Theoretical Description of Robotic Mechanisms

Kinematics of Common Industrial Robots

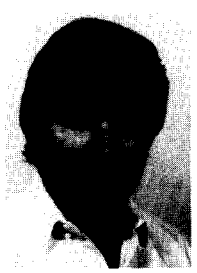
John Lloyd and Vincent Hayward
Computer Vision and Robotics Laboratory, McGill Research Centre for Intelligent Machines, McGill University, Montréal, Québec, Canada

An approach to finding the solution equations for simple manipulators is described which enhances the well known method of Paul, Renaud, and Stevenson, by explicitly making use of known decouplings in the manipulator kinematics. This reduces the set of acceptable equations from which we obtain relationships for the joint variables. For analyzing the Jacobian, such decoupling is also useful since it manifests itself as a block of zeros, which makes inversion much easier. This zero lock can be used to obtain a concise representation for the forward and inverse Jacobian computations. The decoupling also simplifies the calculations sufficiently to allow us to make good use of a symbolic algebra program (MACSYMA) in obtaining our results. Techniques for using MACSYMA in this way are described. Examples are given for several industrial manipulators.

Keywords: Robots, Kinematics, Jacobian matrices, Automated mathematical derivations.

1. Introduction

There are two ways to approach the problem of inverse robot kinematics: numerically and symbolically. Numerical treatments [1] are general, and can be made to yield clean solutions in the presence of singularities, but are computationally burdensome for real time applications. By contrast, analytical solutions, optimized for a particular robot, can be quite rapid. The major drawback is that working out such a solution may not always be possible (although it usually is for most industrial manipulators). In the general case of a robot with 6 revolute joints, it has been shown that the solution for the tangent of the half angle of each joint is a 32 degree polynomial of the coordinates of the last link [12]. Attempts to derive an explicit form for this polynomial have been unsuccessful, although it has been shown to be equivalent to both a 16×16 determinant equation



John Lloyd was born in Victoria, Canada, on April 22, 1958. He received a B.Sc. in physics from McGill University in 1980, worked briefly in the wilds of northern British Columbia, and returned to McGill and received an M.Eng specializing in robotics in 1985. He then worked for a year designing controllers and building laser vision systems for welding robots. During the last two years, Mr. Lloyd has been involved in setting up robot control environments for projects at RCA/General Electric, NASA, and the Jet Propulsion Laboratory. His most recent work has been (with Mike Parker) the development of RCI (Real-time Control Interface), a package for creating real-time control tasks in a multi-CPU VAX/UNIX environment. His research interests include robot kinematics and control, computer languages and real-time computing, and the software engineering aspects of sensor-driven robot and multi-robot systems. He currently lives in Montreal and is pursuing a Ph.D. at the McGill University Research Center for Intelligent Machines.

Vincent Hayward was born in Paris, France, on January 5, 1955. He received the Diplôme d'Ingénieur from Ecole Nationale Supérieure de Mécanique de Nantes, Nantes, France, and the Diplôme de Docteur-Ingénieur from Université de Paris XI at Orsay in computer science, in 1978 and 1981, respectively.
From December 1981 to December 1983, he was at Purdue University, in the Department of Electrical Engineering, the first year as a Visiting Scholar sponsored by CNRS's ARA program (Automatique et Robotique Avancée), and the second year as a Visiting Assistant Professor. There, he developed RCCL, a robot control and programming system. He then joined CNRS at the LIMSI laboratory in Orsay, where he worked as attaché de Recherche on trajectory planning and spatial reasoning until May 1985. He is now Assistant Professor with the Department of Electrical Engineering at McGill University, where he teaches a course on Artificial Intelligence, and a Research Associate with the McGill Research Center for Intelligent Machines. His research interests and publications are in the following areas: robot programming and control, 3-D imaging, computational geometry, spatial reasoning, computational architectures, space and remote applications of robotics and telerobotics.
Dr. Hayward is member of IEEE.

North-Holland
Robotics 4 (1988) 169–191

[2], and a system of eight second degree equations [13].

Some of the original work on obtaining inverse kinematic solutions was done by Pieper, who enumerated special cases in which a closed form solution is feasible. This includes the case where any three adjacent joint axes intersect, which is often true of the last three joints (or "wrist") of many industrial robots and defines the class of "wrist partitioned" manipulators. The kinematics of these robots have been extensively studied [3,4,9].

The best known method for determining a closed kinematic solution is the one described in [7,8] which involves systematically exploring various kinematic relationships and picking out the best ones to yield solutions. In this paper, we adapt this technique so that it is more directed by the special geometry of the manipulator in question. This limits the search for a workable set of equations and is justifiable since special geometry is required to solve a manipulator anyway. We will also show how to use the same in determining the manipulator Jacobian and its inverse. Two special geometries that will be considered in particular are the cases where either three adjacent joints axes intersect, or three adjacent axes are parallel. The approach also lends itself to the use of a computer algebra program (in particular, MACSYMA *), and we will describe how we used this effectively in obtaining our results.

2. Forward and Inverse Kinematics

2.1. Finding the Equations

The forward kinematics for a manipulator are always straight-forward to derive:

$$\mathbf{T}_6 = \mathbf{A}_1\mathbf{A}_2 \ldots \mathbf{A}_N \tag{1}$$

where $\mathbf{T}_6$ is the position of link 6 relative to the robot base, $\mathbf{A}_i$ is the well known "A" matrix for link i [6,7], and N is the number of links.

A method for finding the inverse kinematics is described in [8], where (for a six link arm) we search the following set of equivalent equations for simple relationships between the joint angles we wish to find, and the hand coordinates and the joint angles we have already solved for:

$$\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \mathbf{T}_6 \tag{2}$$

$$\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \mathbf{A}_1^{-}\mathbf{T}_6$$

$$\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \mathbf{A}_2^{-1}\mathbf{A}_1^{-1}\mathbf{T}_6$$

$$\vdots$$

We present here a variation on this approach.

Assume that there are j adjacent joints whose kinematics decouple from those of the $N-j$ surrounding joints. If the products of the transformation matrices of these joints is denoted by $\mathbf{C}$, and the matrices for the surrounding joints are given by $\mathbf{U}_a$ and $\mathbf{U}_b$, then we have

$$\mathbf{U}_a\mathbf{C}\mathbf{U}_b = \mathbf{T}_6 \tag{3}$$

If we can use the decoupling to find an appropriate set of equations for the other $N-j$ joints, then this equation can be rewritten as as

$$\mathbf{C} = \mathbf{U}_a^{-1}\mathbf{T}_6\mathbf{U}_b^{-1} \tag{4}$$

where the right hand side is known. Then, if necessary, we can resolve $\mathbf{C}$ into its component matrices and apply the procedure of (2) to obtain the equations necessary to solve for the remaining j angles which comprise $\mathbf{C}$.

This approach is useful because it breaks the problem cleanly into two smaller problems which are easier to solve, even if it still becomes necessary at that point to resort to a numerical technique to find the remainder of the solution. It is highly applicable to commercial robots which almost always (intentionally) decouple in some straightforward way. Conversely, the idea may be used in reverse to provide guidelines for robot design.

If $\mathbf{C}$ decouples totally, then for an arbitrary adjacent transform $\mathbf{M}$, either $\mathbf{MC}$ or $\mathbf{CM}$ contains $N-j$ independent components which do not depend on $\mathbf{C}$ at all. If the operator S groups these components into a column vector, then for the "$\mathbf{MC}$" case we have

$$\mathbf{U}_a\mathbf{C}\mathbf{U}_b = \mathbf{T}_6 \tag{5}$$

$$\mathbf{U}_a\mathbf{C} = \mathbf{T}_6\mathbf{U}_b^{-1}$$

$$S(\mathbf{U}_a) = S\left(\mathbf{T}_6\mathbf{U}_b^{-1}\right) \tag{6}$$

and, similarly, for the "$\mathbf{CM}$" case:

$$S(\mathbf{U}_b) = S\left(\mathbf{U}_a^{-1}\mathbf{T}_6\right) \tag{7}$$

In the manner of (2), we can use these relations to

* MACSYMA is a large symbolic mathematics program distributed by Symbolics, Inc., Cambridge, Mass.

obtain different sets of equations by taking (5) and successively either premultiplying by the inverse components of $\mathbf{U}_a$, or postmultiplying by the inverse components of $\mathbf{U}_b$.

In formulating **C**, it may be convenient to allow the component matrices to differ slightly from the strict "A" matrix definition. We can do this as follows. An "A" matrix is the product of four components: A rotation about θ about the z axis, a translation d along the z axis, a translation a along the new x axis, and a rotation α about the new z axis:

$$\mathbf{A} = \mathbf{R}_\theta \mathbf{T}_d \mathbf{T}_a \mathbf{R}_\alpha \tag{8}$$

For a joint described by $\mathbf{A}_i$, we may migrate some of the constant terms into the adjacent matrices $\mathbf{A}_{i-1}$ or $\mathbf{A}_{i+1}$, requiring only that $\mathbf{A}_i$ remains dependent on its joint variable. Caution should be exercised in doing this since some analysis techniques involving "A" matrices do in fact assume the standard definition of (8).

We shall now study some applications of the approach, beginning with the well studied instance where three joint axes intersect. This makes it possible to formulate as **C** for the joints in question which has no translational component, which implies that

$$\mathbf{C} = \begin{pmatrix} \mathbf{R} & 0 \\ 0 & 1 \end{pmatrix} \tag{9}$$

where **R** is a 3×3 rotation matrix. Premultiplying this by any transform **X** leaves the fourth column invariant:

$$\mathbf{XCe}_4 = \mathbf{Xe}_4 \tag{10}$$

(where $\mathbf{e}_i$ is defined by $\mathbf{e}_i(j) = 0$ for $j \neq i$, $\mathbf{e}_i(i) = 1$).

In the well known "wrist partioned" case, **C** comprises the last three joints, so that

$$\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{C} = \mathbf{T}_6$$

Applying (6) and (10) allows us to generate the following equations:

$$\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{e}_4 = \mathbf{T}_6\mathbf{e}_4 \tag{11}$$

$$\mathbf{A}_2\mathbf{A}_3\mathbf{e}_4 = \mathbf{A}_1^{-1}\mathbf{T}_6\mathbf{e}_4 \tag{12}$$

$$\mathbf{A}_3\mathbf{e}_4 = \mathbf{A}_2^{-1}\mathbf{A}_1^{-1}\mathbf{T}_6\mathbf{e}_4 \tag{13}$$

This provides enough information to determine $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$. **C** can then be determined from equation (4):

$$\mathbf{C} = \mathbf{A}_3^{-1}\mathbf{A}_2^{-1}\mathbf{A}_1^{-1}\mathbf{T}_6 \tag{14}$$

The rest of the problem can be solved by applying method (2) to **C** alone:

$$\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 = \mathbf{C} \tag{15}$$

$$\mathbf{A}_5\mathbf{A}_6 = \mathbf{A}_4^{-1}\mathbf{C} \tag{16}$$

$$\mathbf{A}_6 = \mathbf{A}_5^{-1}\mathbf{A}_4^{-1}\mathbf{C} \tag{17}$$

If **C** comprises a set of intermediate links, such as (for example) 3, 4, and 5, then we obtain a different version of (11)–(13):

$$\mathbf{T}_6^{-1}\mathbf{A}_1\mathbf{A}_2\mathbf{e}_4 = \mathbf{A}_6^{-1}\mathbf{e}_4 \tag{18}$$

$$\mathbf{A}_1\mathbf{A}_2\mathbf{e}_4 = \mathbf{T}_6\mathbf{A}_6^{-1}\mathbf{e}_4 \tag{19}$$

$$\mathbf{A}_2\mathbf{e}_4 = \mathbf{A}_1^{-1}\mathbf{T}_6\mathbf{A}_6^{-1}\mathbf{e}_4 \tag{20}$$

along with equations analogous to (14)–(17) to solve for the component of **C**.

