

WORKING PAPER 69

KINEMATICS OF THE MIT-AI-VICARM MANIPULATOR

by

Berthold K. P. Horn and Hirochika Inoue*

Massachusetts Institute of Technology

Artificial Intelligence Laboratory

May 1974

Abstract

This paper describes the basic geometry of the electric manipulator designed for the Artificial Intelligence Laboratory by Victor Scheinman while on leave from Stanford University. The procedure for finding a set of joint angles that will place the terminal device in a desired position and orientation is developed in detail. This is one of the basic primitives that an arm controller should have. The orientation is specified in terms of Euler-angles. Typically eight sets of joint angles will produce the same terminal device position and orientation.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-70-A-0362-0005.

Working Papers are informal papers intended for internal use.

* on leave from Electrotechnical Laboratory, Tokyo

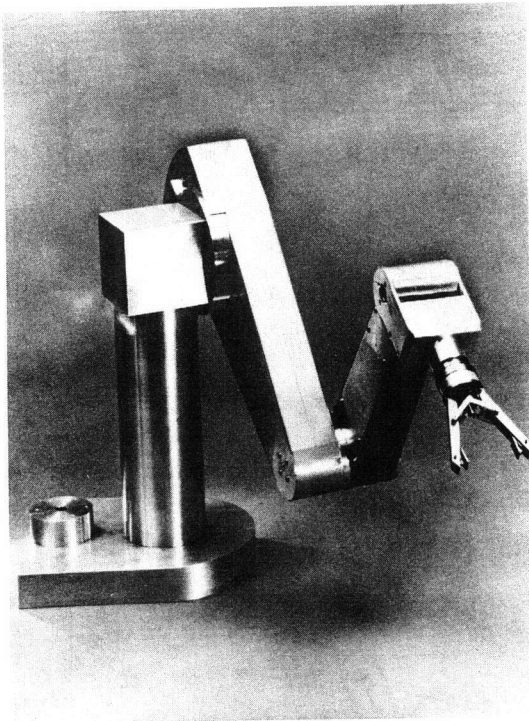


Fig 1

Fig 2

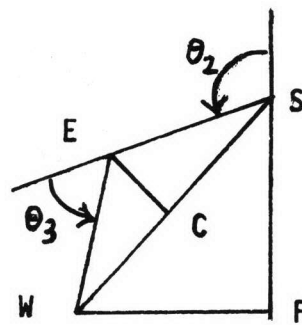
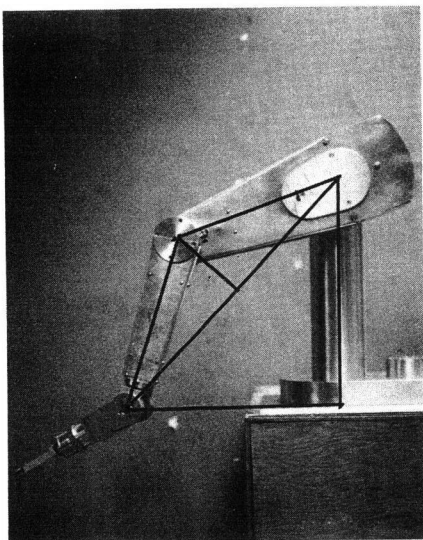
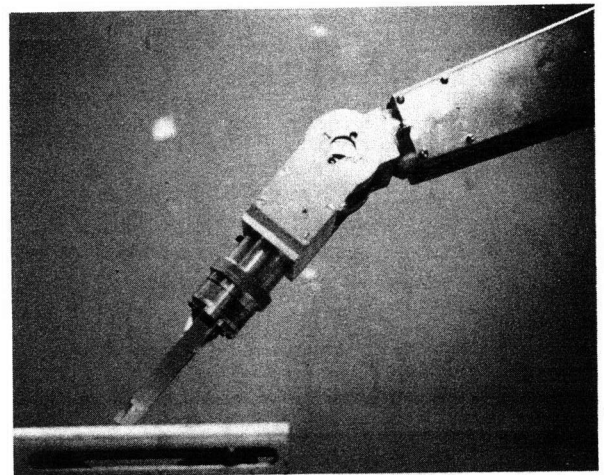


Fig 3

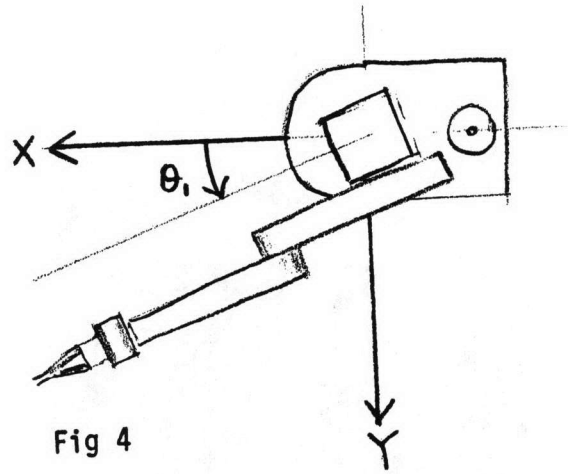
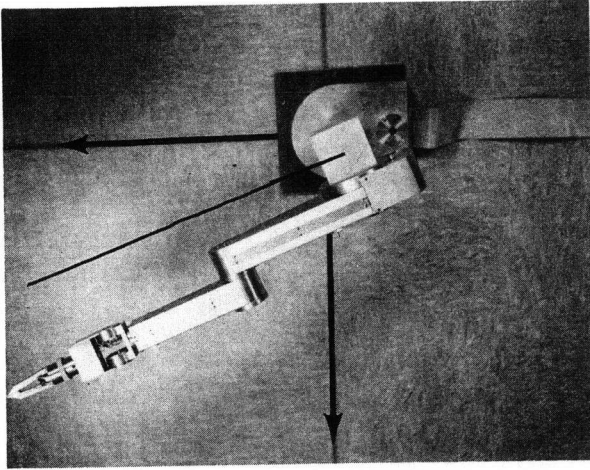


Fig 4

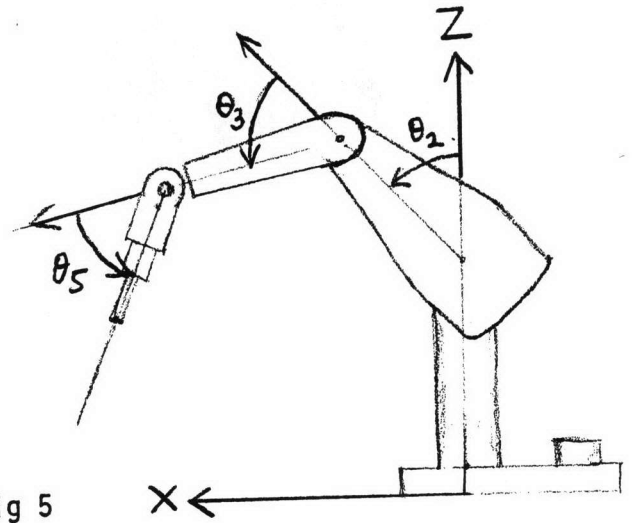
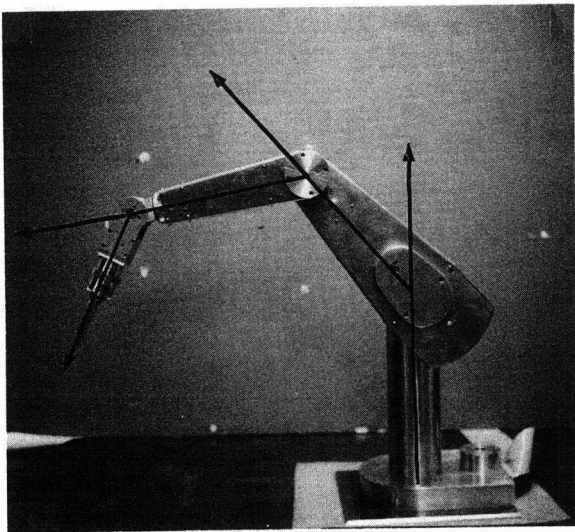


