MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo No. 717 April, 1983

Wrist-Partitioned Inverse Kinematic Accelerations and Manipulator Dynamics

John M. Hollerbach
Gideon Sahar

Abstract. An efficient algorithm is presented for the calculation of the inverse kinematic accelerations for a 6 degree-of-freedom manipulator with a spherical wrist. The inverse kinematic calculation is shown to work synergistically with the inverse dynamic calculation, producing kinematic parameters needed in the recursive Newton-Euler dynamics formulation. Additional savings in the dynamics computation are noted for a class of kinematically well-structured manipulators such as spherical-wrist arms and for manipulators with simply-structured inertial parameters.

Acknowledgements. 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 Systems Development Foundation.

1. Introduction

A precise specification of the trajectory of the end effector is a prerequisite towards successful application of a manipulator to many tasks. Arc welding, spray painting, conveyor belt tracking, and glueing are some tasks which require specification of both the spatial and temporal aspects of a trajectory. In the most general case, not only the Cartesian position and velocity of the end effector, or hand, must be specified, but also the Cartesian accelerations.

The transformation of a Cartesian trajectory of the hand into the corresponding joint angle trajectory of the manipulator, the so-called inverse kinematics problem, has been studied primarily in the context of positions and velocities. Reasonably efficient algorithms have been developed for these transformations (Paul 1981, Featherstone 1983), yet little attention has been paid to the solution of the inverse kinematic accelerations.

In this paper, we present an efficient algorithm for the calculation of the inverse kinematic accelerations for a 6 degree-of-freedom (dof) manipulator with a spherical wrist, based on a technique developed by Featherstone (1983) for inverse kinematic positions and velocities. In addition, we show that the inverse kinematic calculations work synergistically with the inverse dynamic calculations, because the extended Featherstone method yields kinematic parameters needed in the backward recursion steps of the Newton-Euler dynamics formulation (Luh, Walker, and Paul 1980a). We note that consistency argues that dynamics be particularized as well to spherical-wrist arms, resulting in considerable computational savings. Lastly, we examine simplifications in the dynamics computation due to simply-structured inertial parameters as well as to simply-structured kinematic parameters.

1.1. Inverse Kinematic Positions

A benign kinematic structure is a characteristic of most manipulators. Whereas kinematicians might choose to treat arbitrary linkages, manipulators are usually designed to satisfy simplifying kinematic criteria:

(i) there is the kinematic equivalent of a spherical wrist, and

(ii) neighboring joint axes are oriented at 0° or 90° relative to each other.

Pieper (1968) originally showed that a wrist with three intersecting axes of rotation, which is kinematically equivalent to a spherical wrist, is one of the configurations that leads to an analytic inverse kinematic position calculation. The spherical wrist allows a decomposition of the 6-dof inverse kinematics computation into two 3-dof kinematic computations, through a separation of the orientation specification from the position specification. Most 6-dof manipulators are designed with spherical wrists, making this the most important case.

If the manipulator does not satisfy one of Pieper's criteria, then in general the inverse kinematic positions must be found by time-consuming numerical approximation and convergence methods. This renders real-time control of a Cartesian trajectory infeasible (but see the discussion of resolved rate control below). Some manipulators that were originally designed without spherical wrists have been redesigned for this reason. Such manipulators might have been intended to be programmed by a human operator manually guiding the manipulator to desired positions with a teach box, so that the human operator in effect substituted for an inverse kinematics solution. As the manipulator's programming was increasingly automated with higher-level computer control, the non-spherical wrist was found to be an insurmountable stumbling block to automatic real-time generation of Cartesian trajectories.

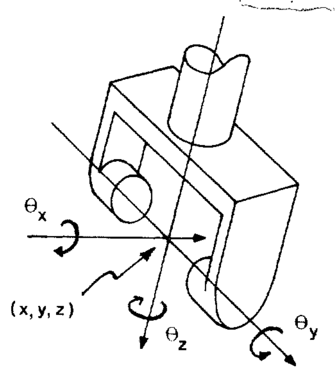

Figure 1. Cartesian position and orientation specification for a reference point on the hand of a manipulator.

1.2. Inverse Kinematic Velocities

Shortly after Pieper's work, Whitney (1969, 1972) proposed the Resolved Motion Rate Control Method, which computes the inverse kinematic velocities while avoiding the computation of the inverse kinematic positions. Whereas an analytic inverse kinematic position calculation is critically dependent on a benign manipulator configuration, the inverse kinematic velocities can be easily computed for any arbitrary 6-dof manipulator. If $\underline{\theta} = (\theta_1, \ldots, \theta_6)$ is the vector of joint angles and $\mathbf{x} = (x_1, \ldots, x_6)$ is the vector specifying the position $(x_1, x_2, x_3) = (x, y, z)$ and orientation $(x_4, x_5, x_6) = (\theta_x, \theta_y, \theta_z)$ of the reference point on the hand (Fig. 1), then the forward kinematic positions are straightforwardly given by the transformation $\mathbf{x} = \mathbf{f}(\underline{\theta})$. The forward kinematic velocities are then given by

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\underline{\theta}} \tag{1}$$

where the elements J_{ij} of the Jacobian **J** are $\partial f_i / \partial x_j$ with $\mathbf{f} = (f_1, \ldots, f_6)$. The E-2 manipulator that Whitney presented in his papers did not have a spherical wrist, and Whitney proposed that the trajectory be differentially approximated by solving the inverse kinematic velocities through inversion of (1) to avoid the intractible inverse kinematic position calculation:

$$\dot{\underline{\theta}} = \mathbf{J}^{-1}\dot{\mathbf{x}} \tag{2}$$

Due to the prohibitive computational cost of solving (2) through 6 × 6 matrix inversion (Table 1), Whitney suggested an interpolation scheme based on precomputed inverse Jacobians evaluated at a few positions.

It should be noted that the inverse Jacobian itself is often not of interest, but only the joint rates. Thus Gaussian elimination could have been used instead of matrix inversion, giving a substantial computational savings (Table 1); the numbers include evaluation of **J**.

More recently, Paul (1981) proposed a method particularized to the Stanford manipulator which is somewhat more efficient than Gaussian elimination (Table 1); the numbers given are slightly larger than stated in (Paul 1981). Paul's method involves manipulation of 4 × 4 transformation matrices, and takes implicit advantage of the spherical-wrist configuration of the Stanford manipulator.

Method	Multiplications	Additions
6 × 6 matrix inversion	287	193
Gaussian elimination	141	98
Paul (1981)	94	55
Featherstone (1983)	36	20

Table 1. Computational complexity of various methods of computing the inverse kinematic velocities for a 6-dof manipulator, in terms of the total number of multiplications and additions required, for the Stanford arm.

Featherstone (1983) takes more explicit advantage of the spherical-wrist configuration of manipulators to propose a highly efficient method for computation of the inverse kinematic velocities. The method is composed of four steps.

(i) Find the linear velocity of the wrist from the hand linear and angular velocity.

(ii) Find the first three joint rates from the wrist linear velocity.

(iii) Find the angular velocity of the hand relative to the forearm.

(iv) Find the last three joint rates from the relative angular velocity of the hand.

Featherstone's method cannot be directly compared to Paul's method because it was developed for a different manipulator and because the Cartesian velocity specification was different. We have reworked Featherstone's method for the Stanford manipulator with Paul's dT_6 Cartesian velocity specification, where the wrist linear velocity and the time derivative of the hand orientation matrix are directly given. Thus the first step in Featherstone's method is unnecessary for purposes of the comparison. Table 1 shows that Featherstone's method is almost 3 times more efficient than Paul's method. Featherstone's method works so well because he most directly takes advantage of the spherical wrist kinematics of robots. This is particularly important in computing the wrist joint rates, where Paul's method requires 75 multiplications to Featherstone's 22.