Another case of interest is when we have three rotational joints whose axes are parallel to each other. If we use one of these axes as a reference, then the three joints form a complete two dimensional system with one rotational freedom about the axis and two translational freedoms perpendicular to it. If we define this axis to be z, then **C** can be defined as

$$\mathbf{C} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & p_{cx} \\ \sin(\theta) & \cos(\theta) & 0 & p_{cy} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{21}$$

where p_{cx} and p_{cy} are the net displacements, and θ is the net rotation. Post multiplication **CX** of this matrix leaves the 3rd row invariant, which allows us to establish

$$\mathbf{e}_3^T\mathbf{CX} = \mathbf{e}_3^T\mathbf{X} \tag{22}$$

If we assume, for example, that **C** is formed from angles 2, 3, and 4 (as is the case for the "elbow" manipulator), then

$$\mathbf{A}_1\mathbf{CA}_5\mathbf{A}_6 = \mathbf{T}_6$$

and applying (7) and (22) gives the following equations:

$$\mathbf{e}_3^T\mathbf{A}_5 = \mathbf{e}_3^T\mathbf{A}_1^{-1}\mathbf{T}_6\mathbf{A}_6^{-1} \tag{23}$$

$$\mathbf{e}_3^T\mathbf{A}_5\mathbf{A}_6 = \mathbf{e}_3^T\mathbf{A}_1^{-1}\mathbf{T}_6 \tag{24}$$

$$\mathbf{e}_3^T\mathbf{A}_5\mathbf{A}_6\mathbf{T}_6^{-1} = \mathbf{e}_3^T\mathbf{A}_1^{-1} \tag{25}$$

This yields a solution for angles 1, 5, and 6. **C** is then known from (4).

$$\mathbf{C} = \mathbf{A}_1^{-1}\mathbf{T}_6\mathbf{A}_6^{-1}\mathbf{A}_5^{-1} \tag{26}$$

and given (21), is it easy to solve for the 3 component matrices which comprise **C** (see the end of Section 2.2).

Decouplings similar to the three parallel axis case can also occur for combinations of prismatic and revolute joints. Prismatic joints are generally much easier to deal with since they introduce no rotational motion. The case where any three adjacent joints are prismatic is easy to solve, since it represents a **C** in which the rotational component, if not 0, is at least constant. If $\mathbf{R}_a$, $\mathbf{R}_c$, $\mathbf{R}_b$, and $\mathbf{R}_6$ are the rotational submatrices of $\mathbf{U}_a$, **C**, $\mathbf{U}_b$ and $\mathbf{T}_6$, respectively, then $\mathbf{R}_c$ and $\mathbf{R}_6$ are known and we can formulate our base equations from the relationship

$$\mathbf{R}_a\mathbf{R}_c\mathbf{R}_b = \mathbf{R}_6 \tag{27}$$

and the component matrices of $\mathbf{R}_a$ and $\mathbf{R}_b$, which together contain only 3 independent variables.

Lastly, we consider the case where only two axes intersect. Though this problem is certainly not always tractable, it is common to most manipulators and hence worth some discussion. As in the case where three joints intersect, it is possible to construct a **C** matrix which has only a rotational component (though using only two component matrices this time). The difficulty is that the resulting equations we can generate for the translational part of the problem will depend on four joint variables instead of three. Even if these relationships are simple, we will still need an additional equation to solve for the subsystem. This can be found if the rotational component of **C** contains any constant entries (which is true when any of the component "twist" angles α_i are multiplies of $\pi/2$). We can then find the necessary equation from

$$\mathbf{R}_c = \mathbf{R}_a^T\mathbf{R}_6\mathbf{R}_b^T \tag{28}$$

by equating these elements which are constant on the left hand side with their corresponding elements on the right. This constraint is actually the one commonly used to solve the elbow manipulator [7,10], although in that instance it is not actually needed because of the three parallel axes.

2.2. Solving the Equations

In the rest of this paper, we will use the common kinematician's shorthand where $s_i = \sin\theta_i$, $c_i = \cos\theta_i$, $s_{ij} = \sin\theta_i + \sin\theta_j$, and $c_{ij} = \cos\theta_i + \cos\theta_j$. We will also make use of the function atan2(y, x), which takes two arguments, y and x, proportional, respectively, to the sine and cosine of some angle θ, and returns the value of that angle in the range $-\pi < \theta_i <= \pi$. This method of performing inverse trigonometry is preferred because it is numerically well behaved and easily resolves ambiguities about what quadrant the angle is in.

Once a reasonable set of equations has been determined by the methods discussed above, solutions are obtained in the same way detailed in [7]: The individual elements are examined for simple relationships describing the joint variables. Some of the common forms of these relationships for revolute joints are described here, along with their solutions.

In the most ideal case we will find two equations of the form

$$as_i = k_1 \tag{29}$$
$$ac_i = k_2$$

where k_1 and k_2 are given or known from a previous part of the solution. θ_i can then be determined unambiguously from

$$\theta_i = \text{atan2}\left(\frac{k_1}{k_2}\right) \tag{30}$$

except when $a = 0$, which probably indicates a singularity. Alternatively, we may find an equation

$$as_i + bc_i = k \tag{31}$$

whose solution may be expressed either as [7]

$$\theta_i = \text{atan2}\left(\frac{k}{\pm\sqrt{a^2+b^2-k^2}}\right) - \text{atan}\,2\left(\frac{b}{a}\right) \tag{32}$$

or [10],

$$\theta_i = 2\,\text{atan2}\left(\frac{a \pm \sqrt{a^2+b^2-k^2}}{k+b}\right) \tag{33}$$

If such relationships are not explicitly available, they can often be found by squaring and combining neighboring equations. Sometimes we may only have a convenient representation for either the sine or the cosine term. This can still be resolved using the above equations; if we have an expression for the cosine, then we can use (33) with $a = 0$ and $b = 1$ to get

$$\theta_i = 2\,\text{atan2}\left(\pm\frac{\sqrt{1-c^2}}{1+c}\right) \tag{34}$$

Another possibility is that an equation of the form (31) is be complemented by an additional equation to give a linear relationship,

$$a_1 s_i + b_1 c_i = k_1 \qquad (35)$$
$$a_2 s_i + b_2 c_i = k_2$$

which may be solved to give s_i and c_i explicitly. These are then used as arguments to atan2().

We conclude this section with the solution of **C** for the three parallel joint case discussed above. For notational convenience, we will assume that the three component joints are 2, 3, and 4. If the translational offset for each link is a_i*, we then have

$$\begin{pmatrix} c_{234} & -s_{234} & 0 & p_{cx} \\ s_{234} & c_{234} & 0 & p_{cy} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} c_{234} & -s_{234} & 0 & a_2c_2 + a_3c_{23} + a_4c_{234} \\ s_{234} & c_{234} & 0 & a_2s_2 + a_3s_{23} + a_4s_{234} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (36)$$

The solution to this can be determined with the law of cosines and a little algebra:

$$\theta_3 = 2 \operatorname{atan} 2\left(\frac{\pm\sqrt{1 + c_3^2}}{1 + c_3} \right) \qquad (37)$$

$$c_3 = \frac{a_2^2 + a_3^2 - d_x^2 - d_y^2}{2a_2a_3}$$

$$d_x = p_{cx} - c_{234}a_4$$

$$d_y = p_{cy} - s_{234}a_4$$

$$\theta_2 = \operatorname{atan} 2\left(\frac{a_3s_3d_x + (a_3c_3 + a_2)d_y}{(a_3c_3 + a_2)d_x + a_3s_3d_y} \right)$$

$$\theta_4 = \operatorname{atan} 2\left(\frac{s_{234}}{c_{234}} \right) - \theta_3 - \theta_2$$

Note that there is a singularity at $\theta_3 = 0$.

3. Forward and Inverse Jacobians

The decoupling which we have considered in the above section on kinematics is also important in any treatment of the manipulator Jacobian, where it will manifest itself as zero entries. It might even be preferable to work out the Jacobian before doing the kinematics (in analyzing a manipulator, one usually finds themself doing both, anyway) in case a partitioning is revealed that was not previously noticed.

It is well known that the Jacobian can be greatly simplified if it is expressed in some alternate frame k, instead of the canonical frame (which is usually in link 6 of the manipulator). This frame k is typically located in an intermediate manipulator link [11]. Such simplification is of particular interest for inversion purposes, although it must be remembered that it is then necessary to map Cartesian forces and velocities to and from frame k.

In establishing a simple (easy to invert) version of the Jacobian, we are interested that it not only be sparse, but also that it be block triangular, if possible. In particular, for wrist partioned manipulators, where the last three joints do not contribute to translation, the Jacobian is of the form

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_{11} & 0 \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix} \qquad (38)$$

Lets begin by reiterating the formulae for the column vectors $\mathbf{j}_i$ of **J** expressed in frame k: If the coordinate transform from frame k to the base frame of axis i (located in link $i-1$) is given by the matrix $\mathbf{K}_i$, where

$$\mathbf{K}_i = \begin{pmatrix} \mathbf{R}_i & \mathbf{p}_i \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{n}_i & \mathbf{o}_i & \mathbf{a}_i & \mathbf{p}_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (39)$$

then we have, using a notation similar to that in [14]:

$$\mathbf{j}_i = \begin{pmatrix} \mathbf{p}_i \times \mathbf{a}_i \\ \mathbf{a}_i \end{pmatrix} \text{ (rotary joint)} \qquad (40)$$

and

$$\mathbf{j}_i = \begin{pmatrix} \mathbf{a}_i \\ 0 \end{pmatrix} \text{ (prismatic joint)} \qquad (41)$$

(The perhaps more familiar formulae given in [7] are obtained using $\mathbf{K}_i^{-1}$ instead of $\mathbf{K}_i$.)

These equations can be used to establish some characteristics of the simplifying cases considered in section 1. If the three rotational joint axes intersect, and we chose our frame k to be coincident on this point of intersection, then $\mathbf{p}_i$ will be zero for each of the three columns associated with the joints in question. The top three elements of

* Because the joints are parallel, d_i can be set to 0.

each column will hence be zero, and so by permuting columns appropriately, it will be possible to transform **J** into form (38).

Next, we consider the case where three rotary joints are always parallel. If we set the z axis of k to be parallel to the joint axes, then $\mathbf{a}_i = (0, 0, 1)^T$ and the z components of $\mathbf{n}_i$ and $\mathbf{o}_i$ are 0 for each of the three associated columns. This gives us columns of the form

$$\mathbf{j}_i = \begin{pmatrix} (\mathbf{p}_i \times \mathbf{a}_i)_x \\ (\mathbf{p}_i \times \mathbf{o}_i)_y \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{42}$$

Again, by permuting rows and columns, it is possible to transform **J** into form (38) (note that the axes in question do not need to be adjacent; prismatic joints may lie between them, for instance).

The case of two intersecting rotary axes (say i and $i-1$) also results in a simplified Jacobian. If k is chosen to be the same frame as that of axis i, then $\mathbf{K}_i = \mathbf{I}$, and $\mathbf{K}_{i-1}$ is simply the inverse of the "A" matrix $\mathbf{A}_{i-1}$. Since the axes intersect, the translational components of this matrix are zero and the matrix is orthogonal:

$$\mathbf{K}_{i-1} = \mathbf{A}_i^{-1} = \begin{pmatrix} c_i & s_i & 0 & 0 \\ -s_i c_\alpha & c_i c_\alpha & s_\alpha & 0 \\ s_i s_\alpha & -c_i s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{43}$$

The Jacobian columns $\mathbf{j}_{i-1}$ and $\mathbf{J}_i$ associated with the two axes are hence

$$(\mathbf{j}_{i-1}, \mathbf{j}_i) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ s_\alpha & 0 \\ c_\alpha & 1 \end{pmatrix} \tag{44}$$

From this, we see that whenever at least 2 joint axes intersect, **J** may be expressed in the form of (38), where $\mathbf{J}_{11}$ is 4×4 and $\mathbf{J}_{22}$ is 2×2. This can be useful; even if numeric inversion is necessary, it is much less expensive to invert a 4×4 matrix than a 6×6.

Just to complete this part of the discussion, we see that by setting k equal to the frame of link 5, we are always able to obtain a Jacobian whose last column is $(0, 0, 0, 0, 0, 1)^T$ and hence have at most a 5×5 inversion to deal with.

Having seen that it is often possible to put the Jacobian into a block triangular form, we should see how we can use this to the best advantage computationally. It should be noted that we are not usually interested in inverting the Jacobian for its own sake, but rather we typically wish to solve for a vector. This may done at less cost than a full matrix inversion, and methods for doing this efficiently are now described.

There are two flavors of Jacobian calculation: the direct computation, which maps between differential joint values $\mathrm{d}\boldsymbol{\Theta}$ and Cartesian values $\mathrm{d}\mathbf{c}$, and the transpose computation, which maps between Cartesian forces **q** and joint torques **t**.