Fig 5

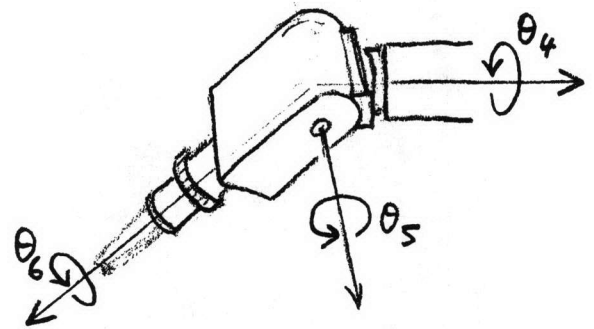
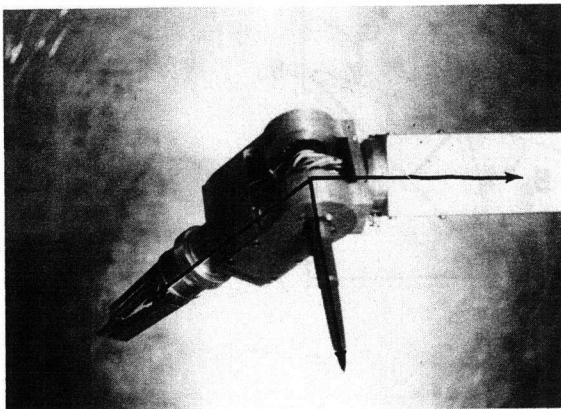


Fig 6

TABLE OF CONTENTS:

1.1	History and Introduction.	4
1.2	Geometry and Operation.	5
2.1	Euler-angles as a means of specifying orientation.	6
2.2	Determining Euler-angles from a rotation matrix.	8
2.3	Other conventions -- Roll, Pitch and Yaw.	10
3.1	Coordinate transformation from base to hand.	12
3.2	Coordinate transformation from hand to base.	16
3.3	Products of the rotation matrices.	16
3.4	Questions of computational efficiency.	18
3.5	The Eight Ways.	19
4.1	Joint angles determined from hand position and orientation.	20
4.2	Finding $\theta_1, \theta_2, \theta_3$.	21
4.3	Finding $\theta_4, \theta_5, \theta_6$.	24
5.1	Various functions defined in the program.	24
5.2	Assorted auxiliary functions.	25
5.3	A LISP program for determining joint angles.	26
5.4	Conventions and other considerations for the PDP11.	28
5.5	Mathematical function subroutines for the PDP11.	29
6.1	Bibliography.	30

APOLOGY: The coordinate systems chosen for the links in this paper allow the use of extremely simple transformation matrices - they are not however the ones that have come into common use when discussing generalized manipulators. It is hoped that this will not prove too much of an inconvenience. It would not be too hard to translate the solution here presented into the corresponding one using the other notation.

1.1 HISTORY AND INTRODUCTION:

A manipulator was designed for the M.I.T. A.I. Laboratory by Victor Scheinman while on leave from Stanford University. He subsequently started manufacturing this manipulator and we are now in possession of a working model with locally designed electronics, interfaced to the Micro-Automation Group's PDP11/40. Two more manipulators are slated for delivery sometime this summer. These will include minor improvements in areas that the prototype was found to be less than optimal. The intent is to allow for two-armed manipulations needed in assembly of small mechanical systems, while retaining one spare arm.

The electronics were designed by Russell Noftsker to provide current servoing of the electric torque motors and back e.m.f sensing of these same motors. Note that the motor torque -- ignoring friction -- is proportional to the current and that the back e.m.f -- if the resistive drop due to the motor current is correctly subtracted out -- is proportional to motor speed. Frederick Drenckhahn developed the layout for the electronics and with William Freeman and Thomas Callahan was responsible for providing the other mechanical and electrical details such as drivers for the brakes in the manipulator. The arm has been disassembled a number of times to make minor repairs, mostly to the wiring. It is not yet Fail-Safe! Please take care!

There is a simple manual controller which allows velocity control of one joint at a time. The inactive joints are locked. There are a number of safety features that prevent the more obvious problems such as gross motions while powering up and down. The current sensing resistor for example also acts as a thermal model of the motor armature and excessive temperature at this point will shut down the electronics in a graceful way.

The authors of this paper developed the transformation equations. Meyer Billmers and Richard Waters are writing a first-order servo program. And of course Richard Waters is interested in implementing a sophisticated manipulator control system.

1.2 GEOMETRY AND OPERATION:

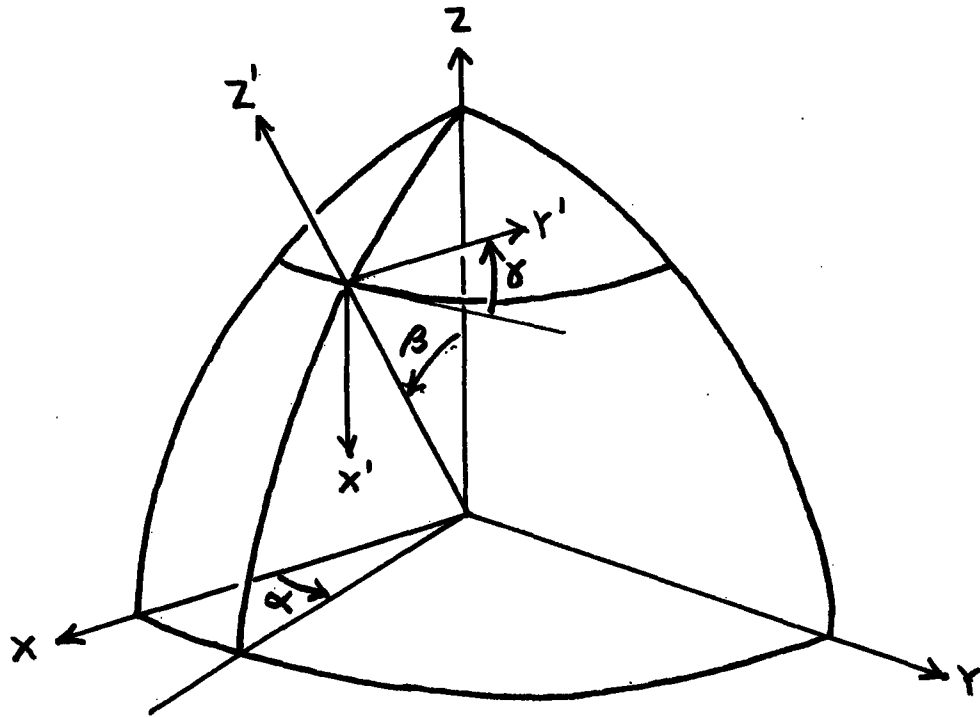
It takes three degrees of freedom to position the terminal device in an arbitrary position in space. It takes another three to give it an arbitrary orientation. A manipulator that wants to achieve arbitrary position and orientation (within some space) ought to have six degrees of freedom. With less, most position orientation combinations will be unattainable. With more, most position orientation combinations can be reached with an infinite variety of joint angle combinations. A manipulator with six degrees of freedom has a finite number of ways of reaching a given position and orientation. In our case this number is eight, as we will see.

The MIT-AI-VICARM manipulator has six rotational joints and no extensional joints. Each joint is equipped with a D.C. torque motor, potentiometers for measuring position, a brake for locking the joint when there is no motion. Eventually some joints will also be equipped with tachometers for a more precise determination of velocity. The arm is about as small as one can hope to make it using the technology of D.C. torque motors and spur gears. The shoulder to elbow link and the elbow to wrist link are each .203 m (8") long. The present terminal device is a rudimentary two-fingered grasper -- the wrist to fingertip distance is .159 m (6.25"). The total offset from shoulder to wrist is .118 m (4.65"), and the shoulder stands .273 m (10.75") above the base.

