, they often generate equations which are easy to solve because the DH convention tends to isolate each joint. It is conventional to use the left column of equations, but the right column is easier to follow in our case, so it will be used.

4.3 Example - 3 Link Planar Manipulator

The process is started by assuming that the forward kinematic solution is known. Names are assigned to each of its elements as follows:

$$T_n^0 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For our example, we know that some of these are zeros and ones, so that the end effector frame is:

$$T_4^0 = \begin{bmatrix} r_{11} & r_{12} & 0 & p_x \\ r_{21} & r_{22} & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The first equation is already known:

$$T_4^0 = A_1 A_2 A_3 A_4$$

$$\begin{bmatrix} r_{11} & r_{12} & 0 & p_x \\ r_{21} & r_{22} & 0 & p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & (c_{123}L_3 + c_{12}L_2 + c_1 L_1) \\ s_{123} & c_{123} & 0 & (s_{123}L_3 + s_{12}L_2 + s_1 L_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (2,1) and (1,1) elements we have:

$$\psi_{123} = atan2(r_{21}, r_{11})$$

The next equation is:

$$T_4^0 A_4^{-1} = A_1 A_2 A_3$$

$$\begin{bmatrix} r_{11} & r_{12} & 0 & -r_{11}L_3 + p_x \\ r_{21} & r_{22} & 0 & -r_{21}L_3 + p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & c_1(c_2 L_2 + L_1) - s_1(s_2 L_2) \\ s_{123} & c_{123} & 0 & s_1(c_2 L_2 + L_1) + c_1(s_2 L_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} r_{11} & r_{12} & 0 & -r_{11}L_3 + p_x \\ r_{21} & r_{22} & 0 & -r_{21}L_3 + p_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{123} & -s_{123} & 0 & c_{12}L_2 + c_1 L_1 \\ s_{123} & c_{123} & 0 & s_{12}L_2 + s_1 L_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (1,4) and (2,4) elements:

$$k_1 = -r_{11}L_3 + p_x = c_{12}L_2 + c_1 L_1$$
$$k_2 = -r_{21}L_3 + p_y = s_{12}L_2 + s_1 L_1$$

These can be squared and added to yield:

$$k_1^2 + k_2^2 = L_2^2 + L_1^2 + 2L_2L_1(c_1c_{12} + s_1s_{12})$$

$$k_1^2 + k_2^2 = L_2^2 + L_1^2 + 2L_2L_1c_2$$

And this gives the angle ψ_2 as:

$$\psi_2 = acos\left[\frac{(k_1^2 + k_2^2) - (L_2^2 + L_1^2)}{2L_2L_1}\right]$$

The result implies that there are two solutions for this angle which are symmetric about zero. These correspond to the elbow up and elbow down configurations. Now, before the expressions were reduced to include a sum of angles, they were:

$$k_1 = -r_{11}L_3 + p_x = c_1(c_2L_2 + L_1) - s_1(s_2L_2)$$

$$k_2 = -r_{21}L_3 + p_y = s_1(c_2L_2 + L_1) + c_1(s_2L_2)$$

With ψ_2 now known, these can be written as:

$$c_1k_3 - s_1k_4 = k_1$$

$$s_1k_3 + c_1k_4 = k_2$$

This is one of the standard forms that recur in inverse kinematics problems for which the solution is:

$$\psi_1 = atan2[(k_2k_3 - k_1k_4), (k_1k_3 + k_2k_4)]$$

Finally, the last angle is:

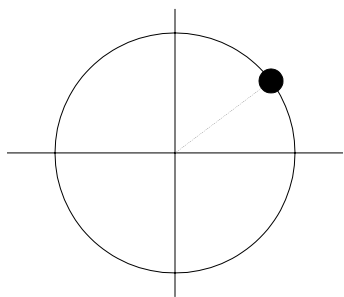$$\psi_3 = \psi_{123} - \psi_2 - \psi_1$$

4.4 Standard Forms

There are a few forms of trigonometric equations that recur is most mechanisms. The following set of solutions is sufficient for most applications. One of the most difficult forms, solved by square and add, was presented in the example. In the following, the letters a, b, and c represent arbitrary known expressions.

4.4.1 Explicit Tangent

This form generates a single solution because both the sine and cosine are fixed in value. It can arise in the last two joints of a three axis wrist, for example. The equation:

$$a = c_n$$
$$b = s_n$$

has the trivial solution:
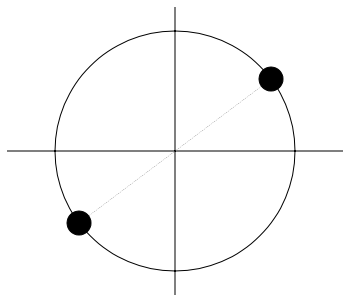
$$\psi_n = atan2(b, a)$$

4.4.2 Point Symmetric Redundancy

This form generates two solutions that are symmetric about the origin. It can arise in the shoulder or hip joint, for example. The equation:

$$s_n a - c_n b = 0$$

can be solved by isolating the ratio of the two trig functions. There are two angles in one revolution which have the same tangent so the two solutions are:

$$\psi_n = atan2(b, a)$$
$$\psi_n = atan2(-b, -a)$$

4.4.3 Line Symmetric Redundancy

This form generates two solutions that are symmetric about an axis because the sine or cosine of the deviation from the axis must be constant. The sine case will be illustrated. It can arise in the shoulder or hip joint, or in an elbow or knee joint, for example. The equation:

$$s_n a - c_n b = c$$

can be solved by the trig substitution:

$$a = r\cos(\theta) \qquad b = r\sin(\theta)$$

where:

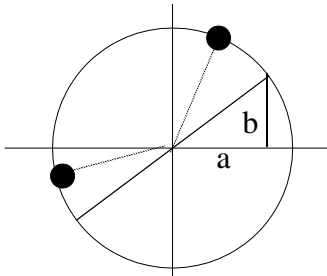$$r = \pm\text{sqrt}(a^2 + b^2) \qquad \theta = atan2(b, a)$$

This gives:

$$s_n c\theta - c_n s\theta = c/r$$
$$s(\theta - \psi_n) = c/r$$

So the cosine is:

$$c(\theta - \psi_n) = \pm\text{sqrt}(1 - (c/r)^2)$$

This gives the result:

$$\psi_n = atan2(b, a) - atan2[c, \pm\text{sqrt}(r^2 - c^2)]$$

5. Differential Kinematics

Differential kinematics is the study of the derivatives of kinematic models. These derivatives are called **Jacobians** and they have many uses ranging from:

- resolved rate control
- sensitivity analysis
- uncertainty propagation
- static force transformation

5.1 Derivatives of Fundamental Operators

The derivatives of the fundamental operators with respect to their own parameters will be important. They can be used to compute derivatives of very complex expressions by using the chain rule of differentiation. For reference, they are quoted below:

$$\frac{\partial}{\partial u}\text{Trans(u, v, w)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \frac{\partial}{\partial \theta}\text{Rotx}(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -s\theta & -c\theta & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial}{\partial v}\text{Trans(u, v, w)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \frac{\partial}{\partial \phi}\text{Roty}(\phi) = \begin{bmatrix} -s\phi & 0 & c\phi & 0 \\ 0 & 1 & 0 & 0 \\ -c\phi & 0 & -s\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial}{\partial w}\text{Trans(u, v, w)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \frac{\partial}{\partial \psi}\text{Rotz}(\psi) = \begin{bmatrix} -s\psi & -c\psi & 0 & 0 \\ c\psi & -s\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Similar expressions can be generated for the fundamental transforms.

5.2 The Mechanism Jacobian

If a sequence of joints can be represented by the product of a series of homogeneous transforms, it is natural to ask about the effect of a differential change in joint variables on the position and orientation of the end of the mechanism. A frame matrix represents orientation indirectly in terms of three unit vectors, so an extra set of equations is required to extract three angles from the rotation matrix in order to represent orientation. In terms of position, however, the last column of the mechanism model gives the position of the end effector with respect to the base.

In general, let a mechanism have variables represented by the vector $\bar{q}$, and let the position and orientation, or **pose**, of the end of the mechanism be given by the vector $\bar{x}$. Then, the end effector

position is given by:

$$\bar{x} = \bar{F}(\bar{q})$$

where the nonlinear multidimensional function $\bar{F}$ comes from the mechanism model. The Jacobian matrix is a multidimensional derivative defined as:

$$J = \frac{\partial}{\partial \bar{q}} \bar{x} = \frac{\partial}{\partial \bar{q}}(\bar{F}(\bar{q})) = \left[\frac{\partial x_i}{\partial q_j}\right] = \begin{bmatrix} \dfrac{\partial x_1}{\partial q_1} & \cdots & \dfrac{\partial x_n}{\partial q_n} \\ \cdots & \cdots & \cdots \\ \dfrac{\partial x_n}{\partial q_1} & \cdots & \dfrac{\partial x_n}{\partial q_n} \end{bmatrix}$$

The differential mapping from small changes in $\bar{q}$ to the corresponding small changes in $\bar{x}$ is:

$$d\bar{x} = J d\bar{q}$$

The Jacobian also gives velocity relationships via the chain rule of differentiation as follows:

$$\frac{d}{dt}\bar{x} = \left(\frac{\partial}{\partial \bar{q}}\bar{x}\right)\left(\frac{d}{dt}\bar{q}\right)$$

which maps joint rates onto end effector velocity. Note that the Jacobian is nonlinear in the joint variables, but linear in the joint rates. This implies that reducing the joint rates by half reduces the end velocity by exactly half and preserves the direction.

5.3 Singularity

Redundancy takes the form of **singularity** of the Jacobian matrix in the differential kinematic solution. A mechanism can lose one or more degrees of freedom:

  • at points where two different inverse kinematic solutions converge
  • when joint axes become aligned or parallel
  • when the boundaries of the workspace are reached

Singularity implies that the Jacobian loses rank and is not invertible. At the same time, the inverse kinematic solution tends to fail because axes become aligned, and infinite rates can be generated by rate control laws.

5.4 Example - 3 Link Planar Manipulator

For the example manipulator, the last column of the manipulator model gives the following two equations:

$$x = (c_{123}L_3 + c_{12}L_2 + c_1L_1)$$

$$y = (s_{123}L_3 + s_{12}L_2 + s_1L_1)$$

which can be differentiated with respect to ψ_1, ψ_2, and ψ_3 in order to determine the velocity of the end effector as the joints move. The solution is:

$$\dot{x} = -(s_{123}\dot{\psi}_{123}L_3 + s_{12}\dot{\psi}_{12}L_2 + s_1\dot{\psi}_1L_1)$$

$$\dot{y} = (c_{123}\dot{\psi}_{123}L_3 + c_{12}\dot{\psi}_{12}L_2 + c_1\dot{\psi}_1L_1)$$

which can be written as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} (-s_{123}L_3 - s_{12}L_2 - s_1L_1) & (-s_{123}L_3 - s_{12}L_2) & -s_{123}L_3 \\ (c_{123}L_3 + c_{12}L_2 + c_1L_1) & (c_{123}L_3 + c_{12}L_2) & c_{123}L_3 \end{bmatrix} \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{bmatrix}$$

5.5 Jacobian Determinant

It is known from the implicit function theorem of calculus that the ratio of differential volumes between the domain and range of a multidimensional mapping is given by the Jacobian determinant. This quantity has applications to a technique of navigation and ranging called **triangulation**.

Thus the product of the differentials forms a volume in both configuration space and in task space. The relationship between them is:

$$(dx_1dx_2...dx_n) = |J|(dq_1dq_2...dq_m)$$

5.6 Jacobian Tensor

At times it is convenient to compute the derivative of a transform matrix with respect to a vector of variables. In this case, the result is the derivative of a matrix with respect to a vector. For example:

$$\frac{\partial}{\partial \bar{q}}[T(\bar{q})] = \left[\frac{\partial T_{ij}}{\partial q_k}\right]$$

This is a three dimensional cube of numbers which can loosely be called a tensor. The mechanism model itself is a matrix function of a vector.

For example, if:

$$T(\bar{q}) = A_1(q_1)A_2(q_2)A_3(q_3)$$

Then there are three slices of the tensor, each a matrix, given by:

$$\frac{\partial T}{\partial q_1} = \frac{\partial A_1}{\partial q_1}A_2A_3$$

$$\frac{\partial T}{\partial q_2} = A_1\frac{\partial A_2}{\partial q_2}A_3$$

$$\frac{\partial T}{\partial q_3} = A_1A_2\frac{\partial A_3}{\partial q_3}$$

6. Vehicle Kinematics

Some aspects of the kinematics of moving vehicles are simpler than that of mechanisms, others are more difficult. This section presents an abbreviated discussion of the kinematic transforms necessary for control of a vehicle in 3D.