The direct Jacobian computation is

$$\begin{aligned} \mathrm{d}\mathbf{c} &= \mathbf{J}\, \mathrm{d}\boldsymbol{\Theta} \\ &= \begin{pmatrix} \mathbf{J}_{11} & 0 \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix} \mathrm{d}\boldsymbol{\Theta} \end{aligned} \tag{45}$$

The inverse of this is

$$\begin{aligned} \mathrm{d}\boldsymbol{\Theta} &= \mathbf{J}^{-1}\, \mathrm{d}\mathbf{c} \\ &= \begin{pmatrix} \mathbf{J}_{11}^{-1} & 0 \\ -\mathbf{J}_{22}^{-1}\mathbf{J}_{21}\mathbf{J}_{11}^{-1} & \mathbf{J}_{22}^{-1} \end{pmatrix} \mathrm{d}\mathbf{c} \end{aligned} \tag{46}$$

The transpose computation is used to determine joint torques:

$$\begin{aligned} \mathbf{t} &= \mathbf{J}^T\mathbf{q} \\ &= \begin{pmatrix} \mathbf{J}_{11}^T & \mathbf{J}_{21}^T \\ 0 & \mathbf{J}_{22}^T \end{pmatrix} \end{aligned} \tag{47}$$

The inverse of this calculation is

$$\begin{aligned} \mathbf{q} &= \mathbf{J}^{-1T}\mathbf{t} \\ &= \begin{pmatrix} \mathbf{J}_{11}^{-1T} & -\mathbf{J}_{11}^{-1T}\mathbf{J}_{21}^T\mathbf{J}_{22}^{-1T} \\ 0 & \mathbf{J}_{22}^{-1T} \end{pmatrix} \mathbf{t} \end{aligned} \tag{48}$$

The most efficient way to handle each of the above computations is to explicitly solve for each of the desired vectors. If we partition each of the input and output vectors into two components, corresponding to the different blocks of **J**.

$$\mathrm{d}\boldsymbol{\Theta} = \begin{pmatrix} \mathrm{d}\boldsymbol{\Theta}_1 \\ \mathrm{d}\boldsymbol{\Theta}_2 \end{pmatrix} \quad \mathrm{d}\mathbf{c} = \begin{pmatrix} \mathrm{d}\mathbf{c}_1 \\ \mathrm{d}\mathbf{c}_2 \end{pmatrix} \tag{49}$$

$$\mathbf{t} = \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{pmatrix} \quad \mathbf{q} = \begin{pmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{pmatrix} \tag{50}$$

then our computations arrange to look something like:

$$\mathbf{dc} = \begin{pmatrix} \mathbf{J}_{11}\,\mathbf{d\Theta}_1 \\ \mathbf{J}_{21}\,\mathbf{d\Theta}_1 + \mathbf{J}_{22}\,\mathbf{d\Theta}_2 \end{pmatrix} \tag{51}$$

$$\mathbf{d\Theta} = \begin{pmatrix} \mathbf{J}_{11}^{-1}\,\mathbf{dc}_1 \\ \mathbf{J}_{22}^{-1}(-\mathbf{J}_{21}\,\mathbf{d\Theta}_1 + \mathbf{dc}_2) \end{pmatrix} \tag{52}$$

$$\mathbf{t} = \begin{pmatrix} \mathbf{J}_{11}^T\mathbf{q}_1 + \mathbf{J}_{21}^T\mathbf{q}_2 \\ \mathbf{J}_{22}^T\mathbf{q}_2 \end{pmatrix} \tag{53}$$

$$\mathbf{q} = \begin{pmatrix} \mathbf{J}_{11}^{-1T}(\mathbf{t}_1 - \mathbf{J}_{21}^T\mathbf{q}_t) \\ \mathbf{J}_{22}^{-1T}\mathbf{t}_2 \end{pmatrix} \tag{54}$$

Likewise, the easiest way to invert the submatrices $\mathbf{J}_{11}$ and $\mathbf{J}_{22}$ is to solve explicitly for the vector in question. If we have the matrix equation $\mathbf{My} = \mathbf{x}$, where $\mathbf{y} = (y_1, y_2, y_3)^T$ and $\mathbf{x} = (x_1, x_2, x_3)^T$ are arbitrary vectors, then we use the notation sol(**M**) to denote the product $\mathbf{M}^{-1}\mathbf{x}$. This is best found in the usual way by putting the composite matrix (**Mx**) into echelon form and then solving for y recursively (elements of the solution vector **y** may hence be explicitly contained in the expression for sol(**M**)). The solution to equations (52) and (54) is completed by finding sol($\mathbf{J}_{11}$), sol($\mathbf{J}_{22}$), sol($\mathbf{J}_{11}^T$), and sol($\mathbf{J}_{22}^T$).

As was mentioned a little earlier, if the Jacobian is derived in an intermediate frame k, then we have to map quantities to and from the frame of interest (often the frame in link 6). If the transform from frame k to the frame in link 6 is $\mathbf{K}_6$, then the matrix $\mathbf{D}_{6k}$ which transforms velocities from **e** to k, and its inverse, $\mathbf{D}_{ke}$, are given by

$$\mathbf{D}_{6k} = \begin{pmatrix} \mathbf{R}_6 & \mathbf{p}_6 \times \mathbf{R}_6 \\ 0 & \mathbf{R}_6 \end{pmatrix} \quad \mathbf{D}_{k6} = \begin{pmatrix} \mathbf{R}_6^T & (\mathbf{p}_6 + \mathbf{R}_6)^T \\ 0 & \mathbf{R}_6^T \end{pmatrix} \tag{55}$$

where the notation $\mathbf{p} \times \mathbf{R}$ indicates a matrix whose columns are the cross products of **p** and each of the columns of **R**. Similarly, the force transformations $\mathbf{F}_{6k}$ and $\mathbf{F}_{k6}$ are given by

$$\mathbf{F}_{6k} = \begin{pmatrix} \mathbf{R}_6 & 0 \\ \mathbf{p}_6 \times \mathbf{R}_6 & \mathbf{R}_6 \end{pmatrix} \quad \mathbf{F}_{k6} = \begin{pmatrix} \mathbf{R}_6^T & 0 \\ (\mathbf{p}_6 \times \mathbf{R}_6)^T & \mathbf{R}_6^T \end{pmatrix} \tag{56}$$

Clearly, these transformations are much simpler to work with if there is no translation between frames k and 6.

4. Using MACSYMA to Determine the Kinematics

The methods described above are easy to implement using a computer algebra program. We will describe a few details about doing this here, using MACSYMA as an illustration.

4.1. Matrices

Most of the MACSYMA work is done using the matrix routines along with the regular algebra facilities. The matrix functions provide for most common matrix operations, including multiplication ($\cdot$) and inversion (*inv()*). Column vectors are also represented as matrices. Multiplying a column vector v by the inverse of a matrix *A1* would look like

```
inv(A1).v;
```

Basic algebraic simplification is performed using the function *ratsimp()*.

4.2. Trigonometric Simplification

Although MACSYMA has rules to handle trigonometric constructions, we found that it was faster (and more compact) to use our own trigonometric notation and provide the required identities explicitly. The trig expressions contained only sines and cosines, which were represented using the convention:

$$siX = \sin(\theta_X)$$

$$coX = \cos(\theta_X)$$

$$siXY = \sin(\theta_X + \theta_Y)$$

$$coXY = \cos(\theta_X + \theta_Y)$$

The only trig identities which we found necessary were

$$siX\hat{}2 + coX\hat{}2 = 1 \tag{57}$$

$$siXY\hat{}2 + coXY\hat{}2 = 1 \tag{58}$$

and

$$sinXY = siXcoY + siYcoX \tag{59}$$

$$cosXY = ciXcoY - siXsiY \tag{60}$$

MACSYMA allows users to specify a single substitution with the function *ratsubst()*, or a list of substitutions with the function *lratsubst()*. The

identities (57) and (58) can be wrapped into a function *tsimp()*, defined as

```
tsimp(x) := lratsubst([tid1, tid2, tid3, ... tid12,
                                    tid23, ...], x)
```

where *tidX* and *tidXY* correspond to identities (57) and (58) for the required joint angles.

Identities (59) and (60) are important whenever the axes of two consecutive joints are parallel. For any two joint angles X and Y, we can define a function *combineXY()* which combines angles, and a function *expandXY()* which expands them.

MACSYMA has some difficulty combining angles which are in the midst of a large expression, although this is usually not a problem since it is possible to combine the angles *before* making the expression complex. For example, if we have the matrices *A1, A2, A3,* and *A4*, then

```
combine23(A1.A2.A3.A4);
```

might have some difficulty, whereas

```
A1.combine23(A2.A3).A4;
```

would succeed. It is often useful to assign such predetermined products to a variable, as in:

```
A12:combine12(A1.A2);
```

For analyzing the Jacobian, several support functions were written. These include *jacobcolr(m)* and *jacobcolp(m)*, which return a column of the Jacobian for revolute and prismatic joints, respectively, given the transform matrix m between the frame of the joint in question and frame k. For historical reasons, our implementation of these functions uses the formula given in [7], rather than (40), so m is the inverse of the matrix $\mathbf{K}_i$ described in Section 2.

```
jacobcolr(m) := block([r],
   r:matrix(  [m[2, 1] * m[1, 4]
             -m[1, 1] * m[2, 4]),
   r:addrow(r, [m[2, 2] * m[1, 4]
             -m[1, 2] * m[2, 4]]),
   r:addrow(r, [m[2, 3] * m[1, 4]
             -m[1, 3] * m[2, 4]]),
   FOR i THRU 3 DO
        r:addrow(r, [m[3, i]]),
   r);

jacobcolp(m) := block ([r],
   r:matrix(  [m[3, 1]]),
   r:addrow(r, [m[3, 2]]),
   r:addrow(r, [m[3, 3]]),
   FOR i THRU 3 DO
        r:addrow(r, [0]),
   r);
```

For exchanging rows and columns of the Jacobian, we have defined *exrow(m, i, j)* and *excol(m, i, j)*, which exchange rows (or columns) i and j of m:

```
exrow(matrix, i, j) := block(
   [r, t],
   r:copymatrix(matrix),
   t:r[j], r[j]:r[i], r[i]:t,
   r);

excol(matrix, i, j) := block(
   [r, t],
   r:transpose(matrix),
   t:r[j], r[j]:r[i], r[i]:t,
   transpose (r));
```

Lastly, the functions *usolve(m, y, x)* and *lsolve(m, y, x)* can be used to determine sol(m) for a matrix which is either closer to upper triangular, or closer to lower triangular, respectively. The arguments x and y provide symbolic descriptions of the input and output vectors.

```
usolve(m, y, x) := block(
   [ue, nc],
   nc:length(m[1]) + 1,
   ue:uechelon(m, x),
   col(ue, nc) - (submatrix(ue, nc)
                  - ident(nc - 1)).y);

lsolve(m, y, x) := block(
   [ue, nc],
   nc:length (m[1]) + 1,
   ue:lechelon(m, x),
   col(ue, nc) - (submatrix(ue, nc)
                  - ident(nc - 1)).y);

uechelon(m, x) := echelon(addcol(m, x));

lechelon(m, x) := block(
   [r, nc],
   nc:length(m[1]) + 1,
   r:echelon(addcol (flipmatrix(m), reverse(x))),
   addcol (flipmatrix(submatrix(r, nc)),
                  reverse(col(r, Nc))));

flipmatrix(m)
   := reverse(transpose(reverse(transpose(m))));
```

5. Examples

In the following examples, the elements of $\mathbf{T}_6$ will be denoted in the usual way.

$$\mathbf{T}_6 = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{61}$$

while the elements of $\mathbf{C}$ will be denoted by

$$\mathbf{C} = \begin{pmatrix} n_{cx} & o_{cx} & a_{cx} & p_{cx} \\ n_{cy} & o_{cy} & a_{cy} & p_{cy} \\ n_{cz} & o_{cz} & a_{cz} & p_{cz} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{62}$$

5.1. Example 1: The ETL Robot

This is a 6 revolute joint robot designed at the Japanese Electrotechnical Laboratory (ETL). (*Fig. 1*). The "A" matrices for this robot are:

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{63}$$

$$\mathbf{A}_2 = \begin{pmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & -d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} c_3 & 0 & s_3 & d_4s_3 \\ s_3 & 0 & -c_3 & -c_3d_4 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Since this is a wrist partitioned manipulator, we start with equations (11)–(13), which yields several equivalently useful relationships. Equation (13) evaluates to

$$\begin{pmatrix} d_4s_3 \\ -c_3d_4 \\ d_3 \\ 1 \end{pmatrix} = \begin{pmatrix} p_ys_{12} + c_{12}p_x - a_1c_2 \\ p_z + d_2 \\ -a_1s_2 + p_xs_{12} - c_{12}p_y \\ 1 \end{pmatrix} \tag{64}$$

The second row gives a single cosine relationship for θ_3.

Squaring and adding rows 1 and 3 gives the equation

$$d_4^2s_3^2 + d_3^2 = -2a_1p_ys_1 + p_y^2 + p_x^2 - 2a_1c_1p_x + a_1^2 \tag{65}$$

which then provides a solution for θ_1.