1.3. Inverse Kinematic Accelerations

As mentioned earlier, a solution to the inverse kinematic accelerations is required in the most general transformation between Cartesian space and joint space. For purposes of control, the joint accelerations are required for input into the inverse dynamics as well as for recent control law formulations in terms of hand acceleration. For the inverse dynamics (Johnson 1983),

$$\underline{\tau} = \mathrm{N}^{-1}(\mathrm{H}\ddot{\underline{\theta}} + \mathrm{c}(\underline{\theta}, \dot{\underline{\theta}})) \qquad (3)$$

where

$\underline{\tau}$ is the vector of joint torques,

N is a matrix which reflects the recursive structure of the dynamic equations,

H is the generalized inertia matrix, and

$\mathrm{c}(\underline{\theta}, \dot{\underline{\theta}})$ is the vector of centripetal and Coriolis torques.

Several schemes have also been proposed recently that close the feedback loop around the hand. Adopting some of the results and notation from (Johnson 1983), these schemes include:

1. Resolved acceleration (Luh, Walker, and Paul 1980b):

Method	Multiplications	Additions
6×6 matrix inversion	394	260
Gaussian elimination	213	162
Wrist partitioning	78	57

Table 2. Computational complexity of various methods of computing the inverse kinematic accelerations for a 6-dof manipulator.

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_d + \mathbf{K}_1 \dot{\mathbf{x}}_d + \mathbf{K}_2(\mathbf{x} - \mathbf{x}_d) \tag{4}$$

2. Nonlinear control (Freund 1982):

$$\ddot{\mathbf{x}} = \underline{\Lambda}\dot{\mathbf{x}}_d + \mathbf{K}_1\dot{\mathbf{x}} + \mathbf{K}_2\mathbf{x} \tag{5}$$

3. Cartesian impedance control (Hogan and Cotter 1982):

$$\ddot{\mathbf{x}} = \mathbf{M}^{-1}(\mathbf{f}_e - \mathbf{K}_1(\dot{\mathbf{x}}) - \mathbf{K}_2(\mathbf{x}) + \mathbf{K}_2(\mathbf{x}_d)) \tag{6}$$

where

$\ddot{\mathbf{x}}$ is the derived hand acceleration,

$\dot{\mathbf{x}}$ and $\mathbf{x}$ are the measured hand velocity and position (as determined from the joint angle measurements),

$\ddot{\mathbf{x}}_d$, $\dot{\mathbf{x}}_d$, and $\mathbf{x}_d$ are the desired or planned hand acceleration, velocity, and position,

$\underline{\Lambda}$ and $\mathbf{K}_1$ are velocity gain matrices,

$\mathbf{K}_2$ is a position gain matrix,

$\mathbf{f}_e$ is a vector of external forces, and

$\mathbf{M} = diag(m, m, m, I_1, I_2, I_3)$ is a matrix which relates accelerations to generalized forces through the mass m of the hand and the principal inertias I_1, I_2, I_3.

As can be seen, the exact formulations of the feedback laws (4)-(6) differ somewhat. Walker's method is the only one requiring a planned hand acceleration $\ddot{\mathbf{x}}_d$. In Hogan's method, the velocity and position gains $\mathbf{K}_1, \mathbf{K}_2$ are considered as arbitrary functions rather than just as matrices. In addition, Hogan also includes the external force $\mathbf{f}_e$ and gives the hand attributes of mass and inertia by the matrix multiplication $\mathbf{M}^{-1}$.

Once the hand acceleration $\ddot{\mathbf{x}}$ has been derived, whether by a trajectory planner or by a hand-based control law, the joint accelerations can be found through differentiation of (1).

$$\ddot{\mathbf{x}} = \mathbf{J}\ddot{\underline{\theta}} + \dot{\mathbf{J}}\dot{\underline{\theta}} \tag{7}$$

$$\ddot{\underline{\theta}} = \mathbf{J}^{-1}(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\underline{\theta}}) \tag{8}$$

As in the inverse kinematic velocity computation, solution of the inverse kinematic accelerations by matrix inversion (8) is very costly (Table 2). Again, since the inverse Jacobian itself is not of interest, the joint accelerations are better found by solving (7) through Gaussian elimination (Table 2). Until now, there had been no better method than Gaussian elimination, but it can be seen that the wrist-partitioning algorithm developed later is substantialy more efficient.

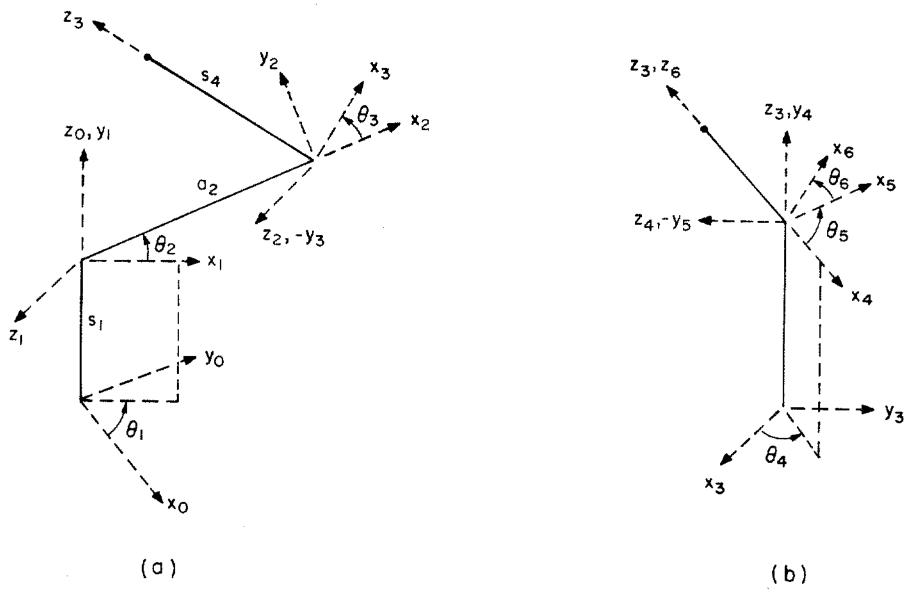

Figure 2. (a) First three joints of the rotary manipulator. (b) Spherical wrist of the rotary manipulator.

The proposals of Walker, Freund, and Hogan, however, seem to require the matrix inversion method. The hand acceleration is incorporated into the inverse dynamics computation (3) by substituting for the joint accelerations (8).

$$\underline{\tau} = N^{-1}(HJ^{-1}(\ddot{x} - \dot{J}\underline{\dot{\theta}}) + c(\underline{\theta}, \underline{\dot{\theta}})) \tag{9}$$

If they had intended Gaussian elimination, then the inverse kinematics and dynamics should have been expressed in the forms (3) and (7). Because of the apparent difficulty of solving the inverse kinematic accelerations, Khatib (1980) proposed a hand-based feedback law which involves resolving hand forces through the inverse Jacobian rather than resolving the hand accelerations.

2. Rotary Manipulator

We now proceed to derive an efficient formulation of the inverse kinematic accelerations. To demonstrate the method, we use a rotary manipulator without offsets. Other manipulator types are readily adapted to Featherstone's method, although the details primarily in step 2 and slightly in step 3 must be reworked; the other computations remain the same if the manipulator has a spherical wrist.

The first three joints are shown in Figure 2a and the last three in Figure 2b. The joint axes for this manipulator are derived from the Denavit-Hartenberg (1955) specification. The rotation axis $\mathbf{z}_i$ corresponds to joint angle θ_{i+1} between links i and $i+1$. The internal coordinate system for link i is completed by defining $\mathbf{x}_i$ from the cross product $\mathbf{z}_{i-1} \times \mathbf{z}_i$, which also locates the origin, and $\mathbf{y}_i = \mathbf{z}_i \times \mathbf{x}_i$. Neighboring coordinate systems are related by three parameters:

a_i is the distance between $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ measured along $\mathbf{x}_i$,

s_i is the distance between $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ measured along $\mathbf{z}_{i-1}$, and

Joint i	a_i	s_i	α_i
1	0	s_1	$\pi/2$
2	a_2	0	0
3	0	0	$-\pi/2$
4	0	s_4	$\pi/2$
5	0	0	$-\pi/2$
6	0	0	0

Table 3. Denavit-Hartenberg parameters for the rotary manipulator of Figure 2.