There is no closed loop analog positional servo. There are a number of reasons for this. The first is that many of the manipulations of interest involve objects that are externally mechanically constrained (nut on bolt, peg in hole, crank on pin, door on hinge). Something akin to force-control is more appropriate in such situations. Another major reason is that because of the non-linear interaction of the moments and torques only slow motions are possible with individual positional control. In a digital model of the arm dynamics such things as Coriolis forces can be dealt with properly.

2.1 SPECIFYING ORIENTATION: (Euler-angles \rightarrow Rotation matrix)

The relative rotation of one coordinate system with respect to another can be given in terms of three angles usually referred to as Euler-angles. Unfortunately there is no standard convention for choosing these angles.



We have chosen α, β, γ as defined in the above diagram. To go from the x', y', z' coordinate system to the x, y, z coordinate system one performs a rotation of $-\gamma$ about the z -axis, a rotation of $-\beta$ about the y -axis and a rotation of $-\alpha$ about the x -axis. Remember that rotations in three-space do not commute. An alternate way of specifying a rotation is by means of an ortho-normal 3 by 3 matrix. Such a matrix would produce a vector in the x, y, z coordinate system when it is multiplied on the right by a vector in the x', y', z' coordinate system. This matrix is clearly the product of the three simpler rotation matrices corresponding to the individual rotations α, β and γ .

$$\begin{vmatrix} x \\ y \\ z \end{vmatrix} = \begin{vmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{vmatrix} \begin{vmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x' \\ y' \\ z' \end{vmatrix}$$

The matrix product becomes:

$$\begin{vmatrix} \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & -\cos \alpha \cos \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \\ \sin \alpha \cos \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \cos \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \\ -\sin \beta \cos \gamma & \sin \beta \sin \gamma & \cos \beta \end{vmatrix}$$

So given the three angles one can easily produce the uniquely determined rotation matrix above. This requires three sine and cosine evaluations, twelve multiplications and four additions/subtractions.

When $\sin \beta = 0$ the z and z' axes line up and the α and γ rotations take place about the same axis. When $\beta = 0$, the above matrix becomes:

$$\begin{vmatrix} \cos(\alpha + \gamma) & -\sin(\alpha + \gamma) & 0 \\ \sin(\alpha + \gamma) & \cos(\alpha + \gamma) & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

When $\beta = \pi$, we have instead:

$$\begin{vmatrix} -\cos(\alpha - \gamma) & -\sin(\alpha - \gamma) & 0 \\ -\sin(\alpha - \gamma) & \cos(\alpha - \gamma) & 0 \\ 0 & 0 & -1 \end{vmatrix}$$

One can restrict β to the range $[0, \pi]$. Outside this range one obtains no new rotation matrices. The matrix defined by $(\pi + \alpha, -\beta, \pi + \gamma)$ is the same as that specified by (α, β, γ) for example.

2.2 GETTING EULER-ANGLES FROM THE ROTATION MATRIX:

Let the rotation matrix be (m_{ij}) for $i=0,1,2$ and $j=0,1,2$.

Lets restrict β to the range $[0, \pi]$ to avoid ambiguity. Since one has 9 entries in the matrix, one can choose between alternate ways of computing the angles.

$$\cos\beta = m_{22}$$

$$\sin^2\beta = m_{02}^2 + m_{12}^2 = m_{20}^2 + m_{21}^2$$

For numerical stability one might want to use some combination, eg:

$$\beta = \text{atan}\left(+\sqrt{(m_{02}^2 + m_{12}^2 + m_{20}^2 + m_{21}^2)/2}, m_{22} \right)$$

Note that this will produce the correct range for β automatically.

Now if $\sin\beta \neq 0$ one can use:

$$\sin\alpha \sin\beta = m_{12} = m_{20}m_{01} - m_{00}m_{21}$$

$$\cos\alpha \sin\beta = m_{02} = m_{20}m_{11} - m_{10}m_{21}$$

to get α , eg: $\alpha = \text{atan}(m_{12}, m_{02})$

Similarly if $\sin\beta \neq 0$ one can use:

$$\sin\delta \sin\beta = m_{21} = m_{02}m_{10} - m_{00}m_{12}$$

$$-\cos\delta \sin\beta = m_{20} = m_{02}m_{11} - m_{01}m_{12}$$

to get δ , eg: $\delta = \text{atan}(m_{21}, -m_{20})$

When $\beta = 0$ however we have a problem since α and γ are no longer uniquely determined. Note however that:

$$(1 + \cos\beta) \sin(\alpha + \gamma) = m_{10} - m_{01}$$

$$(1 + \cos\beta) \cos(\alpha + \gamma) = m_{00} + m_{11}$$

In this case we get: $\alpha + \gamma = \text{atan}(m_{10} - m_{01}, m_{00} + m_{11})$

Similarly when $\beta = \pi$ we have a similar problem, but observe that:

$$(1 - \cos\beta) \sin(\alpha - \gamma) = -m_{01} - m_{10}$$

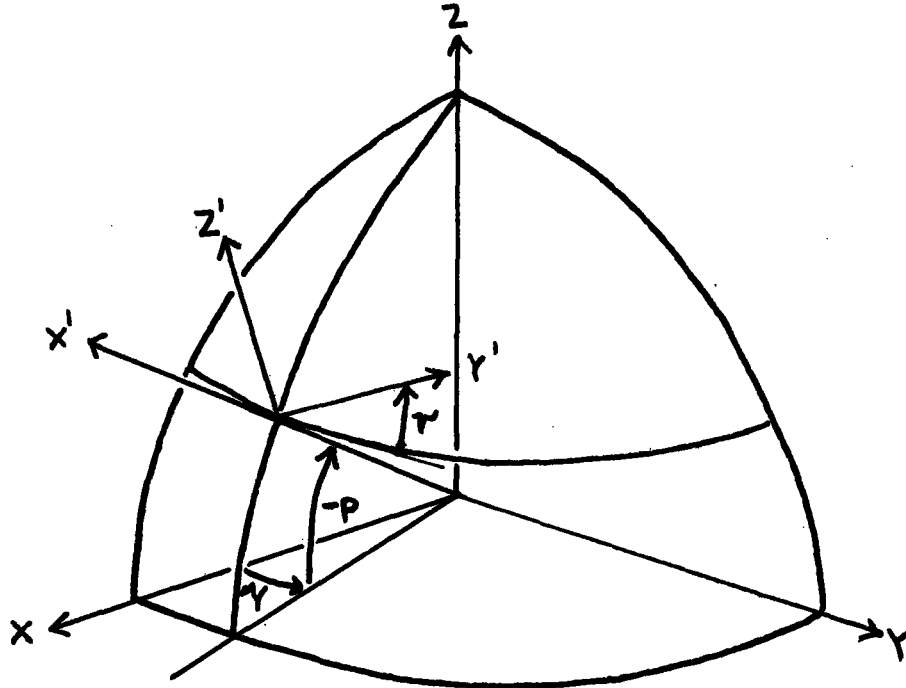
$$(1 - \cos\beta) \cos(\alpha - \gamma) = m_{11} - m_{00}$$

And so we get: $\alpha - \gamma = \text{atan}(-m_{01} - m_{10}, m_{11} - m_{00})$

In the program, the function SPHERES takes α, β, γ and produces the rotation matrix (m_{ij}) . The function ANGLES takes the rotation matrix and produces α, β, γ . In the degenerate cases mentioned above, ANGLES arbitrarily sets $\alpha = 0$. Note that SPHERES is an inverse for ANGLES, but not the other way around.