Equation (12) gives us

$$\begin{pmatrix} c_2d_4s_3 + d_3s_2 \\ d_4s_2s_3 - c_2d_3 \\ -c_3d_4 - d_2 \\ 1 \end{pmatrix} = \begin{pmatrix} p_ys_1 + c_1p_x - a_1 \\ c_1p_y - p_xs_1 \\ p_z \\ 1 \end{pmatrix} \tag{66}$$

Rows 1 and 2 give a set of equations for s_2 and c_2 of the form

$$\begin{pmatrix} d_4s_3 & d_3 \\ -d_3 & d_4s_3 \end{pmatrix} \begin{pmatrix} c_2 \\ s_2 \end{pmatrix} = \begin{pmatrix} k_{21} \\ k_{21} \end{pmatrix} \tag{67}$$

which gives a solution for θ_2. A solution for θ_2

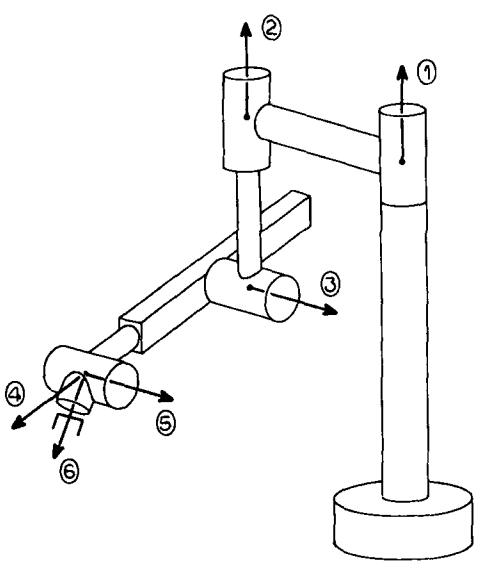


Fig. 1. The ETL robot.

could also be found by squaring and combining a pair of equations, as was done for θ_1.

One we have solved for the first three angles, we can solve for the wrist angles using equations (14) and (15)–(17). These equations are relatively simple since the wrist contains no offsets and the twist angles are multiples of $\pi/2$.

Elements (1.3) and (2.3) of equation (15) give a relation for angle θ_4:

$$\begin{pmatrix} c_4 s_5 \\ s_4 s_5 \end{pmatrix} = \begin{pmatrix} a_{cx} \\ a_{cy} \end{pmatrix} \tag{68}$$

We note the singularity when $\theta_5 = 0$.

Elements (1.3) and (2.3) of equation (16) yield a relation for θ_5:

$$\begin{pmatrix} s_5 \\ -c_5 \end{pmatrix} = \begin{pmatrix} a_{cy} s_4 + a_{cx} c_4 \\ -a_{cz} \end{pmatrix} \tag{69}$$

Finally, θ_6 can be determined from elements (1.1) and (2.1) of equation (17):

$$\begin{pmatrix} c_6 \\ s_6 \end{pmatrix} = \begin{pmatrix} c_5(n_{cy} s_4 + c_4 n_{cx}) - n_{cz} s_5 \\ c_4 n_{cy} - n_{cx} s_4 \end{pmatrix} \tag{70}$$

The complete inverse kinematics for the ETL robot can now be summarized:

$$\theta_3 = 2\ \text{atan2}\left(\frac{\pm\sqrt{d_4^2 - (p_z + d_2)^2}}{p_z + d_2 - d_4} \right) \tag{71}$$

$$\theta_1 = 2\ \text{atan2}\left(\frac{2a_1 p_y \pm \sqrt{4a_1^2 p_y^2 + 4a_1^2 p_x^2 - k_{11}^2}}{k_{11} + 2a_1 p_x} \right)$$

$$k_{11} = p_y^2 + p_x^2 + a_1^2 - d_4^2 s_3^2 - d_3^2$$

$$\theta_2 = 2\ \text{atan2}\left(\frac{d_3 k_{21} + d_4 s_3 k_{22}}{d_4 s_3 k_{21} - d_3 k_{22}} \right)$$

$$k_{21} = p_y s_1 + c_1 p_x - a_1$$

$$k_{22} = c_1 p_y - p_x s_1$$

$$\theta_4 = \text{atan2}\left(\frac{a_{cy}}{a_{cx}} \right)$$

$$\theta_5 = \text{atan2}\left(\frac{a_{cy} s_4 + a_{cx} c_4}{a_{cz}} \right)$$

$$\theta_6 = \text{atan2}\left(\frac{c_4 n_{cy} - n_{cx} s_4}{c_5(n_{cy} s_4 + c_4 n_{cx}) - n_{c2} s_5} \right)$$

where the rotational component of $\mathbf{C}$ is defined by

$$\begin{pmatrix} n_{cx} & o_{cx} & a_{cx} \\ n_{cy} & o_{cy} & a_{cy} \\ n_{cz} & o_{cz} & a_{cz} \end{pmatrix} = \begin{pmatrix} c_{12}c_3 & s_{12} & c_{12}s_3 \\ c_3 s_{12} & -c_{12} & s_{12}s_3 \\ s_3 & 0 & -c_3 \end{pmatrix}$$

We derive the Jacobian for the ETL robot in the frame defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. Because axes 1 and 2 are parallel, columns 1 and 2 can be combined:

$$\mathbf{j}_1' = \mathbf{j}_1 - \mathbf{j}_2 \tag{72}$$

The Jacobian relationship then takes the form

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} a_1 c_3 s_2 & c_3 d_3 & d_4 & 0 & 0 & 0 \\ -a_1 c_2 & -d_4 s_3 & 0 & 0 & 0 & 0 \\ a_1 s_2 s_3 & d_3 s_3 & 0 & 0 & 0 & 0 \\ 0 & s_3 & 0 & 0 & -s_4 & c_4 s_5 \\ 0 & 0 & 1 & 0 & c_4 & s_4 s_5 \\ 0 & -c_3 & 0 & 1 & 0 & c_5 \end{pmatrix} \times \begin{pmatrix} d\theta_1 \\ d\theta_2 - d\theta_1 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix} \tag{73}$$

Permuting columns and rows to optimize the triangularity of the diagonal blocks gives us the following:

$$\begin{pmatrix} dy \\ dz \\ dx \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} -a_1 c_2 & -d_4 s_3 & 0 & 0 & 0 & 0 \\ a_1 s_2 s_3 & d_3 s_3 & 0 & 0 & 0 & 0 \\ a_1 c_3 s_2 & c_3 d_3 & d_4 & 0 & 0 & 0 \\ 0 & s_3 & 0 & c_4 s_5 & -s_4 & 0 \\ 0 & 0 & 1 & s_4 s_5 & c_4 & 0 \\ 0 & -c_3 & 0 & c_5 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} d\theta_1 \\ d\theta_2 - d\theta_1 \\ d\theta_3 \\ d\theta_6 \\ d\theta_5 \\ d\theta_4 \end{pmatrix} \tag{74}$$

We will now solve for the various blocks of **J** using the notation described in section 2. Making the substitution

$$h = a_1 d_4 s_2 s_3 - a_1 c_2 d_3 \tag{75}$$

and solving the equation $\mathbf{J}_{11}\mathbf{y} = \mathbf{x}$ yields

$$\mathrm{sol}(\mathbf{J}_{11}) = \begin{pmatrix} \dfrac{d_4 x_2 + d_3 x_1}{h} \\ -\dfrac{a_1 c_2 y_1 + x_1}{d_4 s_3} \\ -\dfrac{c_3 d_3 y_2 + a_1 c_3 s_2 y_1 - x_3}{d_4} \end{pmatrix} \tag{76}$$

Solving for $\mathbf{J}_{22}$ yields, initially,

$$\begin{pmatrix} \dfrac{s_4 x_2 + c_4 x_1}{s_5} \\ -\dfrac{s_4 s_5 y_1 - x_2}{c_4} \\ x_3 - c_5 y_1 \end{pmatrix} \tag{77}$$

but this is not quite correct since the c_4 term in the denominator of the second row cancels out (this can be determined in advance by examining the determinant of $\mathbf{J}_{22}$ and noticing that there are no singularities at $c_4 = 0$). Substituting for y_1 in the second row gives

$$\mathrm{sol}(\mathbf{J}_{22}) = \begin{pmatrix} \dfrac{s_4 x_2 + c_4 x_1}{s_5} \\ c_4 x_2 - s_4 x_1 \\ x_3 - c_5 y_1 \end{pmatrix} \tag{78}$$

Similarly, we can find solutions for $\mathbf{J}_{11}^T$ and $\mathbf{J}_{22}^T$, respectively:

$$\mathrm{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} \dfrac{c_3 d_3 y_3 + d_3 s_3 y_2 - x_2}{d_4 s_3} \\ -\dfrac{c_3 h y_3 + a_1 c_2 x_2 - d_4 s_3 x_1}{h s_3} \\ \dfrac{x_3}{d_4} \end{pmatrix} \tag{79}$$

$$\mathrm{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} -\dfrac{c_4 c_5 y_3 + s_4 s_5 x_2 - c_4 x_1}{s_5} \\ -\dfrac{c_5 s_4 y_3 - c_4 s_5 x_2 - s_4 x_1}{s_5} \\ x_3 \end{pmatrix} \tag{80}$$

The rest of the solution follows from equations (51)–(54).

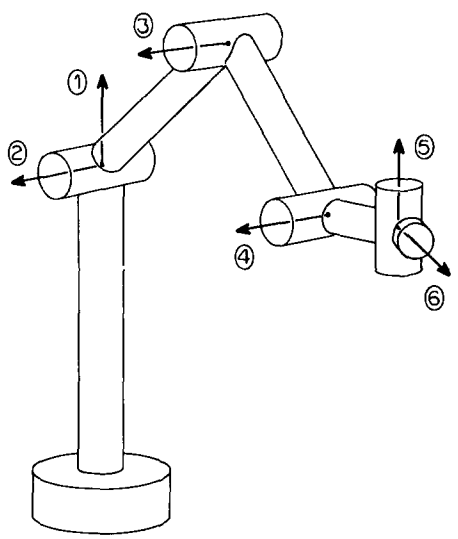


Fig. 2. The elbow manipulator.

5.2. Example 2: The Elbow Manipulator

The elbow manipulator (*Fig. 2*) is a common industrial robot in which the axes of joints 2.3, and 4 are parallel. The "A" matrices for this robot, with $\mathbf{A}_4$ and $\mathbf{A}_5$ modified so that the z axis defined by $\mathbf{A}_4$ is parallel to the z axes of $\mathbf{A}_2$ and $\mathbf{A}_3$, are:

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{81}$$

$$\mathbf{A}_2 = \begin{pmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & -s_4 & 0 & a_4 c_4 \\ s_4 & c_4 & 0 & a_4 s_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ 0 & 1 & 0 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Since this manipulator has three parallel axes, we can first use equations (23)–(25). We find equation (24) (presented in column form) to be

$$\begin{pmatrix} -c_6 s_5 \\ s_5 s_6 \\ c_5 \\ 0 \end{pmatrix} = \begin{pmatrix} n_x s_1 - c_1 n_y \\ o_x s_1 - c_1 o_y \\ a_x s_1 - a_y c_1 \\ p_x s_1 - c_1 p_y \end{pmatrix} \tag{82}$$

The 4th row of this gives a relation for θ_1, while rows 1 and 2 provide a relation for θ_6. We note the existence of a singularity when $\theta_5 = 0$.

Next, we examine equation (23):

$$\begin{pmatrix} -s_5 \\ 0 \\ c_5 \\ 0 \end{pmatrix} = \begin{pmatrix} s_1(c_6 n_x - o_x s_6) - c_1(c_6 n_y - o_y s_6) \\ s_1(n_x s_6 + c_6 o_x) - c_1(n_y s_6 + c_6 o_y) \\ a_x s_1 - a_y c_1 \\ p_x s_1 - c_1 p_y \end{pmatrix} \tag{83}$$

This gives an unambiguous solution for θ_5, since the other two angles have been solved for.

Having solved for the three angles which surround **C**, we now solve for the components of **C** itself (which are of the form given in (21)). Beginning begin with the equation

$$\mathbf{C} = \mathbf{A}_1^{-1}\mathbf{T}_6\mathbf{A}_6^{-1}\mathbf{A}_5^{-1} \tag{84}$$

we find the second row, presented in column form, is

$$\begin{pmatrix} s_{234} \\ c_{234} \\ 0 \\ p_{cy} \end{pmatrix} = \begin{pmatrix} -c_5 o_z s_6 + a_z s_5 + c_5 c_6 n_z \\ n_z s_6 + c_6 o_z \\ o_z s_5 s_6 - c_6 n_z s_5 + a_z c_5 \\ p_z \end{pmatrix} \tag{85}$$

This gives a relation for θ_{234} and p_{cy}.