α_i is the angle between $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ measured in a righthand sense about $\mathbf{x}_i$.

The values for the three parameters above for the rotary arm of Figure 2 are given in Table 3.

The directions for $\mathbf{z}_i$ and $\mathbf{x}_i$ are chosen so that when $\theta_i = 0$, then $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ are parallel and pointing in the same direction. For the rotary manipulator, this desideratum is straightforwardly satisfied for every coordinate sytem except 3 and 5. For coordinate system 3, we desire that the rotation axis $\mathbf{z}_3$ point at the wrist. This means that $\mathbf{x}_3$ must point in the opposite direction from $\mathbf{z}_2 \times \mathbf{z}_3$, which is reflected by $\alpha_3 = -\pi/2$, and that the zero position for the elbow joint is at a right angle pointing upward (Figure 2a). Similarly, because it is desired to point $\mathbf{z}_5$ towards the tip of the hand, $\mathbf{x}_4$ must point in the opposite direction from $\mathbf{z}_4 \times \mathbf{z}_5$, and $\alpha_5 = -\pi/2$ (Figure 2b).

The rotation matrix $\mathbf{A}_i$ which transforms points expressed in link i coordinates to link $i-1$ coordinates is:

$$\mathbf{A}_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i \\ 0 & s\alpha_i & c\alpha_i \end{bmatrix} \tag{10}$$

where we have used the abbreviations $s\theta = \sin\theta$ and $c\theta = \cos\theta$. For convenience, we list below the six joint transformation matrices.

$$\mathbf{A}_1 = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 \\ s\theta_1 & 0 & -c\theta_1 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 \\ s\theta_2 & c\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{A}_3 = \begin{bmatrix} c\theta_3 & 0 & -s\theta_3 \\ s\theta_3 & 0 & c\theta_3 \\ 0 & -1 & 0 \end{bmatrix}$$
$$\mathbf{A}_4 = \begin{bmatrix} c\theta_4 & 0 & s\theta_4 \\ s\theta_4 & 0 & -c\theta_4 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{A}_5 = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 \\ s\theta_5 & 0 & c\theta_5 \\ 0 & -1 & 0 \end{bmatrix} \quad \mathbf{A}_6 = \begin{bmatrix} c\theta_6 & -s\theta_6 & 0 \\ s\theta_6 & c\theta_6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

With these transformation matrices, points on any one link may be referred to the coordinate system of any other link. We will denote with a left superscript which coordinate system a vector is referred to. Then ${}^i\mathbf{v}$ is a vector referred to the ith coordinate system, and ${}^{i-1}\mathbf{v} = \mathbf{A}_i\, {}^i\mathbf{v}$ is this same vector referred to the $i-1$st coordinate system. These transformation matrices can be chained together to refer non-adjacent links i and j (where $j > i$) by defining ${}^i\mathbf{W}_j = \mathbf{A}_{i+1}\mathbf{A}_{i+2}\cdots\mathbf{A}_j$, where ${}^i\mathbf{v} = {}^i\mathbf{W}_j\, {}^j\mathbf{v}$. By convention, the left superscript is omitted when referring to the base coordinate 0, e.g., $\mathbf{v} = {}^0\mathbf{v}$.

Define the following vectors, which will be useful later:

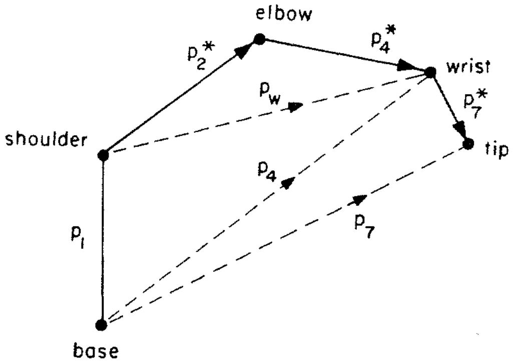

Figure 3. Vector definitions between various coordinate origins of the rotary manipulator.

$\mathbf{p}_i$ is the vector from coordinate origin 0 to coordinate origin i, and

$\mathbf{p}_i^*$ is the vector from coordinate origin $i-1$ to coordinate origin i.

Due to the coincidence of coordinate origins 2-3 and of 4-6, many of these vectors are not distinct. Thus $\mathbf{p}_{i-1} = \mathbf{p}_i$ and $\mathbf{p}_i^* = 0$ for $i = 3, 5, 6$. Furthermore, $\mathbf{p}_1 = \mathbf{p}_1^*$.

3. Specification of the Cartesian Trajectory

In applying Featherstone's method, it is necessary to have available the angular velocity and acceleration vectors $\underline{\omega}_6, \dot{\underline{\omega}}_6$ of the hand as well as a time history of the (x, y, z) Cartesian position of some point on the hand. Yet some of the more common Cartesian trajectory planning algorithms, such as those built around straight-line, constant-velocity segments (Paul 1981, Taylor 1979), yield instead of the angular velocity and acceleration vectors a time-varying hand orientation matrix $\mathbf{W}_6$. In such a case, the angular velocity and acceleration of the hand can be derived from the orientation matrix.

If $\mathbf{p}_7$ is a vector from the base to the point on the tip of the hand used in trajectory planning, and $\mathbf{p}_7^*$ is the internal vector from coordinate origin 6 to this same tip point (Figure 3), then for a spherical-wrist arm

$$\mathbf{p}_7 = \mathbf{p}_4 + \mathbf{W}_6 \, {}^6\mathbf{p}_7^* \tag{12}$$

$$\dot{\mathbf{p}}_7 = \dot{\mathbf{p}}_4 + \dot{\mathbf{W}}_6 \, {}^6\mathbf{p}_7^* \tag{13}$$

$$\ddot{\mathbf{p}}_7 = \ddot{\mathbf{p}}_4 + \ddot{\mathbf{W}}_6 \, {}^6\mathbf{p}_7^* \tag{14}$$

where the first two time derivatives of the rotation matrix $\mathbf{W}_6$ must be available. In the vectorial representation,

$$\dot{\mathbf{p}}_7 = \dot{\mathbf{p}}_4 + \underline{\omega}_6 \times \mathbf{W}_6 \, {}^6\mathbf{p}_7^* \tag{15}$$

$$\ddot{\mathbf{p}}_7 = \ddot{\mathbf{p}}_4 + \dot{\underline{\omega}}_6 \times \mathbf{W}_6 \, {}^6\mathbf{p}_7^* + \underline{\omega}_6 \times (\underline{\omega}_6 \times \mathbf{W}_6 \, {}^6\mathbf{p}_7^*) \tag{16}$$

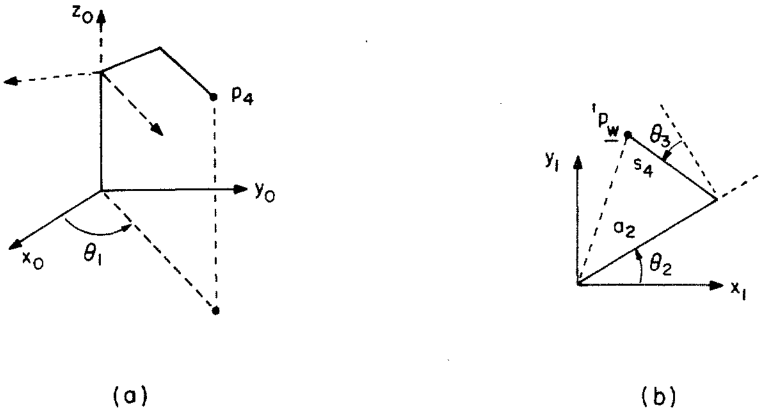

Figure 4. (a) Projection of the wrist point onto the $x_0 - y_0$ plane to find θ_1. (b) Reduction to a planar two-link manipulator for θ_2 and θ_3 by referring positions to link 1 coordinates.