6.1 Axis Conventions

There are at least two prevailing conventions for the assignment of axes for the coordinate systems of vehicles. In aerospace vehicles, it is conventional to point the z axis downward, and this makes it natural to point the x axis forward and the y axis out the right side of the vehicle. The convention used here is that the z axis points up, y forward, and x out the right side. This has the advantage that the projection of 3D information onto the x-y plane is more natural.

It is important to note that the form of rotation matrices depends firstly on the order of their component rotations, and secondly on the linear axis conventions. The convention used here *corresponds to a z-x-y Euler angle sequence*. Therefore, it is not advisable to use the homogeneous transforms developed here until they are verified to be correct for any sensors and actuators that are used.

6.2 Frame Assignment

Several coordinate frames are important for moving vehicles. For legged vehicles, the frames embedded in the legs are assigned according to the DH convention and the frames for the rest of the system are analogous to those for wheeled vehicles. This section will present a set of frames for wheeled vehicles that occur in most applications.

These frames are indicated in the figure below:
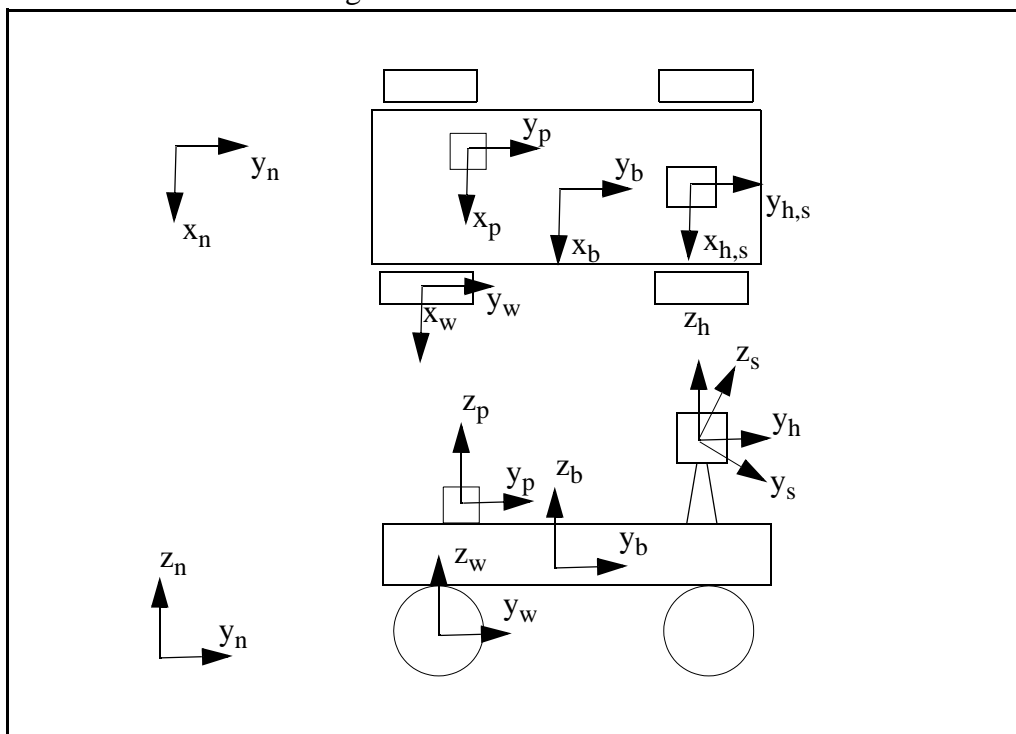
6.2.1   The Navigation Frame

This is the coordinate system in which the vehicle position and attitude is ultimately required. Usually, this frame is taken as locally level (i.e. the z axis is perfectly aligned with the local gravity vector, not the local terrain tangent plane). The z, up, or azimuth axis is aligned with the gravity vector, the y, or north axis is aligned with the geographic pole[1], and the x axis points east to complete a right-handed system. In some applications, any frame that is fixed on the earth is satisfactory whether or not it is aligned with the earth's fields. This frame is identified by the letter n.

6.2.2   The Body Frame

The body frame is positioned at the point on the vehicle body which is most convenient and is considered to be fixed in attitude with respect to the vehicle body. For Ackerman steer vehicles, the center of rear axle is a natural place for this frame. In some applications, the best estimate of the position of the center of gravity is more appropriate. The z axis points up, y forward, and x out the right side. This frame is identified by the letter b.

6.2.3   The Positioner Frame

This frame is positioned at the point on or near any position estimation system which reports ***its own position***. If the system generates attitude and attitude rates only, this frame is not required because the attitude of the device will also be that of the vehicle. For an INS, this is typically the center of the IMU and for GPS it is the phase center of the antenna[2]. Axes directions are defined similarly to the body frame directions. There is a different positioner frame for each positioning device. This frame is identified by the letter p.

6.2.4   The Sensor Head Frame

Sometimes, environmental perception sensors are mounted on stabilized platforms or on pan/tilt mechanisms. These provide isolation of the sensor attitude from that of the vehicle and/or the ability of the system to literally point its "head". Axes directions are defined similarly to the body frame directions. In cases where the rotary axes of the device all intersect at a point, this frame is positioned at the common point of intersection of these axes. Axes directions are defined similarly to the body frame directions.

In cases where the environmental perception sensor axes are not aligned with respect to those of the vehicle (for example, when the sensor looks downward), or in cases where there is a misalignment which must be accounted for, a ***rigid sensor head*** can be defined which tilts the body axes into coincidence with those of the sensor. This frame is identified by the letter h.

6.2.5   The Sensor Frame

This frame is positioned at the center of the environmental perception sensor with axis definitions similar to the body frame when the sensor points forward. For video cameras, it is positioned on the optical axis at the image plane behind the lens. For stereo systems, it is positioned either between both cameras or is associated with the image plane of one of them. For imaging laser

---

1. The geographic pole is determined by the earth's spin axis, not the magnetic field.
2. The antenna may be nowhere near the GPS receiver.

rangefinders, it is positioned as the average point of convergence of the rays through each pixel. Axes directions are defined similarly to the body frame directions. There is a different sensor frame for each sensor.