2.3 ROLL, PITCH AND YAW:

Many other conventions for Euler-angles are in vogue. One is defined as follows:



In this case we have the following sequence of rotations: roll- r about x -axis, pitch- p about y -axis and yaw y about z -axis.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos y & -\sin y & 0 \\ \sin y & \cos y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos p & 0 & \sin p \\ 0 & 1 & 0 \\ -\sin p & 0 & \cos p \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos r & -\sin r \\ 0 & \sin r & \cos r \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

The matrix product becomes:

$$\begin{pmatrix} \cos y \cos p & \cos y \sin p \sin r - \sin y \cos r & \cos y \sin p \cos r + \sin y \sin r \\ \sin y \cos p & \sin y \sin p \sin r + \cos y \cos r & \sin y \sin p \cos r - \cos y \sin r \\ -\sin p & \cos p \sin r & \cos p \cos r \end{pmatrix}$$

To obtain the three angles given the rotation matrix, consider that:

$$\sin p = -m_{20}$$

$$\cos^2 p = m_{00}^2 + m_{10}^2 = m_{21}^2 + m_{22}^2$$

$$\text{So } p = \text{atan}(-m_{20}, \sqrt{(m_{00}^2 + m_{10}^2 + m_{21}^2 + m_{22}^2)/2}) ; -\pi/2 \leq p \leq \pi/2$$

$$\text{If } \cos p \neq 0 \quad \sin y \cos p = m_{10}$$

$$\cos y \cos p = m_{00}$$

$$y = \text{atan}(m_{10}, m_{00})$$

$$\text{If } \cos p \neq 0 \quad \sin r \cos p = m_{21}$$

$$\cos r \cos p = m_{22}$$

$$r = \text{atan}(m_{21}, m_{22})$$

When $p = \pi/2$ one can use instead:

$$(1 + \sin p) \sin (r-y) = m_{01} - m_{20}$$

$$(1 + \sin p) \cos (r-y) = m_{11} + m_{02}$$

$$\text{To get } r-y = \text{atan}(m_{01} - m_{20}, m_{11} + m_{02})$$

When $p = -\pi/2$ one can use:

$$(1 - \sin p) \sin (r+y) = -m_{01} - m_{20}$$

$$(1 - \sin p) \cos (r+y) = m_{11} - m_{02}$$

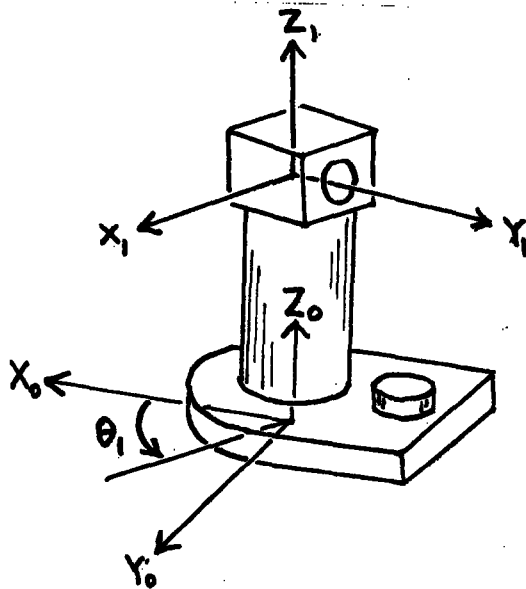
$$\text{To get } r+y = \text{atan}(-m_{01} - m_{20}, m_{11} - m_{02})$$

Note: This formulation of the Euler-angles is more appropriate for the roll-pitch-yaw geometry of the TOS Mini-Six arm designed by Carl Flatau. The previous formulation on the other hand is convenient for the roll-yaw-roll geometry of the wrist in the MIT-AI-VICARM manipulator and will be used from now on in this paper.

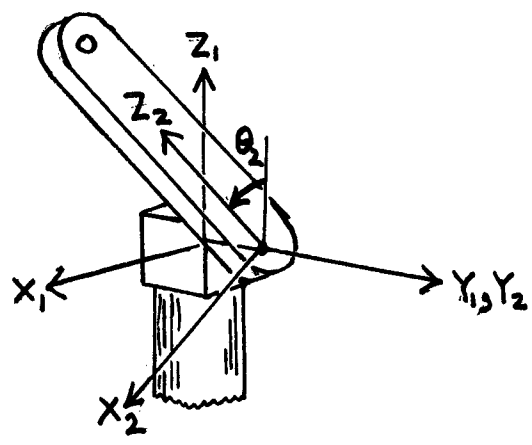
3.1 DETERMINING THE COORDINATE TRANSFORMATION FROM THE BASE TO THE HAND:

To determine where the terminal device is in space and how it is oriented when given the six joint angles, it is convenient to erect a coordinate system in each link of the arm. There are six links (numbered 1, 2 ... 6) and six joints (also numbered 1, 2, ... 6). The base counts as link 0. Each coordinate system is associated with a link and each transformation is associated with a joint. Each transformation can be conveniently written as a 4 by 4 matrix. This allows both a rotation (top left-hand 3 by 3 sub-matrix) and an offset (Top 3 elements of last column). The last row is always constant (0, 0, 0, 1). To obtain the overall transformation later on we will just multiply these six matrices together.

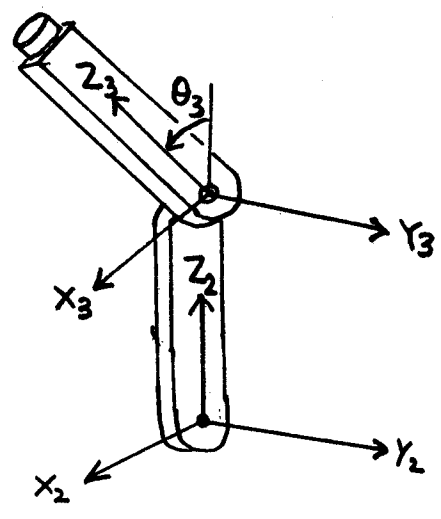
Let	JN	Height of Shoulder above base.	.273 m
	SJ	Horizontal offset of Wrist with respect to Shoulder.	.118 m
	EW	Length of Elbow to Wrist link, also Shoulder to Elbow.	.203 m
	WH	Distance from Wrist to Finger-tips.	.159 m



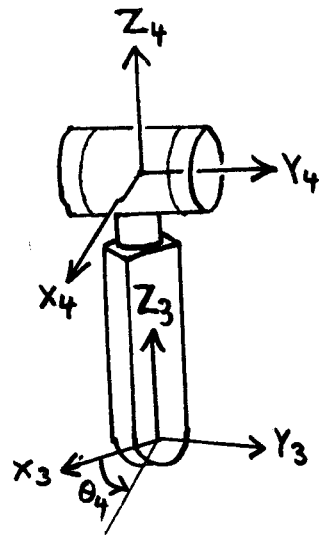
$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & JN \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}$$



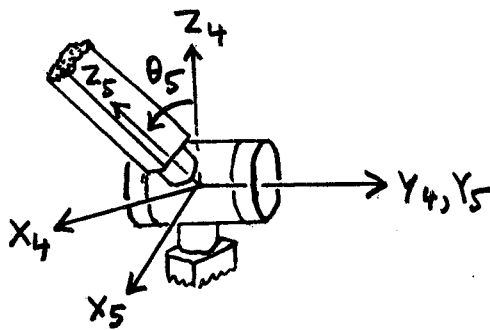
$$\begin{vmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{vmatrix} = \begin{vmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ 0 & 1 & 0 & SJ \\ -\sin \theta_2 & 0 & \cos \theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{vmatrix}$$