Equating the corresponding elements,

$$\tilde{\underline{\omega}}_6 = \dot{\mathbf{W}}_6 \mathbf{W}_6^T \tag{17}$$

$$\tilde{\dot{\underline{\omega}}}_6 = \ddot{\mathbf{W}}_6 \mathbf{W}_6^T + \dot{\mathbf{W}}_6 \dot{\mathbf{W}}_6^T \tag{18}$$

where $\tilde{\underline{\omega}}_6$ and $\tilde{\dot{\underline{\omega}}}_6$ are the matrix representations for the cross product by $\underline{\omega}_6$ and $\dot{\underline{\omega}}_6$ respectively; that is to say, $\tilde{\underline{\omega}}_6 \mathbf{v} = \underline{\omega}_6 \times \mathbf{v}$ for example.

4. Inverse Kinematic Positions

In applying Featherstone's method, it is necessary to have solved for the inverse kinematic positions before finding the inverse kinematic velocities, and to have solved for the inverse kinematic velocities before finding the inverse kinematic accelerations. For completeness of presentation and to show how intermediate results from one inverse kinematic level are used in the next, we begin with a rederivation of the inverse kinematic positions.

Step 1: Find the wrist position. We presume that the orientation matrix $\mathbf{W}_6$ for the hand and the position $\mathbf{p}_7$ of the tip of the hand have been specified. Since $^6\mathbf{p}_7^*$, the internal vector in link 6 which extends from the coordinate 6 origin to the hand tip, is known, then the position of the wrist $\mathbf{p}_4$ is given by

$$\mathbf{p}_4 = \mathbf{p}_7 - \mathbf{W}_6 \, {}^6\mathbf{p}_7^* \tag{19}$$

Step 2: Find the first three joint angles. Joint 1 is directly found from $\mathbf{p}_4$, since joints 2 and 3 act in a plane that does not change the projection of the wrist onto the x_0, y_0 plane (Figure 4a):

$$\theta_1 = \tan^{-1}\left(\frac{p_{4y}}{p_{4x}}\right) \tag{20}$$

where the $\tan^{-1}$ function corresponds to the FORTRAN ATAN2 function, so that the above division is not actually carried out. We follow Paul (1981) in this practice because of the superior numerical conditioning with this inverse trigonometric function compared to others. A degeneracy occurs when the wrist lies on the $\mathbf{z}_0$ axis, where $p_{4x} = p_{4y} = 0$ and θ_1 can assume any value. Since $s\theta_1$ and $c\theta_1$ are required below, they are best found with one additional transcendental function call. Defining $r^2 = p_{4x}^2 + p_{4y}^2$,

$$c\theta_1 = \frac{p_{4x}}{r}, \quad s\theta_1 = \frac{p_{4y}}{r} \tag{21}$$

The next two angles are easily found if $\mathbf{p}_4$ is expressed in joint 1 coordinates, which reduces the problem to a planar two-link problem. Define

$$\mathbf{p}_w = \mathbf{p}_4 - \mathbf{p}_1 = \mathbf{p}_4 - s_1\mathbf{z}_0 \tag{22}$$

When expressed in link 1 coordinates,

$$^1\mathbf{p}_w = \mathbf{A}_1^T\mathbf{p}_w = \begin{bmatrix} r \\ p_{wz} \\ 0 \end{bmatrix} \tag{23}$$

where $^1p_{wz} = 0$ because the wrist point $\mathbf{p}_w$ lies in the x_1/y_1 plane. By the cosine rule (Figure 4b),

$$s\theta_3 = \frac{a_2^2 + s_4^2 - (r^2 + p_{wz}^2)}{2a_2s_4} \tag{24}$$

$c\theta_3$ is found next.

$$c\theta_3 = \pm\sqrt{1 - (s\theta_3)^2} \tag{25}$$

$$\theta_3 = \tan^{-1}\left(\frac{s\theta_3}{c\theta_3}\right) \tag{26}$$

Depending on which quadrant is chosen for θ_3, one obtains an elbow up or elbow down solution.

The angle θ_2 is found by expressing $^1\mathbf{p}_w$ in terms of the joint axis vectors.

$$\mathbf{p}_w = a_2\mathbf{x}_2 + s_4\mathbf{z}_3 \tag{27}$$

$$^1\mathbf{p}_w = \begin{bmatrix} a_2c\theta_2 - s_4s(\theta_2+\theta_3) \\ a_2s\theta_2 + s_4c(\theta_2+\theta_3) \\ 0 \end{bmatrix}$$

Solving by simultaneous equations,

$$s\theta_2 = \frac{{}^1p_{wy}(a_2 - (s_4 s\theta_3)) - {}^1p_{wx}(s_4 c\theta_3)}{a_2^2 + s_4^2 - 2a_2(s_4 s\theta_3)} \tag{28}$$

$$c\theta_2 = \frac{{}^1p_{wx} + s\theta_2(s_4 c\theta_3)}{a_2 - (s_4 s\theta_3)} \tag{29}$$

$$\theta_2 = \tan^{-1}\left(\frac{s\theta_2}{c\theta_2}\right) \tag{30}$$

where the products in the parentheses are computed only once.

Step 3: Find the hand orientation relative to the forearm. The forearm orientation is given by $W_3 = A_1A_2A_3$, so that the hand orientation relative to the forearm, ${}^3W_6 = A_4A_5A_6$, is given by

$${}^3\mathbf{W}_6 = \mathbf{W}_3^T\mathbf{W}_6 \tag{31}$$

Because they yield intermediate results which are useful later, we present the partial matrix multiplications A_1A_2 and $(A_1A_2)A_3$.

$$A_1A_2 = \begin{bmatrix} c\theta_1 c\theta_2 & -c\theta_1 s\theta_2 & s\theta_1 \\ s\theta_1 c\theta_2 & -s\theta_1 s\theta_2 & -c\theta_1 \\ s\theta_2 & c\theta_2 & 0 \end{bmatrix} \tag{32}$$

$$(A_1A_2)A_3 = \begin{bmatrix} c\theta_3(c\theta_1 c\theta_2) - s\theta_3(c\theta_1 s\theta_2) & -s\theta_1 & -s\theta_3(c\theta_1 c\theta_2) - c\theta_3(c\theta_1 s\theta_2) \\ c\theta_3(s\theta_1 c\theta_2) - s\theta_3(s\theta_1 s\theta_2) & c\theta_1 & -s\theta_3(s\theta_1 c\theta_2) - c\theta_3(s\theta_1 s\theta_2) \\ s(\theta_2 + \theta_3) & 0 & c(\theta_2 + \theta_3) \end{bmatrix} \tag{33}$$

where the sine/cosine products in the parentheses are computed only once and the 31 and 33 elements in (33) are computed from the expansion. Finally, we presume ${}^3W_6$ has been computed from a knowledge of W_6 and $\theta_1, \theta_2, \theta_3$, and that its elements are w_{ij}.

Step 4: Find the last three joint angles. Joint angles θ_4 and θ_6 can be found from the elements of ${}^3W_6$, which are identified by some matrix manipulations.

$$A_4A_5 = \begin{bmatrix} c\theta_4 c\theta_5 & -s\theta_4 & -c\theta_4 s\theta_5 \\ s\theta_4 c\theta_5 & c\theta_4 & -s\theta_4 s\theta_5 \\ s\theta_5 & 0 & c\theta_5 \end{bmatrix} \tag{34}$$

$${}^3W_6A_6^T = \begin{bmatrix} w_{11}c\theta_6 - w_{12}s\theta_6 & w_{11}s\theta_6 + w_{12}c\theta_6 & w_{13} \\ w_{21}c\theta_6 - w_{22}s\theta_6 & w_{21}s\theta_6 + w_{22}c\theta_6 & w_{23} \\ w_{31}c\theta_6 - w_{32}s\theta_6 & w_{31}s\theta_6 + w_{32}c\theta_6 & w_{33} \end{bmatrix} \tag{35}$$

Since $A_4A_5 = {}^3W_6A_6^T$, we find by selecting elements 13 and 23 that

$$\theta_4 = \tan^{-1}\left(\frac{-w_{23}}{-w_{13}}\right) \tag{36}$$

Because of the multiplication factor $s\theta_5$, there are two possible solutions to this equation; the quadrant must be chosen by some other criterion such as continuity. Note the singularity when $\sin\theta_5 = 0$, causing the z_3 and z_5 axes to line up. There is no numerically-sound shortcut to finding $s\theta_4$ and $c\theta_4$, required below, so that 2 more transcendental function calls are needed.