For p_{cx}, we see from inspection of the (1.4) elements of (84) that

$$p_{cx} = p_y s_1 + c_1 p_x \tag{86}$$

The results for the elbow manipulator can now be summarized:

$$\theta_1 = 2\ \text{atan2}\left(\frac{p_x \pm \sqrt{p_x^2 + p_y^2}}{-p_y} \right) \tag{87}$$

$$\theta_6 = \text{atan2}\left(\frac{o_x s_1 - c_1 o_y}{c_1 n_y - n_x s_1} \right)$$

$$\theta_5 = \text{atan2}\left(\frac{s_1(c_6 n_x - o_x s_6) - c_1(c_6 n_y - o_y s_6)}{a_x s_1 - a_y c_1} \right)$$

$$s_{234} = -c_5 o_z s_6 + a_z s_5 + c_5 c_6 n_z$$

$$c_{234} = n_z s_6 + c_6 o_z$$

$$p_{cy} = p_z$$

$$p_{cx} = p_y s_1 + c_1 p_x$$

The remainder of the solution follows from equations (37).

We can use the same "A" matrices to derive a Jacobian for the manipulator in the frame k defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$. Since three axes are parallel, we can simplify the Jacobian by combining columns. Specifically, we take

$$\mathbf{j}_2' = \mathbf{j}_2 - \mathbf{j}_3 \tag{88}$$

and

$$\mathbf{j}_3' = \mathbf{j}_3 - \mathbf{j}_4 \tag{89}$$

The Jacobian relationship is then defined by

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} 0 & a_2 s_{34} & a_3 s_4 & 0 & 0 & 0 \\ 0 & a_2 c_{34} & a_3 c_4 & a_4 & 0 & 0 \\ -a_4 c_{234} - a_3 c_{23} - a_2 c_2 & 0 & 0 & 0 & 0 & 0 \\ s_{234} & 0 & 0 & 0 & 0 & s_5 \\ c_{234} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & c_5 \end{pmatrix} \times \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 - d\theta_2 \\ d\theta_4 - d\theta_3 \\ d\theta_5 \\ d\theta_6 \end{pmatrix} \tag{90}$$

Using the substitution

$$h = -a_4 c_{234} - a_3 c_{23} - a_2 c_2 \tag{91}$$

and rearranging columns to migrate the zero block into the upper right corner and optimize the triangularity of the diagonal blocks, we get the relationship

$$\begin{pmatrix} dz \\ d\psi \\ d\rho \\ d\phi \\ dx \\ dy \end{pmatrix} = \begin{pmatrix} h & 0 & 0 & 0 & 0 & 0 \\ s_{234} & s_5 & 0 & 0 & 0 & 0 \\ c_{234} & 0 & 1 & 0 & 0 & 0 \\ 0 & c_5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_2 s_{34} & a_3 s_4 & 0 \\ 0 & 0 & 0 & a_2 c_{34} & a_3 c_4 & a_4 \end{pmatrix} \times \begin{pmatrix} d\theta_1 \\ d\theta_6 \\ d\theta_5 \\ d\theta_2 \\ d\theta_3 - d\theta_2 \\ d\theta_4 - d\theta_3 \end{pmatrix} \tag{92}$$

The solution for

$$\mathbf{J}_{11}\mathbf{y} = \mathbf{x} \tag{93}$$

works out to

$$\mathrm{sol}(\mathbf{J}_{11}) = \begin{pmatrix} -\dfrac{x_1}{h} \\ \dfrac{x_2 - s_{234} y_1}{s_5} \\ x_3 - c_{234} y_1 \end{pmatrix} \tag{94}$$

Solving for $\mathbf{J}_{22}$ initially yields

$$\begin{pmatrix} -\dfrac{s_4 x_3 - c_4 x_2 - a_4 s_4 x_1}{a_2 s_3} \\ -\dfrac{a_2 s_{34} y_1 - x_2}{a_3 s_4} \\ x_1 \end{pmatrix} \tag{95}$$

but the s_4 in the denominator of the second row cancels out to yield

$$\mathrm{sol}(\mathbf{J}_{22}) = \begin{pmatrix} -\dfrac{s_4 x_3 - c_4 x_2 - a_4 s_4 x_1}{a_2 s_3} \\ \dfrac{s_{34} x_3 - c_{34} x_2 - a_4 s_{34} x_1}{a_3 s_3} \\ x_1 \end{pmatrix} \tag{96}$$

Similarly, the solutions for $\mathbf{J}_{11}^T$ and $\mathbf{J}_{22}^T$ are, respectively,

$$\mathrm{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} \dfrac{c_{234} y_3 + s_{234} y_2 - x_1}{h} \\ \dfrac{x_2}{s_5} \\ x_3 \end{pmatrix} \tag{97}$$

and

$$\mathrm{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} x_3 - a_4 y_3 \\ -\dfrac{a_2 c_{34} x_2 - a_3 c_4 x_1}{a_2 a_3 s_3} \\ \dfrac{a_2 s_{34} x_2 - a_3 s_4 x_1}{a_2 a_3 s_3} \end{pmatrix} \tag{98}$$

The rest of the solution follows easily from equations (51)–(54).

6. Summary

The methods described above can be encapsulated into the following procedures:

6.1. Working out the Kinematics

1. *Enter all the "A" matrices*, modifying them as necessary to define a reasonable **C** matrix.
2. *Build the equations* by multiplying the necessary matrices and equating elements. There will typically be two equation sets: a set of column vector equations in terms of the angles extrinsic to **C**, and a set of matrix equations in terms of the angles intrinsic to **C**.
3. *Solve for the angles outside* **C**, first by inspection, and secondly by combining equal elements.
4. *Solve for the angles of* **C** in terms of **C**, in the same way. The elements of **C** are known from step 3.

6.2. Working out the Jacobian

1. *Establish the frame k in which the J is to be derived.* This should be optimized with regard to obtaining a simple Jacobian while not allowing the computation of $\mathbf{K}_6$ to become too complicated.
2. *Derive* **J** using equations (40) and (41) and the successive $\mathbf{K}_i$ found by multiplying "A" matrices. The functions *jacobcolr()* and *jacobcolp()* do this using values of $\mathbf{K}_i^{-1}$ instead.
3. *Rearrange J to simplify the inversion.* Permute the rows and columns of **J** to first put the matrix into a block triangular form, if possible, and then arrange each of the blocks themselves to be as triangular as possible. Additional simplification can be obtained by adding or sub-

tracting columns which correspond to parallel joints.

4. *Find the inverse relationships.* The blocks can be inverted using the functions *usolve()* and *lsolve()*, and the rest of the solutions are obtained directly from (51)–(54).

References

[1] Jorge Angeles: "On the Numerical Solution of the Inverse Kinematic Problem". *International Journal of Robotics Research*, Summer 1985, pp. 21–37 (Vol. 4, No. 2).

[2] Duffy and Crane: "A Displacement Analysis of the General Spatial 7-link, 7R mechanism". *Mechanism and Machine Theory*, 1980, pp. 153–169 (Vol. 15, No. 3).

[3] R. Featherstone: "Position and Velocity Transformations between Robot End-effector Coordinates and Joint Angles". *International Journal of Robotics Research*, Summer 1983, pp. 35–45, (Vol. 2, No. 2).

[4] John M. Hollerbach and Gideon Sahar: "Wrist Partitioned, Inverse Kinematic Accelerations and Manipulator Dynamics". *International Journal of Robotics Research*, Winter 1983, pp. 61–76 (Vol. 2, No. 4).

[5] K.H. Hunt: "The Particular or the General? (Some Examples from Robot Kinematics)". *Mechanism and Machine Theory*, 1986, pp. 481–487 (Vol. 21, No. 6).

[6] C.S.G. Lee: "Robot Arm Kinematics, Dynamics, and Control". *Computer*, December 1982, pp. 61–80 (Vol. 15, No. 12).

[7] Richard Paul: *Robot Manipulators: Mathematics, Programming, and Control*. The MIT Press, Cambridge, Massachusetts, 1981.

[8] Richard Paul, Marc Renaud, and Charles N. Stevenson: "A Systematic Approach for Obtaining the Kinematics of Recursive Manipulators Based on Homogeneous Transformations". *Robotics Research – The first International Symposium*, Michael Brady and Richard Paul, Editors. The MIT Press, Cambridge, Massachusetts, 1984, pp. 707–726.

[9] Richard P. Paul and Hong Zhang: "Computationally Efficient Kinematics for Manipulators with Spherical Wrists Based on the Homogeneous Transform Representation". *International Journal of Robotics Research*, Summer 1986, pp. 32–44 (Vol. 5, No. 2).

[10] G.R. Pennock and A.T. Yang: "Application of Dual-Number Matrices to the Inverse Kinematics Problem of Robot Manipulators". *Journal of Mechanisms, Transmissions, and Automation in Design*, June, 1985, pp. 201–208 (Vol. 197, No. 2).

[11] Marc Renaud: "Geometric and Kinematic Models of a Robot Manipulator: Calculation of the Jacobian Matrix and Its Inverse". *Proceedings of the 11th International Symposium on Industrial Robots*, Tokyo, Japan, October 7–9, 1981, pp. 757–763.

[12] Bernard Roth, J. Rastegar, and V. Scheinman: "On the Design of Computer Controlled Manipulators". *On the Theory and Practice of Robots and Manipulators*, First CISM-IFToMM Symposium, September 1973, pp. 93–113 (Vol. 1).

[13] L.-W. Tsai and A.P. Morgan: "Solving the Kinematics of Most General Six- and Five-degree of Freedom Manipulators by Continuation Methods". *Journal of Mechanisms, Transmissions, and Automation in Design*, June, 1985, pp. 189–200 (Vol. 197, No. 2).

[14] K.J. Waldron, Shih-Liang Wang, and S.J. Bolin: "A Study of the Jacobian Matrix of Serial Manipulators". *Journal of Mechanisms, Transmissions, and Automation in Design*, June, 1985, pp. 230–238 (Vol. 197, No. 2).

Appendix A. MACSYMA Sessions

A.1 Derivation of ETL Robot Kinematics

This is an annotated MACSYMA session used in deriving the inverse kinematics for the ETL robot.

; The variables a1 through a6 denote the 6 A matrices for the
; robot while i1 through i6 denote their inverses. The link pa-
; rameters are given by aa1, dd2, dd3, and dd4. The sines and
; cosines are given by si1 through si6 and co1 through co6. si12
; and co12 are the sine and cosine of angle 1 + angle 2.

; For later convenience, precombine a1.a2 and its inverse:

(c127) a12: combine12 (a1.a2);

$$(d127)\begin{bmatrix} co12 & 0 & si12 & aa1co1 \\ si12 & 0 & -co12 & aa1si1 \\ 0 & 1 & 0 & -dd2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c128) i12: combine12 (tsimp(invert(a2)).tsimp(invert(a1)));

$$(d128)\begin{bmatrix} co12 & si12 & 0 & -aa1co2 \\ 0 & 0 & 1 & dd2 \\ si12 & -co12 & 0 & -aa1si2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

; Now form the first three equations to find angles 1 through 3.
; The vector e4 is used to extract the fourth column of each
; equation.