6.2.6   The Wheel Frame

This frame is positioned at the center of the wheel, on the axle. At times it may be convenient to define this frame at the wheel contact point and it may be useful to have it rotate with the wheel. Each wheel has its own frame.

6.3   The RPY Transform

For the purpose of kinematic modelling, any of the former frames can be considered to be embedded in a rigid body. It turns out that the attitude of a rigid body can be expressed in many different ways. It is usually most convenient to express vehicle attitude in terms of three special angles called **roll**, **pitch**, and **yaw**. Luckily, most pan/tilt mechanisms are kinematically formed from a yaw rotation followed by a pitch with no roll, so they are a degenerate form of a more general transform.

These considerations imply that a general homogeneous transform, called the RPY transform, can be formed which is similar in principle to the DH matrix, except that it has three rotations, and which can serve to transform between the body frame and all others. There are six degrees of freedom involved, three translations and three rotations, and each can be either a parameter or a variable.

Let two general frames be defined called 'a' and 'b' and consider the moving axis operations which transform frame 'a' into coincidence with frame 'b'. *In order*, these are:

- translate along the (x,y,z) axes of frame 'a' by (u,v,w) until its origin coincides with that of frame 'b'
- rotate about the new z axis by an angle ψ called **yaw**
- rotate about the new x axis by an angle θ called **pitch**
- rotate about the new y axis by an angle ϕ called **roll**

Angles are measured counterclockwise positive according to the right hand rule. These operations

are indicated below for the case of transforming the navigation frame into the body frame.
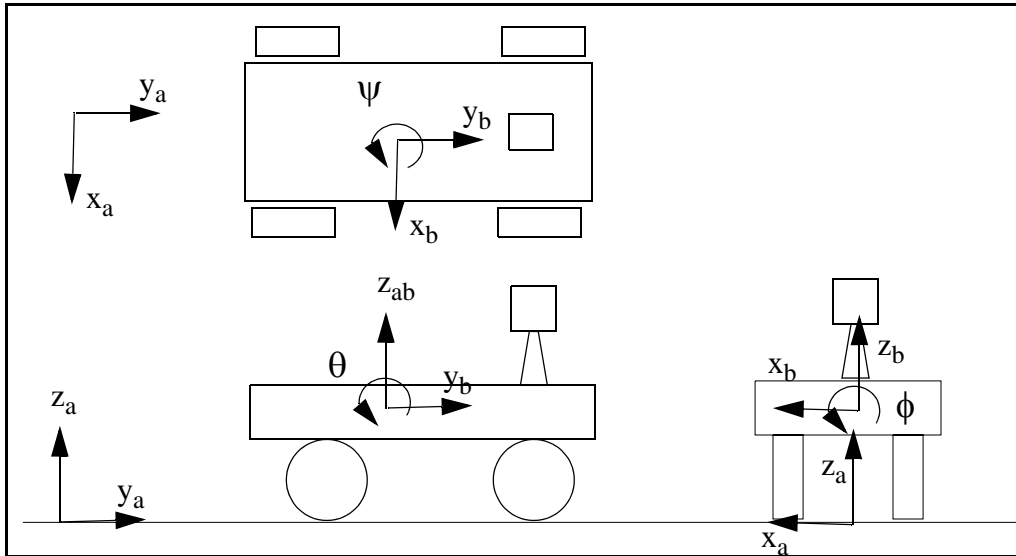
The forward kinematic transform that represents this sequence of operations is, according to our rules for forward kinematics:

$$T_b^a = \text{Trans}(u, v, w)\text{Rotz}(\psi)\text{Rotx}(\theta)\text{Roty}(\phi)$$

$$T_b^a = \begin{bmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\psi & -s\psi & 0 & 0 \\ s\psi & c\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\theta & -s\theta & 0 \\ 0 & s\theta & c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_b^a = \begin{bmatrix} (c\psi c\phi - s\psi s\theta s\phi) & -s\psi c\theta & (c\psi s\phi + s\psi s\theta c\phi) & u \\ (s\psi c\phi + c\psi s\theta s\phi) & c\psi c\theta & (s\psi s\phi - c\psi s\theta c\phi) & v \\ -c\theta s\phi & s\theta & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix has the following two interpretations[1]:

 • its columns represent the axes and origin of frame 'b' expressed in frame 'a' coordinates
 • it converts coordinates from frame 'b' to frame 'a'

The matrix can be considered to be the conversion from a pose to a coordinate frame.

---

1. This transform applies to the Litton INS on the HMMWV, the Stagget stable platform, and the ROS pan/tilt head.

6.4 Frame Parameters for Wheeled Vehicles

Using the RPY transform, it is now possible to specify the parameters and variables which permit coordinate transformation from anywhere on a vehicle to anywhere else. This is accomplished by specifying the six degrees of freedom in a table. In the table, var indicates a variable and fixed indicates a fixed parameter:

Table 2: Frame Parameters for Wheeled Vehicles

Transform	u	v	w	ψ	θ	ϕ
T_b^n	var	var	var	var	var	var
T_h^b	fixed	fixed	fixed	var	var	fixed
T_s^h	fixed	fixed	fixed	fixed	fixed	fixed
T_p^b	fixed	fixed	fixed	fixed	fixed	fixed
T_w^b	fixed	fixed	fixed	fixed	var	fixed

6.5 Inverse Kinematics for the RPY Transform

The inverse kinematic solution to the RPY transform has at least two uses:

- it gives the angles to which to drive a sensor head, or a directional antenna given the direction cosines of the goal frame
- it gives the attitude of the vehicle given the body frame axes, which often correspond to the local tangent plane to the terrain over which it moves

This solution can be considered to be the procedure for extracting a pose from a coordinate frame. There are many different ways to get the solution from different elements of the RPY transform. The one used here is useful for modelling terrain following of a vehicle. Proceeding as for a mechanism, the elements of the transform are assumed to be known:

$$T_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The premultiplication set of equations will be used. The first equation is:

$$T_b^a = \text{Trans}(u, v, w)\text{Rotz}(\psi)\text{Rotx}(\theta)\text{Roty}(\phi)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (c\psi c\phi - s\psi s\theta s\phi) & -s\psi c\theta & (c\psi s\phi + s\psi s\theta c\phi) & u \\ (s\psi c\phi + c\psi s\theta s\phi) & c\psi c\theta & (s\psi s\phi - c\psi s\theta c\phi) & v \\ -c\theta s\phi & s\theta & c\theta c\phi & w \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The translational elements are trivial. From the (1,2) and (2,2) elements:

$$\psi = atan2(r_{22}, -r_{12})$$

This implies that yaw can be determined from a vector which is known to be aligned with the body y axis. The second equation is:

$$[\text{Trans}(u, v, w)]^{-1}T_b^a = \text{Rotz}(\psi)\text{Rotx}(\theta)\text{Roty}(\phi)$$