Similarly, θ_5 and θ_6 can be found by equating A_5A_6 and $A_4^T\,{}^3W_6$:

$$A_5A_6 = \begin{bmatrix} c\theta_5c\theta_6 & -c\theta_5s\theta_6 & -s\theta_5 \\ s\theta_5c\theta_6 & -s\theta_5s\theta_6 & c\theta_5 \\ -s\theta_6 & -c\theta_6 & 0 \end{bmatrix} \tag{37}$$

$$A_4^T\,{}^3W_6 = \begin{bmatrix} w_{11}c\theta_4 + w_{21}s\theta_4 & w_{12}c\theta_4 + w_{22}s\theta_4 & w_{13}c\theta_4 + w_{23}s\theta_4 \\ w_{31} & w_{32} & w_{33} \\ w_{11}s\theta_4 - w_{21}c\theta_4 & w_{12}s\theta_4 - w_{22}c\theta_4 & w_{13}s\theta_4 - w_{23}c\theta_4 \end{bmatrix} \tag{38}$$

θ_5 can be found from the 13 and 23 elements of these last matrices and the value of θ_4.

$$s\theta_5 = -\,w_{13}c\theta_4 - w_{23}s\theta_4 \tag{39}$$

$$c\theta_5 = w_{33} \tag{40}$$

$$\theta_5 = \tan^{-1}\left(\frac{s\theta_5}{c\theta_5}\right) \tag{41}$$

By selecting the 31 and 32 elements of these matrices,

$$s\theta_6 = -\,w_{11}s\theta_4 + w_{21}c\theta_4 \tag{42}$$

$$c\theta_6 = -\,w_{12}s\theta_4 + w_{22}c\theta_4 \tag{43}$$

$$\theta_6 = \tan^{-1}\left(\frac{s\theta_6}{c\theta_6}\right) \tag{44}$$

This particular method of computing θ_6 has been chosen because it yields $s\theta_6$ and $c\theta_6$, and yields θ_6 unambiguously.

4.1. Computational complexity

The total computational complexity for steps 1-4 comes to 64 multiplications, 38 additions, and 10 transcendental function calls, broken down as follows:

Step 1: 9 multiples and 9 additions.

Step 2: 15 multiplications, 9 additions, and 5 transcendental function calls. In deriving the additions and multiplications, we note that for θ_3 the subexpressions $a_2^2 + s_4^2$ and $2a_2s_4$ can be precomputed.

Step 3: 34 multiplications and 17 additions. Note that only seven elements of ${}^3W_6$ are required, namely $w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23}, w_{33}$.

Step 4: 6 multiplications, 3 additions, and 5 transcendental function calls.

5. Inverse Kinematic Velocities

Step 1: Find the wrist linear velocity. The wrist and hand tip linear velocities are related by

$$\dot{\mathbf{p}}_4 = \dot{\mathbf{p}}_7 - \underline{\omega}_6 \times \mathbf{p}_7^* \tag{45}$$

Step 2: Find the first three joint velocities. The wrist position is given by

$$\mathbf{p}_4 = s_1\mathbf{z}_0 + a_2\mathbf{x}_2 + s_4\mathbf{z}_3 \tag{46}$$

Differentiating,

$$\dot{\mathbf{p}}_4 = \underline{\omega}_2 \times a_2\mathbf{x}_2 + \underline{\omega}_3 \times s_4\mathbf{z}_3 \tag{47}$$

When evaluated in link 2 coordinates (Appendix 1),

$${}^2\dot{\mathbf{p}}_4 = \begin{bmatrix} -s_4 c\theta_3(\dot\theta_2 + \dot\theta_3) \\ a_2\dot\theta_2 - s_4 s\theta_3(\dot\theta_2 + \dot\theta_3) \\ -\dot\theta_1\, {}^1p_{wx} \end{bmatrix} \tag{48}$$

where ${}^2\dot{\mathbf{p}}_4 = (\mathrm{A}_1\mathrm{A}_2)^T\dot{\mathbf{p}}_4$. Solving,

$$\begin{aligned} \dot\theta_1 &= -\frac{{}^2\dot p_{4z}}{{}^1p_{wx}} \\ \dot\theta_2 &= \frac{{}^2\dot p_{4y} c\theta_3 - {}^2\dot p_{4x} s\theta_3}{a_2 c\theta_3} \\ \dot\theta_3 &= \frac{{}^2\dot p_{4x} - \dot\theta_2(s_4 c\theta_3)}{s_4 c\theta_3} \end{aligned} \tag{49}$$

Step 3: Find the angular velocity of the hand relative to the forearm. The hand angular velocity relative to the forearm, $\underline{\omega}_h$, is given by

$$\underline{\omega}_h = \underline{\omega}_6 - \underline{\omega}_3 \tag{50}$$

This is best evaluated in joint 4 coordinates, accomplished as follows:

$${}^3\underline{\omega}_3 = \begin{bmatrix} \dot\theta_1 s(\theta_2 + \theta_3) \\ -(\dot\theta_2 + \dot\theta_3) \\ \dot\theta_1 c(\theta_2 + \theta_3) \end{bmatrix} \tag{51}$$

$${}^3\underline{\omega}_6 = (\mathrm{A}_1\mathrm{A}_2\mathrm{A}_3)^T\underline{\omega}_6 \tag{52}$$

$${}^3\underline{\omega}_h = {}^3\underline{\omega}_6 - {}^3\underline{\omega}_3 \tag{53}$$

$${}^4\underline{\omega}_h = \mathrm{A}_4^T\, {}^3\underline{\omega}_h = \begin{bmatrix} {}^3\omega_{hx} c\theta_4 + {}^3\omega_{hy} s\theta_4 \\ {}^3\omega_{hz} \\ {}^3\omega_{hx} s\theta_4 - {}^3\omega_{hy} c\theta_4 \end{bmatrix} \tag{54}$$

Step 4: Find the last three joint velocities. The hand angular velocity relative to the forearm is also given by

$$\underline{\omega}_h = \mathbf{z}_3\dot{\theta}_4 + \mathbf{z}_4\dot{\theta}_5 + \mathbf{z}_5\dot{\theta}_6 \tag{55}$$

Expressed in joint 4 coordinates, we find that

$${}^4\mathbf{z}_3 = {}^4\mathbf{y}_4 \tag{56}$$

$${}^4\mathbf{z}_5 = \begin{bmatrix} -s\theta_5 \\ c\theta_5 \\ 0 \end{bmatrix} \tag{57}$$

$${}^4\underline{\omega}_h = \begin{bmatrix} -\dot{\theta}_6 s\theta_5 \\ \dot{\theta}_4 + \dot{\theta}_6 c\theta_5 \\ \dot{\theta}_5 \end{bmatrix} \tag{58}$$

Solving,

$$\dot{\theta}_5 = {}^4\omega_{hz} \tag{59}$$

$$\dot{\theta}_6 = -\frac{{}^4\omega_{hx}}{s\theta_5} \tag{60}$$

$$\dot{\theta}_4 = {}^4\omega_{hy} - \dot{\theta}_6 c\theta_5 \tag{61}$$

5.1. Computation Complexity

The total for steps 1-4 is 37 multiplications and 25 additions. In arriving at these numbers, we presume to have the results of the inverse kinematic position computation available. The breakdown is as follows:

Step 1: 6 multiplications and 6 additions.

Step 2: 15 multiplications and 7 additions.

Step 3: 14 multiplications and 11 additions.

Step 4: 2 multiplications and 1 addition.