(c129) e4: transpose (matrix([0, 0, 0, 1]));

$$(d129)\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

(c130) eqn1: a12.a3.e4 = t6.e4;

$$(d130)\begin{bmatrix} co12\,dd4\,si3 + dd3\,si12 + aa1co1 \\ dd4\,si12\,si3 + aa1si1 - co12\,dd3 \\ -co3\,dd4 - dd2 \\ 1 \end{bmatrix} = \begin{bmatrix} px \\ py \\ pz \\ 1 \end{bmatrix}$$

(c131) eqn2: a2.a3.e4 = i1.t6.e4;

$$(d131)\begin{bmatrix} co2dd4si3 + dd3si2 \\ dd4si2si3 - co2dd3 \\ -co3dd4 - dd2 \\ 1 \end{bmatrix} = \begin{bmatrix} pysi1 + co1px - aa1 \\ co1py - pxsi1 \\ pz \\ 1 \end{bmatrix}$$

(c132) eqn3 : a3.e4 = i12.t6.e4;

$$(d132)\quad \begin{bmatrix} dd4si3 \\ -co3dd4 \\ dd3 \\ 1 \end{bmatrix} = \begin{bmatrix} pysi12 + co12px - aa1co2 \\ pz + dd2 \\ -aa1si2 + pxsi12 - co12py \\ 1 \end{bmatrix}$$

; Extract solution for angle 3:

(c133) part(eqn3, 1)[2, 1] = part(eqn3, 2)[2, 1];

(d133) $-co3dd4 = pz + dd2$

; Extract solution for angle 1;

(c134) x : part(eqn3, 2);

$$(d134)\quad \begin{bmatrix} pysi12 + co12px - aa1co2 \\ pz + dd2 \\ -aa1si2 + pxsi12 - co12py \\ 1 \end{bmatrix}$$

(c135) part(eqn3, 1);

$$(d135)\quad \begin{bmatrix} dd4si3 \\ -co3dd4 \\ dd3 \\ 1 \end{bmatrix}$$

(c136) x[1, 1]^2 + x[3, 1]^2

= tsimp(expand12(x[1, 1]^2 + x[3, 1]^2));

(d136) $dd4^2si3^2 + dd3^2$

$= -2aa1pysi1 + py^2 + px^2 - 2aa1co1px + aa1^2$

; Extract solution for angle 2

(c137) eqn2;

$$(d137)\quad \begin{bmatrix} co2dd4si3 + dd3si2 \\ dd4si2si3 - co2dd3 \\ -co3dd4 - dd2 \\ 1 \end{bmatrix} = \begin{bmatrix} pysi1 + co1px - aa1 \\ co1py - pxsi1 \\ pz \\ 1 \end{bmatrix}$$

(c138) x : matrix([dd4 * si3, dd3], [− dde, dd4 * si3]);

$$(d138)\quad \begin{bmatrix} dd4si3 & dd3 \\ -dd3 & dd4si3 \end{bmatrix}$$

(c139) invert(x);

$$(d139)\quad \begin{bmatrix} \dfrac{dd4si3}{dd4^2si3^2 + dd3^2} & -\dfrac{dd3}{dd4^2si3^2 + dd3^2} \\ \dfrac{dd3}{dd4^2si3^2 + dd3^2} & \dfrac{dd4si3}{dd4^2si3^2 + dd3^2} \end{bmatrix}$$

(c140) k1 = part(eqn2, 2)[1, 1];

(d140) $k1 = pysi1 + co1px - aa1$

(c141) k2 = part(eqn2, 2)[2, 1];

(d141) $k2 = co1py - pxsi1$

(c142) print (matrix([co2], [si2]),

" = ", invert(x), matrix([k1], [k2]));

$$\begin{bmatrix} co2 \\ si2 \end{bmatrix} = \begin{bmatrix} \dfrac{dd4si3}{dd4^2si3^2 + dd3^2} & -\dfrac{dd3}{dd4^2si3^2 + dd3^2} \\ \dfrac{dd3}{dd4^2si3^2 + dd3^2} & \dfrac{dd4si3}{dd4^2si3^2 + dd3^2} \end{bmatrix} \begin{bmatrix} k1 \\ k2 \end{bmatrix}$$

; Derive the solutions for the last three joint angles. For conven-
; ience, define a macro 'rot' which returns the rotational part
; of a transform.

(c143) rot(m) := submatrix(4, m, 4);

(d143) rot(m) := submatrix(4, x, 4)

; Evaluate C in terms of a1 through a3, and then the find the
; component angles of C.

(c144) c;

$$(d144)\quad \begin{bmatrix} ncx & ocx & acx & pcx \\ ncy & ocy & acy & pcy \\ ncz & ocz & acz & pcz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c145) rot(c) = rot(a12.a3);

$$(d145)\quad \begin{bmatrix} ncx & ocx & acx \\ ncy & ocy & acy \\ ncz & ocz & acz \end{bmatrix} = \begin{bmatrix} co12co3 & si12 & co12si3 \\ co3si12 & -co12 & si12si3 \\ si3 & 0 & -co3 \end{bmatrix}$$

(c146) rot(a4.a5.a6) = rot(c);

$$\begin{bmatrix} co4co5co6 - si4si6 & -co4co5si6 - co6si4 & co4si5 \\ co4si6 + co5co6si4 & co4co6 - co5si4si6 & si4si5 \\ -co6si5 & si5si6 & co5 \end{bmatrix} = \begin{bmatrix} ncx & ocx & acx \\ ncy & ocy & acy \\ ncz & ocz & acz \end{bmatrix}$$

(c147) col(part(%, 1), 3) = col(part(%, 2), 3);

$$(d147)\quad \begin{bmatrix} co4si5 \\ si4si5 \\ co5 \end{bmatrix} = \begin{bmatrix} acx \\ acy \\ acz \end{bmatrix}$$

(c148) rot(a5.a6) = rot(i4.c);

$$(d148)\quad \begin{bmatrix} co5co6 & -co5si6 & si5 \\ co6si5 & -si5si6 & -co5 \\ si6 & co6 & 0 \end{bmatrix} = \begin{bmatrix} ncysi4 + co4ncx & ocysi4 + co4ocx & acysi4 + acxco4 \\ -ncz & -ocz & -acz \\ co4ncy - ncxsi4 & co4ocy - ocxsi4 & acyco4 - acxsi4 \end{bmatrix}$$

(c149) col(part(%, 1), 3) = col(part(%, 2), 3);

$$(d149)\quad \begin{bmatrix} si5 \\ -co5 \\ 0 \end{bmatrix} = \begin{bmatrix} acysi4 + acxco4 \\ -acz \\ acyco4 - acxsi4 \end{bmatrix}$$

(c150) rot(a6);

$$(d150)\quad \begin{bmatrix} co6 & -si6 & 0 \\ si6 & co6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(c151) col(%, 1) = col(rot(i5.i4.c), 1);

$$(d151)\quad \begin{bmatrix} co6 \\ si6 \\ 0 \end{bmatrix} = \begin{bmatrix} co5(ncysi4 + co4ncx) - nczsi5 \\ co4ncy - ncxsi4 \\ (ncysi4 + co4ncx)si5 + co5ncz \end{bmatrix}$$

A.2 Derivation of Elbow Manipulator Jacobian

This is an annotated MACSYMA session to illustrate the derivation and manipulation of Jacobians.

; For the elbow manipulator, the link offsets are given by the
; variables aa2, aa3, and aa4. The composite A matrices a23,
; a34, and a234, along with their inverses i23, i34, and i234,
; are precomputed to help MACSYMA keep the associated angles
; combined.

; Start by computing and simplifying all six columns of the jacobian:

(c111) j1 : tsimp(jacobcolr(a1.a234));

$$(d111)\quad \begin{bmatrix} 0 \\ 0 \\ -aa4co234 - aa3co23 - aa2co2 \\ si234 \\ co234 \\ 0 \end{bmatrix}$$

(c112) j2 : tsimp(jacobcolr(a2.a3.a4));

$$(d112)\quad \begin{bmatrix} (aa2co3 + aa3)si4 + aa2co4si3 \\ -aa2si3si4 + (aa2co3 + aa3)co4 + aa4 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

(c113) j3 : tsimp(jacobcolr(a3.a4));

$$(d113)\quad \begin{bmatrix} aa3si4 \\ aa3co4 + aa4 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

(c114) j4 : tsimp(jacobcolr(a4));

$$(d114)\quad \begin{bmatrix} 0 \\ aa4 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

; k is the matrix between the frame coinciding with joint 5
; and the Jacobian frame. This is not the identity since the
; z axis of k is not parallel with the axis of joint 5.

(c115) display (k);

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

(c116) j5 : tsimp(jacobcolr(k));

$$(d116)\quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

(c117) j6 : jacobcolr(i5);

$$(d117)\quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ si5 \\ 0 \\ co5 \end{bmatrix}$$

; Simplify the columns by subtraction:

(c118) j2 : combine34(j2 − j3)$

(c119) j3 : j3 − j4$

; Now form the Jacobian:

(c120) j : addcol(j1, j2, j3, j4, j5, j6);

$$\begin{bmatrix} 0 & aa2si34 & aa3si4 & 0 & 0 & 0 \\ 0 & aa2co34 & aa3co4 & aa4 & 0 & 0 \\ -aa4co234 - aa3co23 - aa2co2 & 0 & 0 & 0 & 0 & 0 \\ si234 & 0 & 0 & 0 & 0 & si5 \\ co234 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & co5 \end{bmatrix}$$

; Exchange various rows and columns to move the zero block
; into the upper right hand corner and optimize the triangularity
; of the blocks:

(c121) exrow(%, 1, 3)$

(c122) exrow(%, 2,'4)$

(c123) exrow(%, 3, 5)$

(c124) excol(%, 3, 5)$

(c125) excol(%, 2, 5)$

(c126) excol(%, 4, 6)$

(c127) excol(%, 4, 6)$

; Now we have the permuted version of the Jacobian.

(c128) h : − aa4 * co234 − aa3 * co23 − aa2 * co2$

(c129) jp : ratsubst('h, h, d127);

$$(d129)\quad \begin{bmatrix} h & 0 & 0 & 0 & 0 & 0 \\ si234 & si5 & 0 & 0 & 0 & 0 \\ co234 & 0 & 1 & 0 & 0 & 0 \\ 0 & co5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & aa2si34 & aa3si4 & 0 \\ 0 & 0 & 0 & aa2co34 & aa3co4 & aa4 \end{bmatrix}$$

; Define the sub-blocks of the matrix:

(c130) j11 : submatrix(4, 5, 6, jp, 4, 5, 6);

$$(d130)\quad \begin{bmatrix} h & 0 & 0 \\ si234 & si5 & 0 \\ co234 & 0 & 1 \end{bmatrix}$$

(c131) j22 : submatrix(1, 2, 3, jp, 1, 2, 3);

$$(d131)\quad \begin{bmatrix} 0 & 0 & 1 \\ aa2si34 & aa3si4 & 0 \\ aa2co234 & aa3co4 & aa4 \end{bmatrix}$$

(c132) j21 : submatrix(1, 2, 3, jp, 4, 5, 6);

$$(d132)\quad \begin{bmatrix} 0 & co5 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

; Solve j11 recursively, and simpllfy:

(c133) tsimp(lsolve(j11, y, x));

$$(d133)\quad \begin{bmatrix} \frac{x1}{h} \\ -\frac{si234y1 - x2}{si5} \\ x3 - co234y1 \end{bmatrix}$$

; Solve j22 recursively, and simplify:

(c134) v : tsimp(lsolve(j22, y, x));

$$(d134)\quad \begin{bmatrix} \frac{si4x3 - co4x2 - aa4si4x1}{aa2co34si4 - aa2co4si34} \\ -\frac{aa2si34y1 - x2}{aa3si4} \\ x1 \end{bmatrix}$$

(c135) co34 * si4 − co4 * si34;

(d135) co24si4 − co4si34

(c136) v : ratsubst(tsimp(expand34(%)), %, v);

$$(d136)\quad \begin{bmatrix} -\frac{si4x3 - co4x2 - aa4si4x1}{aa2si3} \\ -\frac{aa2si34y1 - x2}{aa3si4} \\ x1 \end{bmatrix}$$

; Substitute y1 into the second row to eliminate si4:

(c137) tsimp(ratsubst(v[1, 1], y1, v));

$$(d137)\quad \begin{bmatrix} -\dfrac{si4x3 - co4x2 - aa4si4x1}{aa2si3} \\ -\dfrac{si34x3 - co34x2 - aa4si34x1}{aa3si3} \\ x1 \end{bmatrix}$$

; Solve for j11 recursively, and simplify:

(c138) tsimp(usolve(transpose(j11), y, x));

$$(d138)\quad \begin{bmatrix} \dfrac{co234y3 + si234y2 - x1}{h} \\ \dfrac{x2}{si5} \\ x3 \end{bmatrix}$$

; Solve for j22 recursively, and simplify:

(c139) v: tsimp(usolve(transpose(j22), y, x));

$$(d139)\quad \begin{bmatrix} x3 - aa4y3 \\ -\dfrac{aa3co4y3 - x2}{aa3si4} \\ -\dfrac{aa2si34x2 - aa3si4x1}{aa2aa3co34si4 - aa2aa3co4si34} \end{bmatrix}$$

*(c140) co34 * si4 − co4 * si34;*

(d140) co34si4 − co4si34

(c141) v: ratsubst(tsimp(expand34(%)), %, v);

$$(d141)\quad \begin{bmatrix} x3 - aa4y3 \\ -\dfrac{aa3co4y3 - x2}{aa3si4} \\ \dfrac{aa2si34x2 - aa3si4x1}{aa2aa3si3} \end{bmatrix}$$

(c142) tsimp(ratsubst(v[3, 1], y3, v);

$$(d142)\quad \begin{bmatrix} x3 - aa4y3 \\ -\dfrac{aa2co34x2 - aa3co4x1}{aa2aa3si3} \\ \dfrac{aa2si34x2 - aa3si4x1}{aa2aa3si3} \end{bmatrix}$$

; Lastly, determine the matrix k6 which maps from k to link 6:

(c143) k6: a5.a6;

$$(d143)\quad \begin{bmatrix} co5co6 & -co5si6 & si5 & 0 \\ si6 & co6 & 0 & 0 \\ -co6si5 & si5si6 & co5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix B. Examples with Other Robots

B.1 The PUMA

Probably the most famous of all research robots, the PUMA (*Fig. 3*) is an anthropomorphic arm with six revolute joints. It is also one of the more complicated industrial arms, since it has 4 offsets.