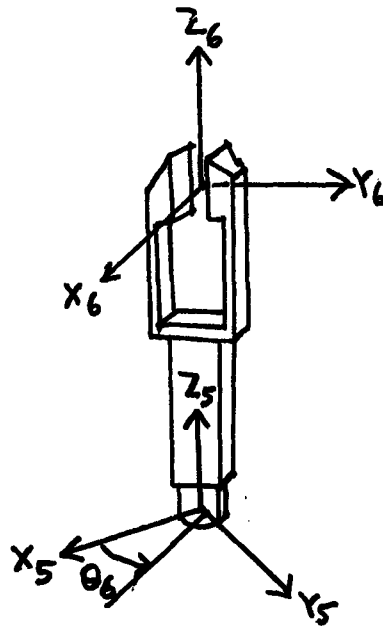
$$\begin{vmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{vmatrix} = \begin{vmatrix} \cos \theta_3 & 0 & \sin \theta_3 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_3 & 0 & \cos \theta_3 & EW \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{vmatrix}$$



$$\begin{vmatrix} x_3 \\ y_3 \\ z_3 \\ 1 \end{vmatrix} = \begin{vmatrix} \cos \theta_4 & -\sin \theta_4 & 0 & 0 \\ \sin \theta_4 & \cos \theta_4 & 0 & 0 \\ 0 & 0 & 1 & \text{EW} \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_4 \\ y_4 \\ z_4 \\ 1 \end{vmatrix}$$



$$\begin{vmatrix} x_4 \\ y_4 \\ z_4 \\ 1 \end{vmatrix} = \begin{vmatrix} \cos \theta_5 & 0 & \sin \theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta_5 & 0 & \cos \theta_5 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x_5 \\ y_5 \\ z_5 \\ 1 \end{vmatrix}$$



$$\begin{bmatrix} x_5 \\ y_5 \\ z_5 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_6 & -\sin \theta_6 & 0 & 0 \\ \sin \theta_6 & \cos \theta_6 & 0 & 0 \\ 0 & 0 & 1 & WH \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_6 \\ y_6 \\ z_6 \\ 1 \end{bmatrix}$$

To summarize we have, when going from the terminal device to the base:

Offset in z	WH
Rotation about z	$-\theta_6$
Rotation about y	$-\theta_5$
Rotation about z	$-\theta_4$
Offset in z	EW
Rotation about y	$-\theta_3$
Offset in z	EW
Rotation about y	$-\theta_2$
Offset in y	$\$J$
Rotation about z	$-\theta_1$
Offset in z	JN

The offsets are all fixed, the rotations are all controlled.

3.2 COORDINATE TRANSFORMATION FROM HAND TO BASE:

We can multiply the six matrices obtained in the previous section to obtain one overall transformation. Let this product matrix be (m_{ij}) for $i=0,1,2,3$ and $j=0,1,2,3$. As pointed out before the rotation of the terminal device relative to the base coordinate system is given by the top left-hand sub-matrix (m_{ij}) $i=0,1,2$ and $j=0,1,2$ and the position of the terminal device is (m_{03}, m_{13}, m_{23}) . To determine the orientation of the hand in terms of Euler-angles we can operate on the matrix as indicated before. Note that each of the terms of the rotation sub-matrix is a polynomial in the sines and cosines of the six joint angles. Each product term in such a polynomial will contain at most one sine or cosine of a particular angle. The offset vector is similar except that the offsets also appear multiplied with the sines and cosines.

3.3 THE TRANSFORMATION MATRICES:

Lets start with the first three going inward from the hand. Abbreviate $\cos \theta_6$ by c_6 etc. then:

$$\begin{vmatrix} c_4 & -s_4 & 0 & 0 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & EW \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} c_5 & 0 & s_5 & 0 \\ 0 & 1 & 0 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & WH \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

This product becomes:

$$\begin{vmatrix} (c_4 c_5 c_6 - s_4 s_6) & (-c_4 c_5 s_6 - s_4 c_6) & c_4 s_5 & c_4 s_5 WH \\ (s_4 c_5 c_6 + c_4 s_6) & (-s_4 c_5 s_6 + c_4 c_6) & s_4 s_5 & s_4 s_5 WH \\ -s_5 c_6 & s_5 s_6 & c_5 & (c_5 WH+EW) \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

That is, just the Euler-angle rotation matrix defined previously, augmented on the right by an offset vector easily calculated from the third column and the known dimensions WH and EW. This takes 15 multiplications. (The Euler-angles were defined to coincide with the last three rotational joints on the arm)

Now let's look at the last three going inward from the hand. Again using the abbreviation c_3 for $\cos \theta_3$ etc. we have:

$$\begin{vmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & JN \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} c_2 & 0 & s_2 & 0 \\ 0 & 1 & 0 & SJ \\ -s_2 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} c_3 & 0 & s_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & EW \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

This product becomes:

$$\begin{vmatrix} c_1 c_{23} & -s_1 & c_1 s_{23} & (c_1 c_2 EW - s_1 SJ) \\ s_1 c_{23} & c_1 & s_1 s_{23} & (s_1 s_2 EW + c_1 SJ) \\ -s_{23} & 0 & c_{23} & (c_2 EW + JN) \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

where $c_{23} = \cos(\theta_2 + \theta_3) = c_2 c_3 - s_2 s_3$

$s_{23} = \sin(\theta_2 + \theta_3) = s_2 c_3 + c_2 s_3$

It takes 14 multiplications to develop this matrix.

Further multiplication of the two matrices developed here produces an overall transformation matrix that contains very complicated terms. There is no point in presenting the result here since it is not used.

3.4 COMPUTING THE TRANSFORMATION MATRIX:

Obviously we could just develop the six 4 by 4 matrices and multiply them together. This requires $(6-1)*4*4*4 = \underline{320}$ multiplications.

Since the last row in each matrix is a constant (0, 0, 0, 1) we can carry it along implicitly. This reduces the arithmetic to $(6-1)*3*4*3 = \underline{180}$ multiplications.

Next one may note that the rotation components of the matrices only affect two coordinates at a time (since we defined the coordinate systems in the links to always line up the rotation axis with one of the coordinate axes). This allows us to replace general matrix multiplication by mere rotation operators (as in the functions ROTY and ROTZ). This reduces the arithmetic further to $(6-1)*4*2*2 = \underline{80}$ multiplications.

Moral: homogeneous coordinates and general schemes are useful for conceptual development, but computationally ineffecient.

It is still possible to reduce the work further by noting that the three wrist rotations correspond to the three Euler-angles we have defined and that as noted we can calculate the product of the first three matrices using only 15 multiplications. Then we end up with $15+3*4*2*2 = \underline{63}$.

It might be thought that separately calculating the product of the first three and the last three matrices as indicated in the last section and then determining their product might save even more - but one gets: $15 + 14 + 3*4*3 = \underline{65}$ multiplications.

Even worse is the situation obtained by multiplying out all six symbolically. Each term in the final matrix requires about 30 multiplications, for a grand total of 287.

The function COORDS, using the fastest of the above methods produces the transformation matrix when given the six joint angles. The functions COORD does the same, but also calculates the position and orientation of the terminal device. Note that the matrix is uniquely determined by the joint angles. The converse is not true. A given terminal device orientation and position can be achieved by up to eight joint angle configurations.

3.5 THE EIGHTFOLD WAY:

Most orientations and positions of the hand can be reached by any one of eight sets of joint angles. These correspond to three two-way choices. One simple way of dividing up the possibilities is according to the signs of the three angles θ_2 , θ_3 and θ_5 . So one could for example restrict each of these to be positive to obtain a unique solution. This has implications in terms of the potentiometers built into the arm, since they could be restricted in their travel and thus provide greater accuracy and resolution.