6. Inverse Kinematic Accelerations

Step 1: Find the wrist linear acceleration. This is readily found as

$$\ddot{\mathbf{p}}_4 = \ddot{\mathbf{p}}_7 - \dot{\underline{\omega}}_6 \times \mathbf{p}_7^* - \underline{\omega}_6 \times (\underline{\omega}_6 \times \mathbf{p}_7^*) \tag{62}$$

Step 2: Find the first three joint accelerations. Differentiating (47), and noting that $\dot{\mathbf{p}}_w = \dot{\mathbf{p}}_4$,

$$\ddot{\mathbf{p}}_4 = \dot{\underline{\omega}}_2 \times \mathbf{p}_w + \underline{\omega}_2 \times \dot{\mathbf{p}}_4 - s_4\ddot{\theta}_3\mathbf{x}_3 - s_4\dot{\theta}_3\underline{\omega}_3 \times \mathbf{x}_3 \tag{63}$$

Expressed in joint 2 coordinates, this equation evaluates to (Appendix II)

$$\begin{bmatrix} -s_4c\theta_3(\ddot{\theta}_2+\ddot{\theta}_3) \\ a_2\ddot{\theta}_2 - s_4s\theta_3(\ddot{\theta}_2+\ddot{\theta}_3) \\ -{}^1p_{wx}\ddot{\theta}_1 \end{bmatrix} = {}^2\ddot{\mathbf{p}}_4 - {}^2\underline{\omega}_2 \times {}^2\dot{\mathbf{p}}_4 + \begin{bmatrix} s_4\dot{\theta}_3 s\theta_3\, {}^3\omega_{3y} \\ -s_4\dot{\theta}_3 c\theta_3\, {}^3\omega_{3y} \\ -s_4\dot{\theta}_3\, {}^3\omega_{3z} - {}^1p_{wy}\dot{\theta}_1\dot{\theta}_2 \end{bmatrix} \tag{64}$$

Defining the right side as ${}^2\ddot{\mathbf{u}}_4$, the joint accelerations can now be found.

$$\ddot{\theta}_1 = -\frac{{}^2\ddot{u}_{4z}}{{}^1p_{wx}} \tag{65}$$

$$\ddot{\theta}_2 = \frac{{}^2\ddot{u}_{4y}c\theta_3 - {}^2\ddot{u}_{4x}s\theta_3}{a_2c\theta_3} \tag{66}$$

$$\ddot{\theta}_3 = -\frac{{}^2\ddot{u}_{4x} + s_4c\theta_3\ddot{\theta}_2}{s_4c\theta_3} \tag{67}$$

Step 3: Find the hand angular acceleration relative to the forearm. By differentiating (50), we find that

$$\dot{\underline{\omega}}_h = \dot{\underline{\omega}}_6 - \dot{\underline{\omega}}_3 - \underline{\omega}_3 \times \underline{\omega}_h \tag{68}$$

The angular acceleration $\dot{\underline{\omega}}_3$ is found in link 3 coordinates from the previous results.

$${}^1\dot{\underline{\omega}}_1 = {}^1\mathbf{y}_1\ddot{\theta}_1 \tag{69}$$

$${}^1\dot{\underline{\omega}}_2 = \begin{bmatrix} \dot{\theta}_1\dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$

$${}^2\dot{\underline{\omega}}_2 = \mathbf{A}_2^T\, {}^1\dot{\underline{\omega}}_2 \tag{70}$$

$${}^2\dot{\underline{\omega}}_3 = {}^2\dot{\underline{\omega}}_2 + {}^2\mathbf{z}_2\ddot{\theta}_3 + \begin{bmatrix} \dot{\theta}_1\dot{\theta}_3c\theta_2 \\ -\dot{\theta}_1\dot{\theta}_3s\theta_2 \\ 0 \end{bmatrix} \tag{71}$$

$${}^3\dot{\underline{\omega}}_3 = \mathbf{A}_3^T\, {}^2\dot{\underline{\omega}}_3 \tag{72}$$

Evaluating in link 4 coordinates,

$${}^3\dot{\underline{\omega}}_6 = (\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3)^T\dot{\underline{\omega}}_6 \tag{73}$$

$${}^4\dot{\underline{\omega}}_h = \mathbf{A}_4^T({}^3\dot{\underline{\omega}}_6 - {}^3\dot{\underline{\omega}}_3 - {}^3\underline{\omega}_3 \times {}^3\underline{\omega}_h) \tag{74}$$

Kinematic Parameter	Multiplications	Additions
Joint Angles	64	38
Joint Velocities	37	25
Joint Accelerations	78	57
Combined Total	179	120

Table 4. Computational complexity for evaluation of the inverse kinematic positions, velocities, and accelerations.

Step 4: Find the last three joint accelerations. Written in terms of the last three angular accelerations, the relative hand acceleration is

$$\dot{\underline{\omega}}_h = \mathbf{z}_3\ddot{\theta}_4 + \mathbf{z}_4\ddot{\theta}_5 + \mathbf{z}_5\ddot{\theta}_6 + \mathbf{z}_3\dot{\theta}_4 \times \mathbf{z}_4\dot{\theta}_5 + (\mathbf{z}_3\dot{\theta}_4 + \mathbf{z}_4\dot{\theta}_5) \times \mathbf{z}_5\dot{\theta}_6 \tag{75}$$

When evaluated in joint 4 coordinates,

$$^4\dot{\underline{\omega}}_h = \begin{bmatrix} \dot{\theta}_4\dot{\theta}_5 - \dot{\theta}_5\dot{\theta}_6 c\theta_5 \\ -\dot{\theta}_5\dot{\theta}_6 s\theta_5 \\ \dot{\theta}_4\dot{\theta}_6 s\theta_5 \end{bmatrix} + \begin{bmatrix} -\ddot{\theta}_6 s\theta_5 \\ \ddot{\theta}_6 c\theta_5 + \ddot{\theta}_4 \\ \ddot{\theta}_5 \end{bmatrix} \tag{76}$$

The joint accelerations can now be found.

$$\ddot{\theta}_6 = -\frac{^4\dot{\omega}_{hx} - \dot{\theta}_4\dot{\theta}_5 + \dot{\theta}_5\dot{\theta}_6 c\theta_5}{s\theta_5} \tag{77}$$

$$\ddot{\theta}_5 = {}^4\dot{\omega}_{hz} - \dot{\theta}_4\dot{\theta}_6 s\theta_5 \tag{78}$$

$$\ddot{\theta}_4 = {}^4\dot{\omega}_{hy} + \dot{\theta}_5\dot{\theta}_6 s\theta_5 - \ddot{\theta}_6 c\theta_5 \tag{79}$$

6.1. Computational Complexity

The total is 78 multiplications and 57 additions, using the results of the previous inverse calculations. This is constituted as follows:

Step 1: 12 multiplications and 12 additions. Here $\underline{\omega}_6 \times \mathbf{p}_7^*$ is already known.

Step 2: 29 multiplications and 17 additions. It is required to compute $^2\underline{\omega}_2$ here.

Step 3: 30 multiplications and 23 additions.

Step 4: 8 multiplications and 5 additions.

Table 4 summarizes the results for computation of the inverse kinematic positions, velocities, and accelerations.

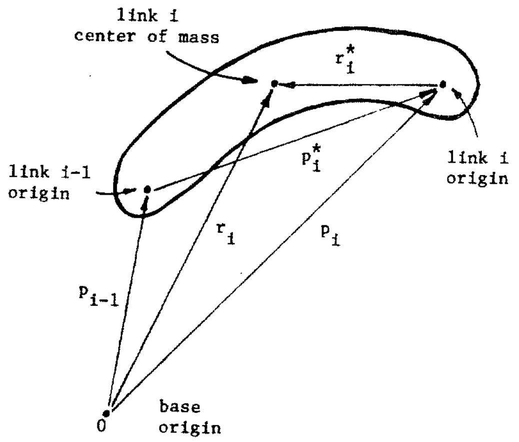

Figure 5. An illustration of vectors defined in the text.

7. Dynamics Computation

In the recursive Newton-Euler dynamics formulation, which is the most efficient one (Hollerbach 1980), a kinematics computation precedes application of the Newton-Euler equations. Because of the simplified kinematics of spherical wrist robots, the kinematic portion of the dynamics computation is simplified as well. In addition, the inverse kinematic computations produce some of the same quantities as does the inverse dynamics computation. Therefore a combined inverse kinematic - inverse dynamics computation would save some operations in the latter computation.