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (c\psi c\phi - s\psi s\theta s\phi) & -s\psi c\theta & (c\psi s\phi + s\psi s\theta c\phi) & 0 \\ (s\psi c\phi + c\psi s\theta s\phi) & c\psi c\theta & (s\psi s\phi - c\psi s\theta c\phi) & 0 \\ -c\theta s\phi & s\theta & c\theta c\phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which generates nothing new. The next equation is:

$$[\text{Rotz}(\psi)]^{-1}[\text{Trans}(u, v, w)]^{-1}T_b^a = \text{Rotx}(\theta)\text{Roty}(\phi)$$

$$\begin{bmatrix} (r_{11}c\psi + r_{21}s\psi) & (r_{12}c\psi + r_{22}s\psi) & (r_{13}c\psi + r_{23}s\psi) & 0 \\ (-r_{11}s\psi + r_{21}c\psi) & (-r_{12}s\psi + r_{22}c\psi) & (-r_{13}s\psi + r_{23}c\psi) & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\phi & 0 & s\phi & 0 \\ s\theta s\phi & c\theta & -s\theta c\phi & 0 \\ -c\theta s\phi & s\theta & c\theta c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (2,2) and (3,2) elements:

$$\theta = atan2(r_{32}, -r_{12}s\psi + r_{22}c\psi)$$

Which implies that pitch can also be determined from a vector known to be aligned with the body

y axis. A good solution for ϕ is available from the (1,1) and (1,3) elements. However, for reasons of convenience, the solution will be delayed until the next equation. The next equation is:

$$[\text{Rotx}(\theta)]^{-1}[\text{Rotz}(\psi)]^{-1}[\text{Trans}(u, v, w)]^{-1}T_b^a = \text{Roty}(\phi)$$

$$\begin{bmatrix} (r_{11}c\psi + r_{21}s\psi) & (r_{12}c\psi + r_{22}s\psi) & (r_{13}c\psi + r_{23}s\psi) & 0 \\ c\theta[-r_{11}s\psi + r_{21}c\psi] + r_{31}s\theta & c\theta[-r_{12}s\psi + r_{22}c\psi] + r_{32}s\theta & c\theta[-r_{13}s\psi + r_{23}c\psi] + r_{33}s\theta & 0 \\ -s\theta[-r_{11}s\psi + r_{21}c\psi] + r_{31}c\theta & -s\theta[-r_{12}s\psi + r_{22}c\psi] + r_{32}c\theta & -s\theta[-r_{13}s\psi + r_{23}c\psi] + r_{33}c\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$=$$

$$\begin{bmatrix} c\phi & 0 & s\phi & 0 \\ 0 & 1 & 0 & 0 \\ -s\phi & 0 & c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

From the (1,1) and (3,1) elements:

$$\phi = atan2(s\theta[-r_{11}s\psi + r_{21}c\psi] - r_{31}c\theta, (r_{11}c\psi + r_{21}s\psi))$$

This implies that roll can be derived from a vector known to be aligned with the body x axis.

6.6 Angular Velocity

The roll, pitch, and yaw angles are, as we have defined them, measured about moving axes. Therefore, they are a sequence of **Euler angles**, specifically, the z-x-y sequence[1]. The Euler angle definition of vehicle attitude has the disadvantage that the roll, pitch, and yaw angles are not the quantities that are actually indicated by strapped down vehicle mounted sensors such as gyros.

The relationship between the rates of the Euler angles and the angular velocity vector is nonlinear. The angles are measured neither about the body axes nor about the navigation frame axes. It is important to know the exact relationship between the two because it provides the basis for determining vehicle attitude from angular rate measurements.

In order to determine the angular velocity, consider that the total angular velocity is the sum of three components, each measured about one of the intermediate axes in the chain of rotations which bring the navigation frame into coincidence with the body frame. Using the fundamental transforms, each of the three rotation rates are transformed into the body frame by the remaining

---

1. The sequence depends on the convention for assigning the directions of the linear axes.

rotations in the sequence to give the result in the body frame.

$$\overline{\omega}^b = \begin{bmatrix} 0 \\ \dot{\phi} \\ 0 \end{bmatrix} + \text{rot}(y, \phi)\begin{bmatrix} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} + \text{rot}(y, \phi)\text{rot}(x, \theta)\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

$$\overline{\omega}^b = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} c\phi\dot{\theta} - s\phi c\theta\dot{\psi} \\ \dot{\phi} + s\theta\dot{\psi} \\ s\phi\dot{\theta} + c\phi c\theta\dot{\psi} \end{bmatrix} = \begin{bmatrix} c\phi & 0 & -s\phi c\theta \\ 0 & 1 & s\theta \\ s\phi & 0 & c\phi c\theta \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix}$$

This result gives the vehicle angular velocity expressed in the body frame in terms of the Euler angle rates. Notice that when the vehicle is level the x and y components are zero and the z component is just the yaw rate as expected.

This relationship is also very useful in its inverted form. One can verify by substitution that:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \omega_x c\phi + \omega_z s\phi \\ \omega_y - t\theta[\omega_z c\phi - \omega_x s\phi] \\ [\omega_z c\phi - \omega_x s\phi]/c\theta \end{bmatrix} = \begin{bmatrix} c\phi & 0 & s\phi \\ t\theta s\phi & 1 & -t\theta c\phi \\ -\dfrac{s\phi}{c\theta} & 0 & \dfrac{c\phi}{c\theta} \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

$$\text{because} \quad \omega_z c\phi - \omega_x s\phi = c\theta\dot{\psi}$$
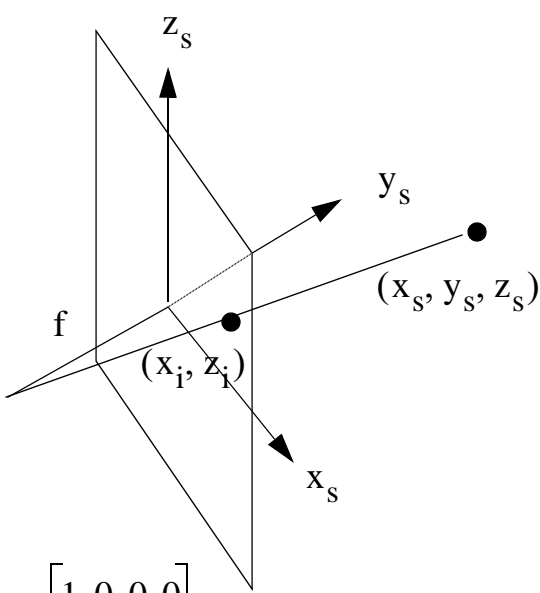
7. Sensor Kinematics