A somewhat more natural division of the redundant solution space is provided by the following:

Sign of $\theta_2 + \theta_3/2$: When this is positive, the wrist is in front of the "chest". The "chest" is the side of the metal cube at the shoulder that is $\pi/2$ clockwise from the arm attachment when viewed from above. (That is, the arm is thought of as a left arm). If the sign of $\theta_2 + \theta_3/2$ is negative on the other hand, the wrist will be "behind". ~~The same set of~~ positions can be reached in this fashion, provided θ_1 is changed by almost π . The other angles will be affected as well to some degree.

Sign of θ_3 : If $\theta_2 + \theta_3/2$ is positive then the sign of θ_3 determines whether the elbow is above or below the line connecting the wrist to the shoulder. The same set of positions can be reached in this fashion provided θ_2 is adjusted by $2\theta_3$. The other angles will be affected as well to some degree. If the sign of $\theta_2 + \theta_3/2$ is negative, the two positions are swapped around.

Sign of θ_5 : Since the last three joints in the arm correspond to the Euler angles we defined, there is a redundancy in that the position $\theta_4, \theta_5, \theta_6$ is equivalent to the position $\pi + \theta_4, -\theta_5, \pi + \theta_6$. Even the space occupied by the arm is the same ~~for these two~~ sets of joint angles. The other three angles are not affected.

4.1 OBTAINING JOINT ANGLES FROM HAND POSITION AND ORIENTATION:

The hand position and orientation allows one to calculate the transformation matrix from the hand coordinate system to the base coordinate system. We now have to find what six joint angles will give rise to this transformation. Since each element in the matrix is a polynomial in the sines and cosines of the angles and the various offsets we could try to solve this set of twelve polynomial equations so formed for the twelve sines and cosines. This can be done by successively eliminating variables by combining equations somewhat analogously to the method used in the solution of linear equation. The difficulty is that the degrees of the polynomials multiply in these operations. When finally only one equation in one unknown remains it will be found to have a degree of a few thousand!

One alternative is interpolation from a stored set of configurations combined with iteration and some hill-climbing method. This implies repeatedly evaluating the transformation matrix given the guessed joint angles and is quite slow. Fortunately the arm under consideration was specifically designed to avoid this problem. The rotational axes in the wrist intersect in one point and allow a decomposition of the problem into two smaller ones. One can find θ_1 , θ_2 , θ_3 independently first and then use these to find θ_4 , θ_5 , θ_6 . Briefly, the desired hand orientation allows one to determine where the wrist should be, and only the first three angles affect the position of the wrist. Having determined these angles one can calculate the orientation of the hand in the elbow-wrist coordinate system. Since the wrist angles correspond to the Euler-angles previously defined it is quite easy then to determine the last three angles.

4.2 FINDING $\theta_1, \theta_2, \theta_3$:

The position of the wrist can be found in terms of the first three joints angles using a matrix previously developed to get us to the elbow-wrist coordinate system and noting that in this system, the wrist is located at (0, 0, EW):

$$W_x = c_1 (s_{23} + s_2)EW - s_1 SJ$$

$$W_y = s_1 (s_{23} + s_2)EW + c_1 SJ$$

$$W_z = (c_{23} + c_2)EW + JN$$

(It would have been nicer if the shoulder-offset SJ was zero !)

Remember that $s_{23} = s_2 c_3 + c_2 s_3$ and that $c_{23} = c_2 c_3 - s_2 s_3$.

We now have to solve this set of equations for $s_1, c_1, s_2, c_2, s_3, c_3$ -- keeping in mind that $s_1^2 + c_1^2 = 1$ and so on. Presumably one could use MATHLAB to do this, but its easy enough by hand if one introduces some intermediate variables as follows:

$$AO^2 = W_x^2 + W_y^2 = (s_{23} + s_2)^2 EW^2 + SJ^2$$

$$BO^2 = AO^2 - SJ^2 = (s_{23} + s_2)^2 EW^2$$

$$BO = \pm \sqrt{BO^2}$$

$$SF = JN - W_z = -(c_{23} + c_2)EW$$

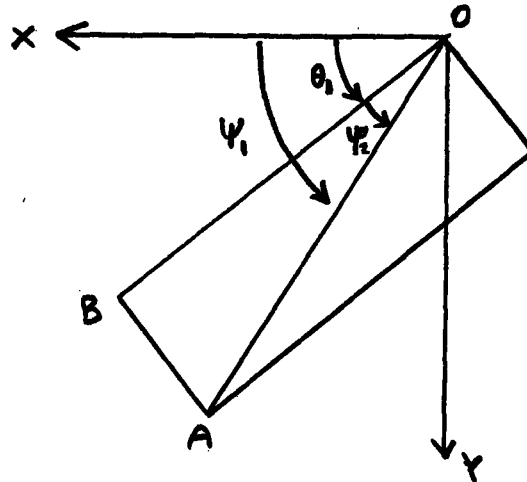
$$SW^2 = BO^2 + SF^2 = 2(1 + c_3)EW^2$$

$$CW^2 = SW^2/4 = (1 + c_3)EW^2/2$$

$$EC^2 = EW^2 - CW^2 = (1 - c_3)EW^2/2$$

$$EC = \pm \sqrt{EC^2}$$

Note that the choice of sign for BO determines the sign of $\theta_2 + \theta_3/2$, while the choice of sign for EC determines the sign of θ_3 . Further if $BO^2 < 0$, the wrist position is too close the vertical column to be reached, while if $EC^2 < 0$, the wrist position is too far out to be reached.



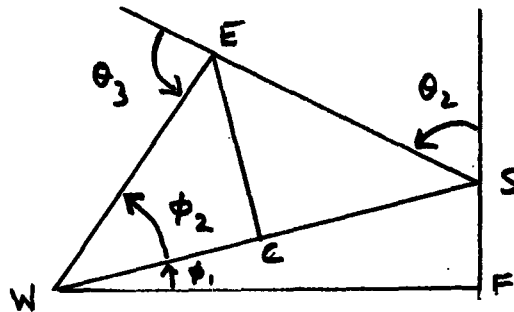
To understand the above formulae it also helps to introduce some intermediate angles. In the above diagram for example we have the point A at (w_x, w_y) and so:

$$\sin \psi_1 = w_y/AO \quad \cos \psi_1 = w_x/AO$$

$$\sin \psi_2 = AB/AO = SJ/AO \quad \cos \psi_2 = BO/AO$$

One then gets sine and cosine of θ_1 by noting that $\theta_1 = \psi_1 - \psi_2$.

Further:



$$\sin \phi_1 = SF/SW \quad \cos \phi_1 = WF/SW = BO/SW$$

$$\sin \phi_2 = EC/EW \quad \cos \phi_2 = CW/EW$$

One then gets sines and cosines of θ_2 and θ_3 by noting that:

$$\theta_2 = \pi/2 + \phi_1 - \phi_2 \quad \theta_3 = 2\phi_2$$

4.4 FINDING $\theta_4, \theta_5, \theta_6$:

Once $\theta_1, \theta_2, \theta_3$ are known, the orientation of the elbow-wrist link can be determined. This implies that it is now possible to determine what the orientation of the hand is in terms of the elbow-wrist coordinate system. Since the last three joints correspond to the Euler-angles in this system one can easily determine their values using methods already demonstrated. The rotation matrix for the hand orientation has to be rotated about the z-axis by θ_1 and about the y-axis by $\theta_2 + \theta_3$.

This is by far the most comprehensible approach. We have various other solutions in terms of vector products, geometric methods and the like, but they are all hard to understand and debug.

5.1 AUXILIARY FUNCTIONS:

OFFY	o_y	adds o_y into matrix as offset along y-axis.
OFFZ	o_z	adds o_z into matrix as offset along z-axis.
ROTY	θ_y	rotates transformation by θ_y about y-axis.
ROTZ	θ_z	rotates transformation by θ_z about z-axis.
SPHERES	α, β, δ	Produces ortho-normal rotation matrix (m_{ij}) $i, j=0,1,2$ given orientation in terms of Euler-angles.
ANGLES		Returns Euler-angles determined from rotation matrix. β will be positive or negative depending on whether argument is +1.0 or -1.0
ATANS		Takes two arguments, returns atan of their ratio. Result in range $-\pi$ to $+\pi$.