7.1. Recursive Newton-Euler Dynamics

The recursive Newton-Euler computation of the dynamics proceeds by a two-step recursive procedure (Luh, Walker, and Paul 1980a). First, the angular velocities and accelerations of each link are computed along with the linear acceleration of each joint.

$$ {}^i\underline{\omega}_i = \mathbf{A}_i^T({}^{i-1}\underline{\omega}_{i-1} + {}^{i-1}\,\mathbf{z}_{i-1}\dot{\theta}_i) \tag{80}$$

$$ {}^i\underline{\dot{\omega}}_i = \mathbf{A}_i^T({}^{i-1}\underline{\dot{\omega}}_{i-1} + {}^{i-1}\mathbf{z}_{i-1}\ddot{\theta}_i + {}^{i-1}\underline{\omega}_{i-1} \times {}^{i-1}\mathbf{z}_{i-1}\dot{\theta}_i) \tag{81}$$

$$ {}^i\ddot{\mathbf{p}}_i = \mathbf{A}_i^T\,{}^{i-1}\ddot{\mathbf{p}}_{i-1} + {}^i\underline{\dot{\omega}}_i \times {}^i\mathbf{p}_i^* + {}^i\underline{\omega}_i \times ({}^i\underline{\omega}_i \times {}^i\mathbf{p}_i^*) \tag{82}$$

where these particular equations presume a rotary joint manipulator only. From these results, the Newton-Euler equations may be applied to find the net force and torque acting at each link.

$$ {}^i\ddot{\mathbf{r}}_i = {}^i\ddot{\mathbf{p}}_i + {}^i\underline{\dot{\omega}}_i \times {}^i\mathbf{r}_i^* + {}^i\underline{\omega}_i \times ({}^i\underline{\omega}_i \times {}^i\mathbf{r}_i^*) \tag{83}$$

$$ {}^i\mathbf{f}_i = m_i\,{}^i\ddot{\mathbf{r}}_i \tag{84}$$

$$ {}^i\mathbf{n}_i = {}^i\mathbf{I}_i\,{}^i\underline{\dot{\omega}}_i + {}^i\underline{\omega}_i \times ({}^i\mathbf{I}_i\,{}^i\underline{\omega}_i) \tag{85}$$

where previously undefined terms are (Figure 5):

- ${}^i\mathbf{r}_i^*$ is a vector from coordinate system i to the center of mass of link i,
- ${}^i\ddot{\mathbf{r}}_i$ is the acceleration of the center of mass of link i,
- ${}^i\mathbf{f}_i$ is the net force on link i,
- ${}^i\mathbf{n}_i$ is the net torque on link i, and
- ${}^i\mathbf{I}_i$ is the inertia tensor of link i about its center of mass.

Secondly, the forces and torques are propagated from the tip to the base.

$$ {}^i\mathbf{f}_{i-1,i} = {}^i\mathbf{f}_i + \mathbf{A}_{i+1}\, {}^{i+1}\mathbf{f}_{i,i+1} \tag{86} $$

$$ {}^i\mathbf{n}_{i-1,i} = {}^i\mathbf{n}_i + \mathbf{A}_{i+1}\, {}^{i+1}\mathbf{n}_{i,i+1} + ({}^i\mathbf{p}_i^* + {}^i\mathbf{r}_i^*) \times {}^i\mathbf{f}_i + {}^i\mathbf{p}_i^* \times {}^i\mathbf{f}_{i,i+1} \tag{87} $$

$$ \tau_i = {}^{i-1}\mathbf{z}_{i-1} \cdot {}^{i-1}\mathbf{n}_{i-1,i} \tag{88} $$

where

- ${}^i\mathbf{f}_{i-1,i}$ is the force exerted on link i by link $i-1$,
- ${}^i\mathbf{n}_{i-1,i}$ is the moment exerted on link i by link $i-1$, and
- τ_i is the input torque at joint i.

Evaluation of these dynamics for a general 6-rotary-joint manipulator requires 688 multiplications and 558 additions (a slightly lower figure than given in (Hollerbach 1980)).

7.2. Dynamics Particularized to a Spherical-Wrist Manipulator

By taking advantage of the simple kinematic structure of the spherical-wrist, rotary manipulator, the dynamic complexity is reduced to 448 multiplications and 361 additions. The breakdown for the savings is as follows.

$\mathbf{A}_i$: 116 multiplications and 87 additions are saved. A matrix multiply now takes 4 multiplications and 2 additions instead of 8 multiplications and 5 additions because each α_i is either 0 or $\pm\pi/2$. There are 29 applications of the link transformation matrices $\mathbf{A}_i$ for the six-joint manipulator.

$\ddot{\mathbf{p}}_i$: an additional 92 multiplications and 78 additions are saved. This comes about because (i) $\ddot{\mathbf{p}}_i = \ddot{\mathbf{p}}_{i-1}$ for $i = 1, 3, 5, 6$, although the term ${}^i\ddot{\mathbf{p}}_i = \mathbf{A}_i^T\, {}^{i-1}\ddot{\mathbf{p}}_{i-1}$ must still be evaluated; and (ii) $\mathbf{p}_2^* = \mathbf{z}_1 a_2$, $\mathbf{p}_4^* = \mathbf{z}_3 s_4$ simplify some of the cross products.

$\mathbf{n}_{i-1,i}$: an additional 32 multiplications and 32 additions saved. As above, this comes about from the simple $\mathbf{p}_i^*$ vectors when evaluating ${}^i\mathbf{p}_i^* \times {}^i\mathbf{f}_{i,i+1}$.

$\underline{\omega}_0, \dot{\underline{\omega}}_0$: If the assumption of a non-spinning base is made, i.e., $\underline{\omega}_0 = \dot{\underline{\omega}}_0 = \mathbf{0}$, then an additional 40 multiplications and 37 additions is saved in evaluating ${}^1\underline{\omega}_1$, ${}^2\underline{\omega}_2$, ${}^1\dot{\underline{\omega}}_1$, ${}^2\dot{\underline{\omega}}_2$, ${}^1\ddot{\mathbf{r}}_1$, and ${}^1\mathbf{n}_1$.

Counting the non-spinning base assumption, the dynamic complexity is reduced to 408 multiplications and 324 additions, a savings of 280 multiplications and 234 additions. It should be noted that if the $\mathbf{A}_i$ matrices are not provided to the dynamic computation, then an additional 4 multiplications are required to form each matrix in the general case. This would be unnecessary for the particular manipulator considered here.

7.3. Inverse Kinematics Considered with Inverse Dynamics

In computing the inverse kinematics, there is an additional modest savings in the inverse dynamics of 40 multiplications and 28 additions, bringing the total dynamics complexity down to 368 multiplications and 296 additions. This results from the precomputation of the following quantities.

${}^1\underline{\omega}_1, {}^2\underline{\omega}_2, {}^3\underline{\omega}_3$: 6 multiplications and 3 additions are saved.

${}^1\dot{\underline{\omega}}_1, {}^2\dot{\underline{\omega}}_2, {}^3\dot{\underline{\omega}}_3$: 10 multiplications and 7 additions are saved.

${}^4\ddot{\mathbf{p}}_4$: 4 multiplications and 4 additions are saved. Since ${}^2\ddot{\mathbf{p}}_4$ is known, then

$$ {}^3\ddot{\mathbf{p}}_4 = \mathbf{A}_3^T\,{}^2\ddot{\mathbf{p}}_4, \qquad {}^4\ddot{\mathbf{p}}_4 = \mathbf{A}_4^T\,{}^3\ddot{\mathbf{p}}_4 $$

requires 8 multiplications and 4 additions for evaluation instead of 12 multiplications and 8 additions from (82).