The modified "A" matrices are defined as follows:

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} c_3 & 0 & -s_3 & a_3c_3 - d_4s_3 \\ s_3 & 0 & c_3 & a_3s_3 + c_3d_4 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0. \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse kinematics solution is

$$\theta_1 = 2\,\mathrm{atan2}\left(\frac{p_x \pm \sqrt{p_x^2 + p_y^2 - d_3^2}}{d_3 - p_y}\right)$$

$$\theta_2 = \mathrm{atan2}\left(\frac{2a_2p_z \pm \sqrt{4a_2^2\left(p_z^2 + k_{11}^2\right) - k_{21}^2}}{k_{21} - 2a_2k_{11}}\right)$$

$$k_{11} = p_ys_1 + c_1p_x$$

$$k_{21} = p_z^2 + k_{11}^2 + a_2^2 - d_4^2 - a_3^2$$

$$\theta_3 = \mathrm{atan2}\left(\frac{-d_4k_{31} + a_3k_{32}}{a_3k_{31} + d_4k_{32}}\right)$$

$$k_{31} = p_zs_2 + c_2k_{11} - a_2$$

$$k_{32} = c_2p_z - k_1s_2$$

$$\theta_4 = \mathrm{atan2}\left(\frac{-a_{cy}}{-a_{cx}}\right)$$

$$\theta_5 = \mathrm{atan2}\left(\frac{-a_{cy}s_4 - a_{cx}c_4}{a_{cz}}\right)$$

$$\theta_6 = \mathrm{atan2}\left(\frac{-o_{cz}s_5 - c_5\left(o_{cy}s_4 + c_4o_{cx}\right)}{c_4o_{cy} - o_{cx}s_4}\right)$$

where the rotational components of **C** can be computed from the first three angles by

$$\begin{pmatrix} n_{cx} & o_{cx} & a_{cx} \\ n_{cy} & o_{cy} & a_{cy} \\ n_{cz} & o_{cz} & a_{cz} \end{pmatrix} = \begin{pmatrix} c_1c_{23} & -s_1 & -c_1s_{23} \\ c_{23}s_1 & c_1 & -s_1s_{23} \\ s_{23} & 0 & c_{23} \end{pmatrix}$$

The manipulator Jacobian is derived in the link defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. This gives a $\mathbf{K}_6$ matrix defined by

$$\mathbf{K}_6 = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - c_6s_4 & -c_4s_5 & 0 \\ c_4s_6 + c_5c_6s_4 & c_4c_6 - c_5s_4s_6 & -s_4s_5 & 0 \\ c_6s_5 & -s_5s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The corresponding Jacobian relationship is

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} c_{23}d_3 & a_2s_3 & -d_4 & 0 & 0 & 0 \\ h & 0 & 0 & 0 & 0 & 0 \\ -d_3s_{23} & a_2c_3 & a_3 & 0 & 0 & 0 \\ s_{23} & 0 & 0 & 0 & s_4 & -c_4s_5 \\ 0 & 0 & -1 & 0 & -c_4 & -s_4d_5 \\ c_{23} & 0 & 0 & 1 & 0 & c_5 \end{pmatrix} \times \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 - d\theta_2 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix}$$

where rows 2 and 3 have been combined and h is defined by

$$h = -d_4s_{23} + a_3c_{23} + a_2c_2$$

The permuted form of the relationship is

$$\begin{pmatrix} dy \\ dx \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} c_{23}d_3 & a_2s_3 & -d_4 & 0 & 0 & 0 \\ -d_3s_{23} & a_2c_3 & a_3 & 0 & 0 & 0 \\ s_{23} & 0 & 0 & -c_4s_5 & s_4 & 0 \\ 0 & 0 & -1 & -s_4s_5 & -c_4 & 0 \\ c_{23} & 0 & 0 & c_5 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 - d\theta_2 \\ d\theta_6 \\ d\theta_5 \\ d\theta_4 \end{pmatrix}$$

The solution is given by

$$\mathrm{sol}(\mathbf{J}_{11}) = \begin{pmatrix} \dfrac{x_1}{h} \\ \dfrac{(d_3d_4s_{23} - a_3c_{23}d_3)y_1 + d_4x_3 + a_3x_2}{a_2a_3s_3 + a_2c_3d_4} \\ \dfrac{x_3 - a_2c_3y_2 + d_3s_{23}y_1}{a_3} \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}) = \begin{pmatrix} -\dfrac{s_4x_2 + c_4x_1}{s_5} \\ s_4x_1 - c_4x_2 \\ x_3 - c_5y_1 \end{pmatrix}$$

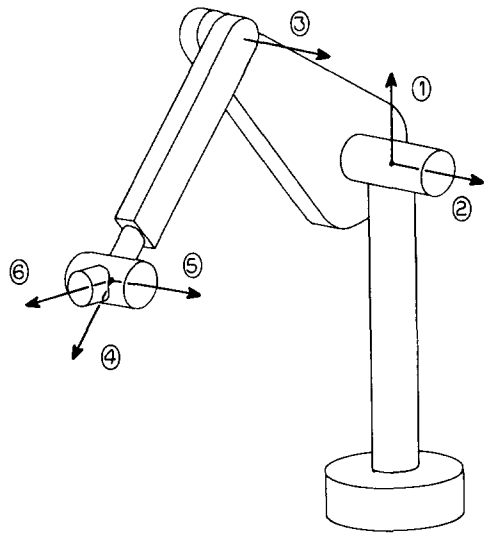


Fig. 3. The PUMA manipulator.

$$\mathrm{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} \dfrac{d_3s_{23}y_3 - c_{23}d_3y_2 + x_1}{h} \\ \dfrac{a_3y_3 - x_3}{d_4} \\ \dfrac{a_2s_3x_3 + d_4x_2}{a_2a_3s_3 + a_2c_3d_4} \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} \dfrac{c_45c_5y_3 + s_4s_5x_2 - c_4x_1}{s_5} \\ \dfrac{c_5s_4y_3 - c_4s_5x_2 - s_4x_1}{s_5} \\ x_3 \end{pmatrix}$$

B.2 The Stanford Manipulator

The Stanford manipulator (*Fig. 4*) is a spherical robot with an offset at the shoulder.

The "A" matrices for the robot are:

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

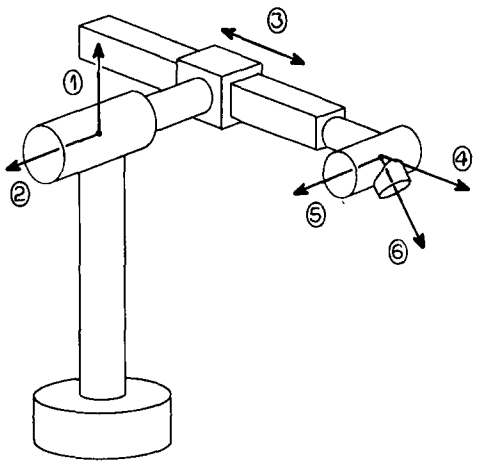


Fig. 4. The Stanford manipulator.

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse kinematic solution is

$$\theta_1 = 2\,\text{atan}\,2\left(\frac{-p_x \pm \sqrt{p_x^2 + p_y^2 - d_2^2}}{d_2 + p_y}\right)$$

$$d_3 = \sqrt{(p_y s_1 + c_1 p_x)^2 + p_z^2}$$

$$\theta_2 = \text{atan2}\left(\frac{p_y s_1 + c_1 p_x}{p_z}\right)$$

$$\theta_4 = \text{atan2}\left(\frac{a_{cy}}{a_{cx}}\right)$$

$$\theta_5 = \text{atan2}\left(\frac{a_{cy} s_4 + a_{cx} c_4}{a_{cx}}\right)$$

$$\theta_6 = \text{atan2}\left(\frac{o_{cz} s_5 - c_5(o_{cy} s_4 + c_4 o_{cx})}{c_4 o_{cy} - o_{cx} s_4}\right)$$

where the rotational components of $\mathbf{C}$ can be computed from the first three angles by

$$\begin{pmatrix} n_{cx} & o_{cx} & a_{cx} \\ n_{cy} & o_{cy} & a_{cy} \\ n_{cz} & o_{cz} & a_{cz} \end{pmatrix} = \begin{pmatrix} c_1 c_2 & -s_1 & c_1 s_2 \\ c_2 s_1 & c_1 & s_1 s_2 \\ -s_2 & 0 & c_2 \end{pmatrix}$$

The manipulator Jacobian is derived in the link defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. This gives a $\mathbf{K}_6$ matrix defined by

$$\mathbf{K}_6 = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - c_6 s_4 & c_4 s_5 & 0 \\ c_4 s_6 + c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & s_4 s_5 & 0 \\ -c_6 s_5 & s_5 s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The corresponding Jacobian relationship is

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} -c_2 d_2 & d_3 & 0 & 0 & 0 & 0 \\ d_3 s_2 & 0 & 0 & 0 & 0 & 0 \\ -d_2 s_2 & 0 & 1 & 0 & 0 & 0 \\ -s_2 & 0 & 0 & 0 & -s_4 & c_4 s_5 \\ 0 & 1 & 0 & 0 & c_4 & s_4 s_5 \\ c_2 & 0 & 0 & 1 & 0 & c_5 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix}$$

The permuted form of the relationship is

$$\begin{pmatrix} dy \\ dx \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} d_3 s_2 & 0 & 0 & 0 & 0 & 0 \\ -c_2 d_2 & d_3 & 0 & 0 & 0 & 0 \\ -d_2 s_2 & 0 & 1 & 0 & 0 & 0 \\ -s_2 & 0 & 0 & c_4 s_5 & -s_4 & 0 \\ 0 & 1 & 0 & s_4 s_5 & c_4 & 0 \\ c_2 & 0 & 0 & c_5 & 0 & 1 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_6 \\ d\theta_5 \\ d\theta_4 \end{pmatrix}$$

The solution to this is given by

$$\text{sol}(\mathbf{J}_{11}) = \begin{pmatrix} \dfrac{x_1}{d_3 s_2} \\ \dfrac{c_2 d_2 y_1 + x_2}{d_3} \\ d_2 s_2 y_1 + x_3 \end{pmatrix}$$

$$\text{sol}(\mathbf{J}_{22}) = \begin{pmatrix} \dfrac{s_4 x_2 + c_4 x_1}{s_5} \\ c_4 x_2 - s_4 x_1 \\ x_3 - c_5 y_1 \end{pmatrix}$$

$$\text{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} \dfrac{d_2 s_2 y_3 + c_2 d_2 y_2 + x_1}{d_3 s_2} \\ \dfrac{x_2}{d_3} \\ x_3 \end{pmatrix}$$

$$\text{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} -\dfrac{c_4 c_5 y_3 + s_4 s_5 x_2 - c_4 x_1}{s_5} \\ -\dfrac{c_5 s_4 y_3 - c_4 s_5 x_2 - s_4 x_1}{s_5} \\ x_3 \end{pmatrix}$$

B.3 The SCARA, type 1

The SCARA robots are wrist partioned manipulators where the first three joints consist of two rotary joints and one prismatic joint, with all the joint axes parallel.

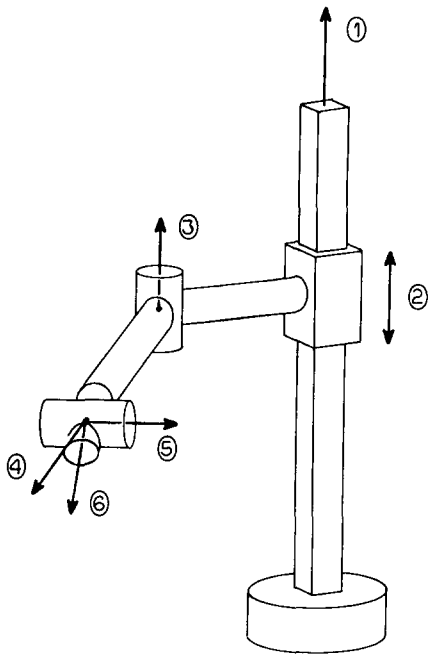


Fig. 5. The SCARA manipulator, type 1.

In the first type of SCARA robot (*Fig. 5*), the prismatic joint is joint 2.