5.2 THE FUNCTIONS IN THE PROGRAM:

- COORDS** $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ $(m_{ij}) \ i=0,1,2 \ j=0,1,2,3$
 Produces the unique matrix of transformation from the hand to the base coordinate system.
- JOINTS** $(m_{ij}) \ i=0,1,2 \ j=0,1,2,3$ $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$
 Produces a set of joint-angles from the transformation matrix. BOSGN, ECSGN, PMSGN control which of the eight possibilities is chosen (Each must be set to +1.0 or -1.0).
- COORDSX** $(m_{ij}) \ i=0,1,2 \ j=0,1,2,3$ $H_x, H_y, H_z, \alpha, \beta, \gamma$
 Produces the position of the centre of the finger-tips and the orientation of the terminal-device in terms of Euler-angles.
- JOINTSX** $H_x, H_y, H_z, \alpha, \beta, \gamma$ $(m_{ij}) \ i=0,1,2 \ j=0,1,2,3$
 Produces the transformation matrix from the hand position and orientation given in terms of Euler-angles.
- COORD** $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ $H_x, H_y, H_z, \alpha, \beta, \gamma$
 Composition of COORDS and COORDSX.
- JOINT** $H_x, H_y, H_z, \alpha, \beta, \gamma$ $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$
 Composition of JOINTSX and JOINTS.

```

5.3 (DECLARE (SPECIAL PI PI2 PI4 WH EW SJ JN SJS EWS BOSGN ECSGN PMSGN))
(SETQ PI 3.14159265 PI2 (//$ PI 2.0) PI4 (//$ PI2 2.0))
(SETQ WH .159 SJ .118 JN .273 EW .203 SJS (*$ SJ SJ) EWS (*$ EW EW))
(SETQ BOSGN 1.0 ECSGN 1.0 PMSGN 1.0)
(ARRAY M T 3 4)
(DEFUN ATANS (S C) (-$ (ATAN (MINUS S) (MINUS C)) PI))
(DEFUN OFFY (D) (STORE (M 1 3) (+$ (M 1 3) D)))
(DEFUN OFFZ (D) (STORE (M 2 3) (+$ (M 2 3) D)))
(DEFUN ROTY (TH) (PROG (SINT COST TA TB)
  (SETQ SINT (SIN TH) COST (COS TH))
  (DO J 0 (1+ J) (= J 4)
    (SETQ TA (+$ (*$ COST (M 0 J)) (*$ SINT (M 2 J)))
      TB (-$ (*$ COST (M 2 J)) (*$ SINT (M 0 J))))
    (STORE (M 0 J) TA) (STORE (M 2 J) TB))))
(DEFUN ROTZ (TH) (PROG (SINT COST TA TB)
  (SETQ SINT (SIN TH) COST (COS TH))
  (DO J 0 (1+ J) (= J 4)
    (SETQ TA (-$ (*$ COST (M 0 J)) (*$ SINT (M 1 J)))
      TB (+$ (*$ COST (M 1 J)) (*$ SINT (M 0 J))))
    (STORE (M 0 J) TA) (STORE (M 1 J) TB))))
(DEFUN SPHERES (AL BE GA) (PROG (SA CA SB CB SG CG)
  (SETQ SA (SIN AL) CA (COS AL)
    SB (SIN BE) CB (COS BE)
    SG (SIN GA) CG (COS GA))
  (STORE (M 0 0) (-$ (*$ CA CB CG) (*$ SA SG)))
  (STORE (M 0 1) (MINUS (+$ (*$ CA CB SG) (*$ SA CG))))
  (STORE (M 1 0) (+$ (*$ SA CB CG) (*$ CA SG)))
  (STORE (M 1 1) (+$ (MINUS (*$ SA CB SG)) (*$ CA CG)))
  (STORE (M 0 2) (*$ CA SB))
  (STORE (M 2 0) (MINUS (*$ SB CG)))
  (STORE (M 1 2) (*$ SA SB))
  (STORE (M 2 1) (*$ SB SG))
  (STORE (M 2 2) CB)))
(DEFUN ANGLES (SGN) (PROG (AL BE GA SINB COSB)
  (SETQ SINB (SQRT (//$ (+$ (*$ (M 0 2) (M 0 2)) (*$ (M 1 2) (M 1 2))
    (*$ (M 2 0) (M 2 0)) (*$ (M 2 1) (M 2 1)))
    2.0))
  COSB (M 2 2)
  BE (ATANS (*$ SGN SINB) COSB))
  (COND ((> SINB 1.0E-5)
    (SETQ AL (ATANS (*$ SGN (M 1 2)) (*$ SGN (M 0 2)))
      GA (ATANS (*$ SGN (M 2 1)) (*$ SGN (MINUS (M 2 0)))))
    ((PLUSP COSB) (SETQ AL 0.0
      GA (ATANS (-$ (M 1 0) (M 0 1)) (+$ (M 1 1) (M 0 0))))
    ((MINUSP COSB) (SETQ AL 0.0
      GA (ATANS (+$ (M 1 0) (M 0 1)) (-$ (M 1 1) (M 0 0))))
    (T (RETURN 'MATRIX-BAD)))
  (RETURN (LIST AL BE GA))))

```

```
(DEFUN COORDS (TH1 TH2 TH3 TH4 TH5 TH6)
```

```
(SPHERES TH4 TH5 TH6)
```

```
(STORE (M 0 3) (*$ WH (M 0 2)))
```

```
(STORE (M 1 3) (*$ WH (M 1 2)))
```

```
(STORE (M 2 3) (*$ WH (M 2 2)))
```

```
(OFFZ EW) (ROTY TH3) (OFFZ EW)
```

```
(ROTY TH2) (OFFY SJ) (ROTZ TH1) (OFFZ JN))
```

```
(DEFUN JOINTS NIL (PROG (WX WY WZ BOS BO SF CWS CW ECS EC TH1 TH2 TH3)
```

```
(SETQ WX (-$ (M 0 3) (*$ WH (M 0 2)))
```

```
WY (-$ (M 1 3) (*$ WH (M 1 2)))
```

```
WZ (-$ (M 2 3) (*$ WH (M 2 2)))
```

```
BOS (-$ (+$ (*$ WX WX) (*$ WY WY)) SJS))
```

```
(COND ((MINUSP BOS) (RETURN 'TOO-CLOSE)))
```

```
(SETQ BO (*$ BOSGN (SQRT BOS))
```

```
SF (-$ JN WZ)
```

```
CWS (//$ (+$ BOS (*$ SF SF)) 4.0)
```

```
CW (SQRT CWS)
```

```
ECS (-$ EWS CWS)
```

```
(COND ((MINUSP ECS) (RETURN 'TOO-FAR)))
```

```
(SETQ EC (*$ ECSGN (SQRT ECS))
```

```
TH1 (ATANS (-$ (*$ WY BO) (*$ WX SJ)) (+$ (*$ WX BO) (*$ WY SJ)))
```

```
TH2 (ATANS (+$ (*$ CW BO) (*$ EC SF)) (-$ (*$ EC BO) (*$ CW SF)))
```

```
TH3 (ATANS (*$ 2.0 EC CW) (-$ CWS ECS)))
```

```
(ROTZ (MINUS TH1))
```

```
(ROTY (MINUS (+$ TH2 TH3)))
```

```
(RETURN (APPEND (LIST TH1 TH2 TH3) (ANGLES PMSGN))))
```

```
(DEFUN COORDSX NIL
```

```
(APPEND (LIST (M 0 3) (M 1 3) (M 2 3)) (ANGLES 1.0)))
```

```
(DEFUN JOINTSX (HX HY HZ AL BE GA)
```

```
(SPHERES AL BE GA)
```

```
(STORE (M 0 3) HX) (STORE (M 1 3) HY) (STORE (M 2 3) HZ))
```

```
(DEFUN COORD (TH1 TH2 TH3 TH4 TH5 TH6)
```

```
(COORDS TH1 TH2 TH3 TH4 TH5 TH6) (COORDSX))
```

```
(DEFUN JOINT (HX HY HZ AL BE GA) (JOINTSX HX HY HZ AL BE GA) (JOINTS))
```