Many sensors used on robot vehicles are of the imaging variety. For this class of sensors, the process of image formation must be modelled. Image coordinates are identified by the letter i. Typically, these transformations are not linear, and hence cannot be modelled by homogeneous transforms. This section provides the transforms necessary for modelling such sensors. These transforms include projection, reflection, and polar coordinates.

7.1 Perspective Projection

In the case of passive imaging systems, a system of lenses forms an image on an array of sensitive elements called a CCD. These systems include traditional video cameras and infra red cameras. The transformation from the sensor frame to the image plane row and column coordinates is the standard perspective projection. This type of transform is unique in two ways:

- it reduces the dimension of the input vector by one and hence it discards information
- it requires a post normalization step where the output is divided by the scale factor in order to re-establish a unity scale factor

This transformation can be derived by similar triangles.

$$x_i = \frac{x_s f}{y_s + f} = \frac{x_s}{1 + y_s/f}$$

$$z_i = \frac{z_s f}{y_s + f} = \frac{z_s}{1 + y_s/f}$$

$$y_i = 0$$

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ w_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{f} & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \qquad P_s^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{f} & 0 & 1 \end{bmatrix}$$

Note in particular that all projections are not invertible. Here, the second row is all zeros, so the matrix is singular. This, of course, is the ultimate source of the difficulty of measuring scene geometry with a single camera. Therefore, the transform is identified by the special capital letter P.

7.2 Matrix Reflection Operator

We know from Snell's law for reflection of a ray that:

- the incident ray, the normal to the surface, and the reflected ray, all lie in the same plane.
- the angle of incidence equals the angle of reflection

From these two rules, we can formulate a very useful matrix operator to reflect a vector off of any surface, given the unit normal to the surface.

$$\vec{v}_r = \vec{v}_i - 2(\vec{v}_i \cdot \hat{n})\hat{n}$$

$$\vec{v}_r = \vec{v}_i - 2(\hat{n} \otimes \hat{n})\vec{v}_i$$
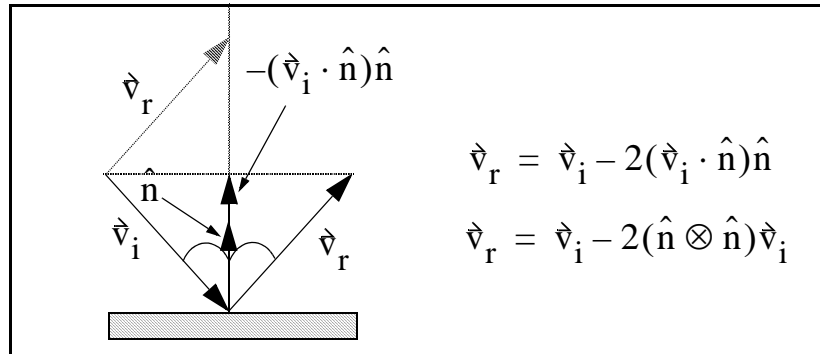
This can be reformulated as a matrix by using the outer product ($\otimes$) of the normal with itself:

$$\vec{v}_r = \text{Ref}(\hat{n})\vec{v}_i \qquad \text{Ref}(\hat{n}) = I - 2(\hat{n} \otimes \hat{n})$$

$$\text{Ref}(\hat{n}) = \begin{bmatrix} 1 - n_x n_x & n_x n_y & n_x n_z \\ n_x n_y & 1 - n_y n_y & n_y n_z \\ n_x n_z & n_y n_z & 1 - n_z n_z \end{bmatrix}$$

The result is expressed in the same coordinates in which both the normal and the incident ray were expressed. This operator can be used to model the "optical" kinematics of laser rangefinders. Notice that a reflection is equivalent to a rotation of twice the angle of incidence about an axis which is normal to the plane formed by the incident ray and the surface normal. A similar matrix refraction operator can be defined.

7.3 Polar Coordinates

For 2D scanning laser rangefinders, the transform from sensor coordinates to image coordinates is the spherical polar transform, and it is not a projection because such a sensor generates a measurement of range as well. This transform is a nonlinear function that cannot be modelled as a matrix.

There are two fundamental possibilities for the kinematics of such sensors because there are two rotations involved which may occur in either order. These are derived in detail in [12]. These transforms are most easily derived in reverse. That is, from image to sensor coordinates. For a sensor where the azimuth rotation comes first, the image to sensor transform is[1]:

$$x_s = Rs\psi$$
$$y_s = Rc\theta c\psi$$
$$z_s = -Rs\theta c\psi$$
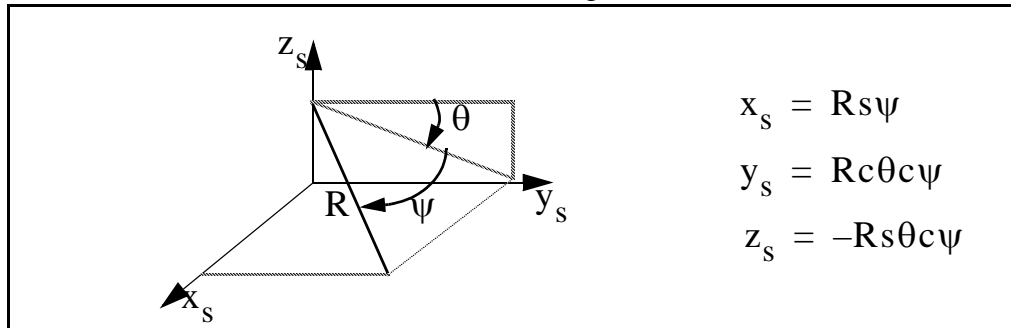
This is easily inverted.

$$
\begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} =
\begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ atan(x_s / \sqrt{y_s^2 + z_s^2}) \\ atan(-z_s / y_s) \end{bmatrix} = h(x_s, y_s, z_s)
$$

---

1. This transform combined with the last two provide the basics of a computer graphics system both in general, and for its particular use in mobile robot simulation.