The "A" matrices for this robot are

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} c_3 & 0 & s_3 & d_4 s_3 \\ s_3 & 0 & -c_3 & -c_3 d_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse kinematic solution is

$$\theta_1 = 2\,\text{atan}\,2\left(\frac{2a_2 p_y \pm \sqrt{4a_2^2\left(p_y^2 + p_x^2\right) - k_{11}^2}}{k_{11} + 2a_2 p_x}\right)$$

$$k_{11} = p_y^2 + p_x^2 - d_4^2 + a_2^2$$

$$d_2 = p_z$$

$$\theta_3 = \text{atan2}\left(\frac{p_y s_1 + c_1 p_x - a_2}{p_x s_1 - c_1 p_y}\right)$$

$$\theta_4 = \text{atan2}\left(\frac{a_{cy}}{a_{cx}}\right)$$

$$\theta_5 = \text{atan2}\left(\frac{a_{cy} s_4 + a_{cx} c_4}{a_{cz}}\right)$$

$$\theta_6 = \text{atan2}\left(\frac{o_{cz} s_5 - c_5\left(o_{cy} s_4 + c_4 o_{cx}\right)}{c_4 o_{cy} - o_{cx} s_4}\right)$$

where the rotational components of $\mathbf{C}$ can be computed from the first three angles by

$$\begin{pmatrix} n_{cx} & o_{cx} & a_{cx} \\ n_{cy} & o_{cy} & a_{cy} \\ n_{cz} & o_{cz} & a_{cz} \end{pmatrix} = \begin{pmatrix} c_{13} & 0 & s_{13} \\ s_{13} & 0 & -c_{13} \\ 0 & 1 & 0 \end{pmatrix}$$

The manipulator Jacobian is derived in the link defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. This gives a $\mathbf{K}_6$ matrix defined by

$$\mathbf{K}_6 = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - c_6 s_4 & c_4 s_5 & 0 \\ c_4 s_6 + c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & s_4 s_5 & 0 \\ -c_6 s_5 & s_5 s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The corresponding Jacobian relationship is

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} a_2 s_3 & 0 & d_4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -a_2 c_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -s_4 & c_4 s_5 \\ 0 & 0 & 0 & 1 & 0 & c_5 \\ & & & & & \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 - d\theta_1 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix}$$

where columns 1 and 3 have been combined. The permuted form of the relationship is

$$\begin{pmatrix} dz \\ dy \\ dx \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} -a_2 c_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ a_2 s_3 & 0 & d_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_4 s_5 & -s_4 & 0 \\ 0 & 0 & 1 & s_4 s_5 & c_4 & 0 \\ 0 & 0 & 0 & c_5 & 0 & 1 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 - d\theta_1 \\ d\theta_6 \\ d\theta_5 \\ d\theta_4 \end{pmatrix}$$

The solution to this is given by

$$\text{sol}(\mathbf{J}_{11}) = \begin{pmatrix} -\dfrac{x_1}{a_2 c_3} \\ x_2 \\ -\dfrac{a_2 s_3 y_1 - x_3}{d_4} \end{pmatrix}$$

$$\text{sol}(\mathbf{J}_{22}) = \begin{pmatrix} \dfrac{s_4 x_2 + c_4 x_1}{s_5} \\ c_4 x_2 - s_4 x_1 \\ x_3 - c_5 y_1 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} \dfrac{a_2 s_3 y_3 - x_1}{a_2 c_3} \\ x_2 \\ \dfrac{x_3}{d_4} \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} -\dfrac{c_4 c_5 y_3 + s_4 s_5 x_2 - c_4 x_1}{s_5} \\ -\dfrac{c_5 s_4 y_3 - c_4 s_5 x_2 - s_4 x_1}{s_5} \\ x_3 \end{pmatrix}$$

B.4 The SCARA, Type 2

In the second type of SCARA robot (*Fig. 6*), the prismatic joint is joint 3. The "A" matrices for the robot are:

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & -s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

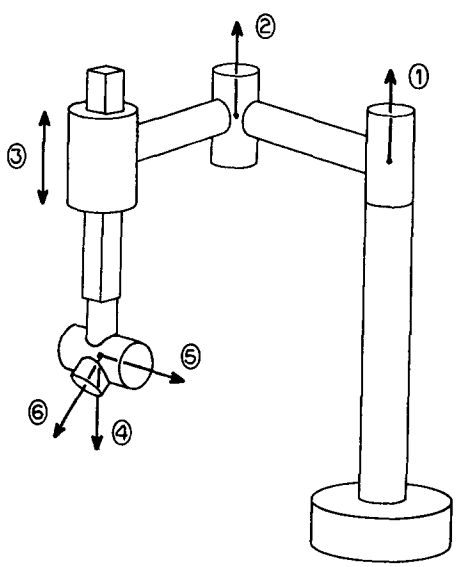


Fig. 6. The SCARA manipulator, type 2.

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse kinematic solution is

$$\theta_1 = 2\,\mathrm{atan}\,2\left(\frac{2a_1 p_y \pm \sqrt{4a_1^2(p_y^2 + p_x^2) - k_{11}^2}}{k_{11} + 2a_1 p_x} \right)$$

$$k_{11} = p_y^2 + p_x^2 - a_2^2 + a_1^2$$

$$\theta_2 = \mathrm{atan}\,2\left(\frac{c_1 p_y - p_x s_1}{p_y s_1 + c_1 p_x - a_1} \right)$$

$$d_3 = p_z$$

$$\theta_4 = \mathrm{atan}\,2\left(\frac{-a_{cy}}{-a_{cx}} \right)$$

$$\theta_5 = \mathrm{atan}\,2\left(\frac{-a_{cy} s_4 - a_{cx} c_4}{a_{cx}} \right)$$

$$\theta_6 = \mathrm{atan}\,2\left(\frac{-o_{cz} s_5 - c_5(o_{cy} s_4 + c_4 o_{cx})}{c_4 o_{cy} - o_{cx} s_4} \right)$$

where the rotational components of $\mathbf{C}$ can be computed from the first three angles by

$$\begin{pmatrix} n_{cx} & o_{cx} & a_{cx} \\ n_{cy} & o_{cy} & a_{cy} \\ n_{cz} & o_{cz} & a_{cz} \end{pmatrix} = \begin{pmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The manipulator Jacobian is derived in the link defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$. This gives a $\mathbf{K}_6$ matrix defined by

$$\mathbf{K}_6 = \begin{pmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - c_6 s_4 & -c_4 s_5 & 0 \\ c_4 s_6 + c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & -s_4 s_5 & 0 \\ c_6 s_5 & -s_5 s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The corresponding Jacobian relationship is

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} a_1 s_2 & 0 & 0 & 0 & 0 & 0 \\ a_1 c_2 & a_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_4 & -c_4 s_5 \\ 0 & 0 & 0 & 0 & -c_4 & -s_4 s_5 \\ 0 & 1 & 0 & 1 & 0 & c_5 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 - d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix}$$

where columns 1 and 2 have been combined. The permuted form of the relationship is

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} a_1 s_2 & 0 & 0 & 0 & 0 & 0 \\ a_1 c_2 & a_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -c_4 s_5 & s_4 & 0 \\ 0 & 0 & 0 & -s_4 s_5 & -c_4 & 0 \\ 0 & 1 & 0 & c_5 & 0 & 1 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 - d\theta_2 \\ d\theta_3 \\ d\theta_6 \\ d\theta_5 \\ d\theta_4 \end{pmatrix}$$

The solution is given by

$$\mathrm{sol}(\mathbf{J}_{11}) = \begin{pmatrix} \dfrac{x_1}{a_1 s_2} \\ -\dfrac{a_1 c_2 y_1 - x_2}{a_2} \\ x_3 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}) = \begin{pmatrix} -\dfrac{s_4 x_2 + c_4 x_1}{s_5} \\ s_4 x_1 - c_4 x_2 \\ x_3 - c_5 y_1 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} -\dfrac{a_1 c_2 y_2 - x_1}{a_1 s_2} \\ \dfrac{x_2}{a_2} \\ x_3 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} \dfrac{c_4 c_5 y_3 + s_4 s_5 x_2 - c_4 x_1}{s_5} \\ \dfrac{c_5 s_4 y_3 - c_4 s_5 x_2 - s_4 x_1}{s_5} \\ x_3 \end{pmatrix}$$

B.5 The Microbo ECUREUIL

This is a cylindrical robot manufactured by the Swiss company Microbo (*Fig. 7*).

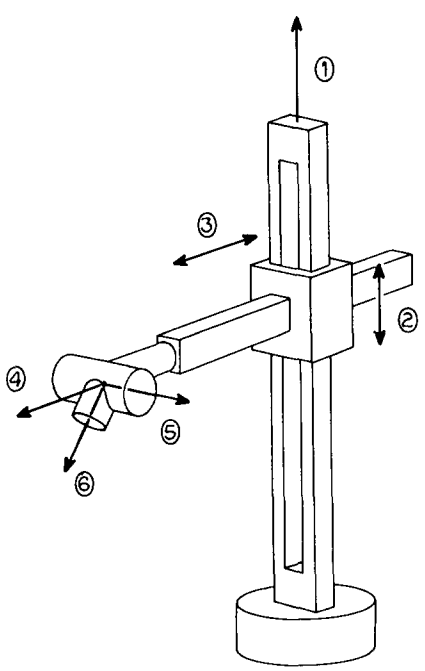


Fig. 7. The microbo ECUREUIL.

It is cylindrical, with two prismatic joints, and consequently its kinematics are quite straightforward.

The "A" matrices for this robot are:

$$\mathbf{A}_1 = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The inverse kinematic solution is

$$\theta_1 = \mathrm{atan2}\left(\frac{p_y}{p_x}\right)$$

$$d_2 = p_z$$

$$d_3 = \sqrt{p_x^2 + p_y^2}$$

$$\theta_4 = \mathrm{atan2}\left(\frac{a_{cy}}{a_{cx}}\right)$$

$$\theta_5 = \mathrm{atan2}\left(\frac{a_{cy} s_4 + a_{cx} c_4}{-a_{cx}}\right)$$

$$\theta_6 = \mathrm{atan2}\left(\frac{-o_{cz} s_5 - c_5 (o_{cy} s_4 + c_4 o_{cx})}{o_{cx} s_4 - c_4 o_{cy}}\right)$$

where the rotational components of $\mathbf{C}$ can be computed from the first three angles by

$$\begin{pmatrix} n_{cx} & o_{cx} & a_{cx} \\ n_{cy} & o_{cy} & a_{cy} \\ n_{cz} & o_{cz} & a_{cz} \end{pmatrix} = \begin{pmatrix} -s_1 & 0 & c_1 \\ c_1 & 0 & s_1 \\ 0 & 1 & 0 \end{pmatrix}$$

The Jacobian is derived in the frame defined by $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$, from which $\mathbf{K}_6$ is defined by

$$\mathbf{K}_6 = \begin{pmatrix} c_5 c_6 & -c_5 s_6 & s_5 & 0 \\ c_6 s_5 & -s_5 s_6 & -c_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The resulting Jacobian relationship is

$$\begin{pmatrix} dx \\ dy \\ dz \\ d\psi \\ d\rho \\ d\phi \end{pmatrix} = \begin{pmatrix} c_4 d_3 & s_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ d_3 s_4 & -c_4 & 0 & 0 & 0 & 0 \\ s_4 & 0 & 0 & 0 & 0 & s_5 \\ 0 & 0 & 0 & 1 & 0 & -c_5 \\ -c_4 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{pmatrix}$$

The permuted form of this relationship is

$$\begin{pmatrix} dx \\ dz \\ dy \\ d\psi \\ d\phi \\ d\rho \end{pmatrix} = \begin{pmatrix} c_4 d_3 & s_4 & 0 & 0 & 0 & 0 \\ d_3 s_4 & -c_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ s_4 & 0 & 0 & s_5 & 0 & 0 \\ -c_4 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -c_5 & 0 & 1 \end{pmatrix} \begin{pmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_6 \\ d\theta_5 \\ d\theta_4 \end{pmatrix}$$

The solution is given by

$$\mathrm{sol}(\mathbf{J}_{11}) = \begin{pmatrix} \dfrac{s_4 x_2 + c_4 x_1}{d_3} \\ s_4 x_1 - c_4 x_2 \\ x_3 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}) = \begin{pmatrix} \dfrac{x_1}{s_5} \\ x_2 \\ c_5 y_1 + x_3 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{11}^T) = \begin{pmatrix} \dfrac{d_3 s_4 x_2 + c_4 x_1}{d_3} \\ -\dfrac{c_4 d_3 x_2 - s_4 x_1}{d_3} \\ x_3 \end{pmatrix}$$

$$\mathrm{sol}(\mathbf{J}_{22}^T) = \begin{pmatrix} \dfrac{c_5 y_3 + x_1}{s_5} \\ x_2 \\ x_3 \end{pmatrix}.$$