5.4 REPRESENTATION OF ANGLES, FRACTIONS AND DISTANCES

For angles, lets choose 2^{16} to correspond to one revolution. This makes for automatic mod(2 pi) operation. Furthermore you can think of the number either as signed and representing angles in (-pi, +pi) or as unsigned numbers, representing angles in (0,2pi). So:

$$\text{number} = (\text{angle}/ \text{pi}) 2^{15}$$

For fractions, lets choose 2^{15} to correspond to unity. This allows a range of -1.0 to almost + 1.0 (Or 0.0 to almost 2.0 if you like to think of it as an unsigned number) and is this just right for output from SIN and COS for example. (It is also consistent with above as a fraction of pi).

$$\text{number} = (\text{fraction}) 2^{15}$$

For distances lets use the metric system and since the arm can reach at most .86 meters (when pointed straight up) lets choose 2^{15} to correspond to one meter. This again makes for consistency.

$$\text{number} = (\text{meters}) 2^{15}$$

5.5 MATHEMATICAL FUNCTION SUBROUTINES FOR THE PDP11 (courtesy of bkph)

SQRT will take double precision argument in R0, R1; return in R0. since the maximum result is $2^{15}-1$, the maximum input is $2^{30}-1$. So the first two bits in R0 better be off. This is mainly used for hacking things of the form $SQRT[a^2+b^2]$ - where a, b are single precision,

SIN/COS will take single argument in R0; return answer in R0,R1. is an angle between $-\pi$ and $+\pi$ expressed as $(\text{angle}/\pi) 2^{15}$. The output is a fraction between -1.0 and +1.0 expressed as $(\text{fraction}) 2^{15}$.

ATAN will take two arguments in R0 and R1; returning an angle in R0. The angle will be in the range $-\pi$ to $+\pi$ and will be $ATAN [R0/ R1]$. The idea of using two arguments is that singularities are avoided at multiples of $\pi/2$ and in addition the ambiguity in quadrant can be resolved. The angle produced will be expressed as $(\text{angle}/\pi) 2^{15}$, while the inputs can be in any units since one works with their ratio.

MULTIPLICATION: Multiplication of integers is handled in the obvious way by picking off the low order register. When multiplying by fractions expressed as $(\text{fraction}) 2^{15}$, care has to be taken to get the result with the right scale. This is done by shifting the two words left one and then choosing the high-order word.

BIBLIOGRAPHY:

- Beckett, Jon Terry (1967) "A COMPUTER-AIDED CONTROL TECHNIQUE FOR A REMOTE MANIPULATOR" Digital Systems Laboratory, Engineering Division, Case Institute of Technology.
- Ejiri, M. et al (1971) "AN INTELLIGENT ROBOT WITH COGNITION AND DECISION-MAKING ABILITY" Second International Joint Conference on Artificial Intelligence, Proceedings of.
- Ernst, Heinrich Arnold (1961) "MH-1, A COMPUTER-OPERATED MECHANICAL HAND" D.Sc. Thesis, Department of Electrical Engineering, M.I.T.
- Flatau, Carl R. (1969) "COMPACT MASTER-SLAVE MANIPULATOR WITH OPTIMIZED COMMUNICATION LINKS" Proceedings of 17th Conference on Remote Systems Technology.
- Flatau, Carl R. et al (1972) "MA-22-A COMPACT BILATERAL SERVO MASTER-SLAVE MANIPULATOR" Proceedings of the 20th Conference on Remote Systems Technology.
- Flatau, Carl R. (1973) "DESIGN OUTLINE FOR MINI-ARMS BASED ON MANIPULATOR TECHNOLOGY" A.I. Memo 300.
- Gavrilovic Momcilo M. (1973) "OPTIMAL PATH PLANNING FOR MANIPULATION AND LEGGED LOCOMOTION SYSTEMS" Technical Report, Laboratory for Automatic Control, Mihailo Pupin Institute, Beograd, Yugoslavia.
- Gresser, Jean-Yves (1968) "DESCRIPTION AND CONTROL OF MANIPULATION BY COMPUTER-CONTROLLED ARM" A.I. Memo 165.
- Hill, J.W. and Sword, A.J. (1972) "STUDIES TO DESIGN AND DEVELOP IMPROVED REMOTE MANIPULATOR SYSTEMS" Stanford Research Institute.
- Horn, Berthold K. P. (1972) "VISEM: A BAG OF "ROBOTICS" FORMULAE" Working Paper 34.

- Ignat'yev, M.B. et al (1972) "ROBOT-MANIPULATOR CONTROL ALGORITHMS"
Leningrad, Translated and published by Joint Publications Research
Service 59717.
- Inoue, H. (1971) "COMPUTER CONTROLLED BILATERAL MANIPULATORS" Bulletin
of the Japanese Society of Mechanical Engineers, Vol. 14, No. 69.
- Kahn, M.E. (1969) "THE NEAR-MINIMUM-TIME CONTROL OF OPEN-LOOP ARTICULATED
KINEMATIC CHAINS" Stanford AIM 106.
- Markiewicz, B.R. (1973) "ANALYSIS OF THE COMPUTED TORQUE DRIVE METHOD
AND COMPARISON WITH CONVENTIONAL POSITION SERVO FOR A COMPUTER
CONTROLLED MANIPULATOR" TM 33-601 Jet Propulsion Laboratory.
- Minsky, Marvin (1972) "MANIPULATOR DESIGN VIGNETTES" A.I. Memo 267.
- Nevins, J.L. et al (1973) "REPORT ON ADVANCED AUTOMATION" R-764, The
Charles Stark Draper Laboratory.
- Nevins, J.L. et al (1974) "EXPLORATORY RESEARCH IN INDUSTRIAL MODULAR
ASSEMBLY" R-800, The Charles Stark Draper Laboratory,
- Paul, Richard (1972) "MODELLING, TRAJECTORY CALCULATION AND SERVOING OF
A COMPUTER CONTROLLED ARM" Stanford AIM 177.
- Pieper, Donald Lee (1968) "THE KINEMATICS OF MANIPULATORS UNDER COMPUTER
CONTROL" Stanford AIM 72.
- Scheinman, Victor David (1969) "DESIGN OF A COMPUTER CONTROLLED MANIPU-
LATOR" Stanford AIM 92.
- Silver, David (1973) "THE LITTLE ROBOT SYSTEM" A.I. Memo 273.
- Uicker, J.J. Jr. (1967) "DYNAMIC FORCE ANALYSIS OF SPACIAL LINKAGES"
Transactions ASME.

Waters, Richard (1973) "MECHANICAL ARM CONTROL" Working Paper 42.

Waters, Richard (1974) "A MECHANICAL ARM CONTROL SYSTEM" A.I. Memo 301.

Whitney, D.E. (1968) "STATE SPACE MODELS OF REMOTE MANIPULATION TASKS"
Ph.D. Thesis, Department of Mechanical Engineering, M.I.T.

Whitney, Daniel E. (1971) "THE MATHEMATICS OF COORDINATED CONTROL OF
PROSTHESES AND MANIPULATORS" Department of Mechanical Engineering,
M.I.T.