7.4 Imaging Jacobian

The imaging Jacobian provides the relationship between the differential quantities in the sensor frame and the associated position change in the image. It is important for landmark recognition applications.

7.4.1 Perspective Jacobian

The Jacobian is:

$$x_i = \frac{x_s f}{y_s + f} \qquad y_i = 0 \qquad z_i = \frac{z_s f}{y_s + f}$$

$$J_s^i = \frac{\partial \bar{v}^i}{\partial \bar{v}^s} = \begin{bmatrix} \dfrac{\partial x_i}{\partial x_s} & \dfrac{\partial x_i}{\partial y_s} & \dfrac{\partial x_i}{\partial z_s} \\[2mm] \dfrac{\partial y_i}{\partial x_s} & \dfrac{\partial y_i}{\partial y_s} & \dfrac{\partial y_i}{\partial z_s} \\[2mm] \dfrac{\partial z_i}{\partial x_s} & \dfrac{\partial z_i}{\partial y_s} & \dfrac{\partial z_i}{\partial z_s} \end{bmatrix} = \begin{bmatrix} \dfrac{f}{y_s + f} & \dfrac{-x_s f}{(y_s + f)^2} & 0 \\[2mm] 0 & 0 & 0 \\[2mm] 0 & \dfrac{-z_s f}{(y_s + f)^2} & \dfrac{f}{y_s + f} \end{bmatrix}$$

7.4.2 Azimuth Polar Jacobian

When the azimuth rotation comes first, the Jacobian is:

$$\begin{bmatrix} R \\ \psi \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} & atan\left(\dfrac{x_s}{\sqrt{y_s^2 + z_s^2}}\right) & atan\left(\dfrac{-z_s}{y_s}\right) \end{bmatrix}^T$$

$$J_s^i = \frac{\partial \bar{v}^i}{\partial \bar{v}^s} = \begin{bmatrix} \dfrac{\partial R}{\partial x_s} & \dfrac{\partial R}{\partial y_s} & \dfrac{\partial R}{\partial z_s} \\[2mm] \dfrac{\partial \psi}{\partial x_s} & \dfrac{\partial \psi}{\partial y_s} & \dfrac{\partial \psi}{\partial z_s} \\[2mm] \dfrac{\partial \theta}{\partial x_s} & \dfrac{\partial \theta}{\partial y_s} & \dfrac{\partial \theta}{\partial z_s} \end{bmatrix} = \begin{bmatrix} \dfrac{x_s}{R} & \dfrac{y_s}{R} & \dfrac{z_s}{R} \\[2mm] \dfrac{1}{x_s + \sqrt{y_s^2 + z_s^2}} & \dfrac{-y_s}{x_s + \sqrt{y_s^2 + z_s^2}} & \dfrac{-z_s}{x_s + \sqrt{y_s^2 + z_s^2}} \\[2mm] 0 & \dfrac{z_s}{y_s^2 + z_s^2} & \dfrac{-y_s}{y_s^2 + z_s^2} \end{bmatrix}$$

7.5 Projection Tables

The directions of the rays through each pixel of an imaging sensor are fixed with respect to the sensor frame. The trigonometric overhead of computing this information can be severe in some applications, so it is possible and worthwhile to compute it and store it in tables. These tables can be useful both in real time perception applications and in off-line ray tracing simulation. This section provides the equations necessary to map (row,column) coordinates onto the direction cosines of the ray through those coordinates.

7.5.1 Perspective

Let the width of the image plane be W and the height be H as shown in the figure. Let the horizontal and vertical field of view be $HFOV$ and $VFOV$.
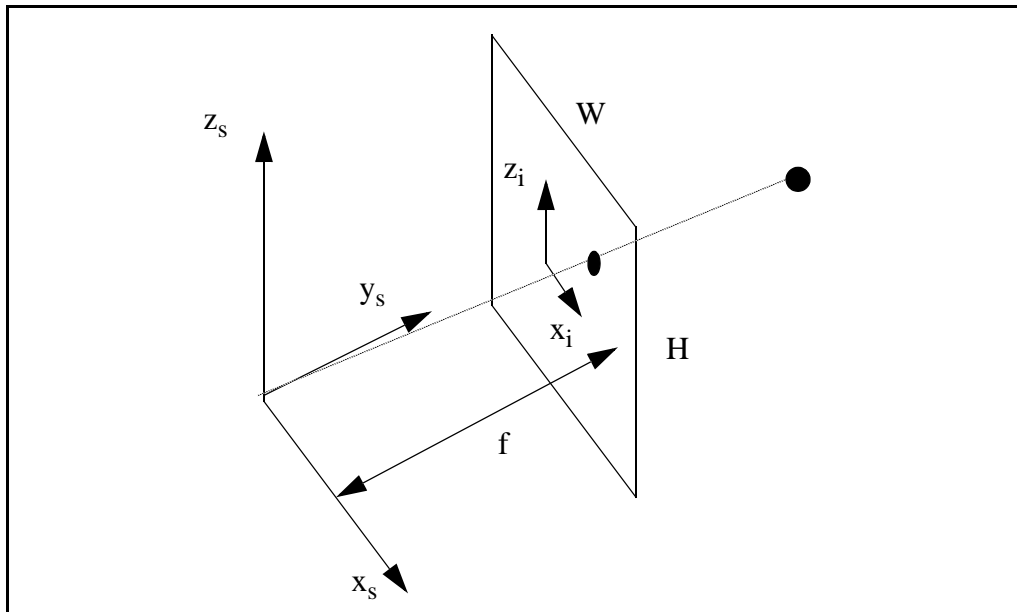
The perspective transformation is:

$$\frac{x_i}{f} = \frac{x_s}{y_s} \qquad\qquad \frac{y_i}{f} = \frac{y_s}{y_s} = 1 \qquad\qquad \frac{z_i}{f} = \frac{z_s}{y_s}$$

A vector from the sensor frame origin to the image plane at the pixel is:

$$\vec{u} = \begin{bmatrix} x_i & f & z_i \end{bmatrix}^T$$

A unit vector in this direction is:

$$\hat{u} \; = \; \frac{1}{\sqrt{1 + (x_i/f)^2 + (z_i/f)^2}}\left[\frac{x_i}{f} \; 1 \; \frac{z_i}{f}\right]$$

The tangents scale linearly with distance along the image plane. Therefore, these are functions of the pixel coordinates as follows:

$$\frac{x_i}{f} \; = \; tan\,\psi \; = \; \frac{(col - cols/2)}{cols}\frac{W}{f}$$

$$\frac{z_i}{f} \; = \; tan\,\theta \; = \; \frac{(row - rows/2)}{rows}\frac{H}{f}$$

When the field of view is small enough to satisfy a small angle assumption, then the focus distance and image plane size can be ignored and their ratios can be approximated as follows:

$$\frac{W}{f} \approx HFOV \qquad\qquad \frac{H}{f} \approx VFOV$$

and this permits using a camera model that is independent of the focal length.

7.5.2 Azimuth Polar

When the azimuth rotation of a scanning mechanism comes first, we have the kinematic relationships:

$$x_s \; = \; Rs\psi$$
$$y_s \; = \; Rc\theta c\psi$$
$$z_s \; = \; -Rs\theta c\psi$$

and therefore the unit vector is simply:

$$\hat{u} \; = \; \left[s\psi \; c\theta c\psi \; -s\theta c\psi\right]$$

In this case, the pixel coordinates are angles, and not tangents so that:

$$\psi \; = \; \frac{(col - cols/2)}{cols}HFOV$$

$$\theta \; = \; \frac{(row - rows/2)}{rows}VFOV$$

8. Actuator Kinematics

For wheeled vehicles, the transformation from the angles of the steerable wheels and their velocities onto path curvatures, or equivalently angular velocities, can be very complicated. One reason for this is that there can be more degrees of freedom of steer than are necessary. In this case, the equations which relate curvature to steer angle are ***overdetermined***. In one particular case, however, the steering mechanism is designed such that this will not be the case. This mechanism is used on most conventional automobiles and is called **Ackerman steering**.

8.1 The Bicycle Model

It is useful to approximate the kinematics of the Ackerman steering mechanism by assuming that the two front wheels turn slightly differentially so that the instantaneous center of rotation of the vehicle can be determined purely by kinematic means. This amounts to assuming that the steering mechanism is the same as that of a bicycle. Let the angular velocity vector directed along the body z axis be called $\dot{\beta}$.

Using the bicycle model approximation, the path curvature κ, radius of curvature R, and steer angle α are related by the wheelbase L.

$$\frac{1}{R} = \frac{t\alpha}{L} = \frac{d\beta}{ds}$$
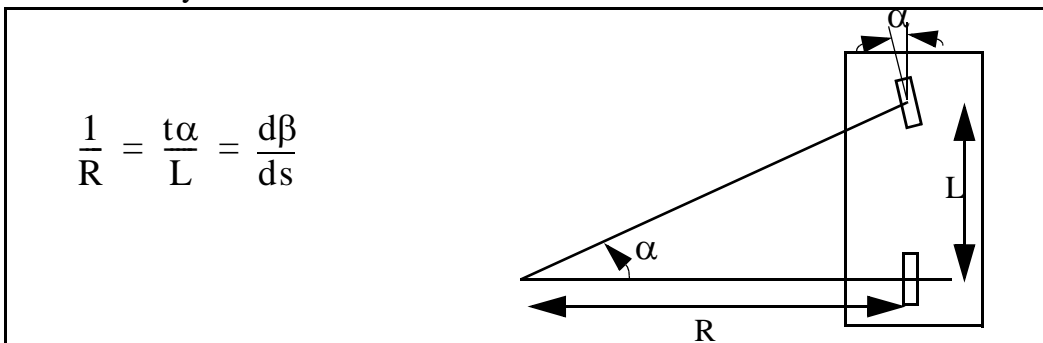
Where $t\alpha$ denotes the tangent of α. Rotation rate is obtained from the speed V as:

$$\dot{\beta} = \frac{d\beta}{ds}\frac{ds}{dt} = \kappa V = \frac{Vt\alpha}{L}$$

Thus, the steer angle α is an indirect measurement of the ratio of $\dot{\beta}$ to velocity through the function:

$$\alpha = atan\left(\frac{L\dot{\beta}}{V}\right) = atan(\kappa L)$$

Although it is common to think of these equations in kinematic terms, this is only possible when the dependence on time is avoided. In fact, this steering mechanism is modelled by a very complicated nonlinear differential equation thus:

$$\frac{d\beta(t)}{dt} = \frac{1}{L}tan[\alpha(t)]\frac{ds}{dt} = \kappa(t)\frac{ds}{dt}$$

9. References

[1] O. Amidi, "Integrated Mobile Robot Control", Robotics Institute Technical Report CMU-RI-TR-90-17.

[2] J. Craig, "Introduction to Robotics, Mechanics and Control", Addison Wesley, 1986

[3] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," J. Applied Mechanics, pp 215-221, June 1955.

[4] J. Farrell, "Integrated Aircraft Navigation", Academic Press, 1976.

[5] D. H. Shin, S. Singh and Wenfan Shi. "A partitioned Control Scheme for Mobile Robot Path Planning", Proceedings IEEE Conference on Systems Engineering, Dayton, Ohio, August 1991

[6] A. J. Kelly, "Modern Inertial and Satellite Navigation Systems", CMU Robotics Institute Technical Report CMU-RI-TR-94-15.

[7] A. J. Kelly, "A Partial Analysis of the High Speed Autonomous Navigation Problem", CMU Robotics Institute Technical Report CMU-RI-TR-94-16.

[8] A. J. Kelly, "A Feedforward Control Approach to the Local Navigation Problem for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-17.

[9] A. J. Kelly, "Adaptive Perception for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-18.

[10] A. J. Kelly, "A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-19.

[11] A. J. Kelly, "An Intelligent Predictive Controller for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-20.

[12] A. J. Kelly, "Concept Design of A Scanning Laser Rangefinder for Autonomous Vehicles", CMU Robotics Institute Technical Report CMU-RI-TR-94-21.

[13] D. Pieper and B. Roth, "The Kinematics of Manipulators Under Computer Control", Ph.D. Thesis, Stanford University, 1968

Index