${}^2\ddot{\mathbf{p}}_2$: 8 multiplications and 6 additions are saved. It is not necessary to compute $\ddot{\mathbf{p}}_2$ because ${}^2\ddot{\mathbf{r}}_2$ can be computed without it and $\ddot{\mathbf{p}}_4$ is known. From (82) and (83),

$$ {}^2\ddot{\mathbf{r}}_2 = \mathbf{A}_2^{T\,1}\ddot{\mathbf{p}}_1 + {}^2\dot{\underline{\omega}}_2 \times ({}^2\mathbf{p}_2^* + {}^2\mathbf{r}_2^*) + {}^2\underline{\omega}_2 \times ({}^2\underline{\omega}_2 \times ({}^2\mathbf{p}_2^* + {}^2\mathbf{r}_2^*)) \qquad (89) $$

${}^3\ddot{\mathbf{p}}_3$: 4 multiplications and 2 additions saved. Similarly, it is not necessary to compute $\ddot{\mathbf{p}}_3$ because ${}^3\ddot{\mathbf{r}}_3$ can be computed from ${}^3\ddot{\mathbf{p}}_4$.

$$ {}^3\ddot{\mathbf{p}}_3 = {}^3\ddot{\mathbf{p}}_4 - {}^3\dot{\underline{\omega}}_3 \times s_4\,{}^3\mathbf{z}_3 - {}^3\underline{\omega}_3 \times ({}^3\underline{\omega}_3 \times s_4\,{}^3\mathbf{z}_3) \qquad (90) $$

$$ {}^3\ddot{\mathbf{r}}_3 = {}^3\ddot{\mathbf{p}}_4 + {}^3\dot{\underline{\omega}}_3 \times ({}^3\mathbf{r}_3^* - s_4\,{}^3\mathbf{z}_3) + {}^3\underline{\omega}_3 \times ({}^3\underline{\omega}_3 \times ({}^3\mathbf{r}_3^* - s_4\,{}^3\mathbf{z}_3)) \qquad (91) $$

where ${}^3\mathbf{r}_3^* - s_4\,{}^3\mathbf{z}_3$ is fixed and is precomputed.

${}^4\underline{\omega}_4, {}^5\underline{\omega}_5, {}^6\underline{\omega}_6$: 3 multiplications and 2 additions saved. This results from calculating ${}^6\underline{\omega}_6$ directly from $\underline{\omega}_6$ and the components of ${}^4\underline{\omega}_h$ separately.

$$ {}^6\underline{\omega}_6 = \mathbf{W}_6^T \underline{\omega}_6 \qquad (92) $$

$$ {}^5\underline{\omega}_6 = \mathbf{A}_6\,{}^6\underline{\omega}_6 \qquad (93) $$

$$ {}^4\underline{\omega}_6 = \mathbf{A}_5\,{}^5\underline{\omega}_6 \qquad (94) $$

$$ {}^4\underline{\omega}_3 = \mathbf{A}_4^{T\,3}\underline{\omega}_3 \qquad (95) $$

$$ {}^4\underline{\omega}_h = {}^4\underline{\omega}_6 - {}^4\underline{\omega}_3 \qquad (96) $$

$$ {}^4\underline{\omega}_4 = {}^4\underline{\omega}_3 + {}^4\mathbf{y}_4\dot{\theta}_4 \qquad (97) $$

$$ {}^5\underline{\omega}_5 = {}^5\underline{\omega}_6 - {}^5\mathbf{z}_5\dot{\theta}_6 \qquad (98) $$

Ordinarily, 24 multiplications and 19 additions would have been required to calculate these quantities.

${}^4\underline{\dot{\omega}}_4, {}^5\underline{\dot{\omega}}_5, {}^6\underline{\dot{\omega}}_6$: 5 multiplications and 4 additions saved. Similarly, some savings can be accomplished by computing ${}^6\underline{\dot{\omega}}_6$ from $\underline{\dot{\omega}}_6$ and the components of ${}^4\underline{\dot{\omega}}_h$ separately.

$$ {}^6\underline{\dot{\omega}}_6 = \mathbf{W}_6^T \underline{\dot{\omega}}_6 \tag{99} $$

$$ {}^5\underline{\dot{\omega}}_6 = \mathbf{A}_6 {}^6\underline{\dot{\omega}}_6 \tag{100} $$

$$ {}^4\underline{\dot{\omega}}_6 = \mathbf{A}_5 {}^5\underline{\dot{\omega}}_6 \tag{101} $$

$$ {}^4\underline{\dot{\omega}}_3 = \mathbf{A}_4^{T\,3}\underline{\dot{\omega}}_3 \tag{102} $$

$$ {}^4\underline{\dot{\omega}}_h = {}^4\underline{\dot{\omega}}_6 - {}^4\underline{\dot{\omega}}_3 - {}^4\underline{\omega}_3 \times {}^4\underline{\omega}_h \tag{103} $$

$$ {}^4\underline{\dot{\omega}}_4 = {}^4\underline{\dot{\omega}}_3 + {}^4\mathbf{y}_4\ddot{\theta}_4 + {}^4\underline{\omega}_3 \times {}^4\mathbf{y}_4\dot{\theta}_4 \tag{104} $$

$$ {}^5\underline{\dot{\omega}}_5 = {}^5\underline{\dot{\omega}}_6 - {}^5\mathbf{z}_5\ddot{\theta}_6 - {}^5\underline{\omega}_5 \times {}^5\mathbf{z}_5\dot{\theta}_6 \tag{105} $$

Ordinarily, 36 multiplications and 31 additions are required for these quantities.

7.4. Simplified Inertial Parameters

No presumptions were made above about the inertial parameters of the manipulator, namely the center of gravity, the principal inertias, and the orientation of the principal axes of inertia for each link. While the kinematic structure of manipulators is deliberate, the inertial parameters are seldom a factor in design. Therefore a simplifying set of inertial parameters cannot be assumed *a priori.* Paul (1981) considered one way in which the dynamics might be simplified if a particular set were assumed. In the present study, if the center of gravity and the principal axes lined up with the internal link coordinate system, an additional simplification of 174 multiplications and 168 additions in the dynamics would result.

${}^i\ddot{\mathbf{r}}_i$: 60 multiplications and 54 additions are saved. If ${}^i\mathbf{r}_i^*$ lies along a coordinate axis, then each ${}^i\ddot{\mathbf{r}}_i$ requires only 8 multiplications and 6 additions for evaluation.

${}^i\mathbf{n}_{i-1,i}$: 24 multiplications and 24 additions are saved. The term $({}^i\mathbf{p}_i^* + {}^i\mathbf{r}_i^*) \times {}^i\mathbf{f}_i$ now requires 2 multiplications and 2 additions for its formation and its addition, where ${}^i\mathbf{p}_i^* + {}^i\mathbf{r}_i^*$ lies along an axis.

${}^i\mathbf{n}_i$: 90 multiplications and 90 additions are saved. If ${}^i\mathbf{I}_i$ is aligned with the coordinate axes, then it is diagonal. Three additions are saved in the ${}^i\underline{\omega}_i \times {}^i\mathbf{I}_i {}^i\underline{\omega}_i$ term because the principal inertia differences can be precomputed.

If the simplified inertial parameters were considered along with all the other computational savings, then the dynamics would require only 194 multiplications and 138 additions. Table 5 summarizes the dynamics complexity for the various kinematic and dynamic conditions that have been considered above. It appears that the dynamic equations are even more approachable from a computational standpoint than had been considered heretofore (Hollerbach 1980). Under the best conditions, an exact evaluation of the dynamics has roughly the same complexity as a full evaluation of the inverse kinematics (Table 4).

Condition	Multiplications	Additions
General Rotary Manipulator	688	558
Spherical Wrist Manipulator	408	324
Precomputed Inverse Kinematics	368	296
Simplified Inertial Parameters	194	138

Table 5. Dynamics computation complexity for a 6-dof manipulator under various conditions.

8. Discussion and Summary

An algorithm for calculation of the inverse kinematic accelerations has been presented, which is the most efficient one to date. Based on a method developed by Featherstone (1983), the algorithm directly takes advantage of the structure of spherical-wrist manipulators to decompose the 6-dof inverse problem into two 3-dof inverse problems through a 4-step procedure. The resultant equations for the first 3 joints, but not the last three, are particular to a given manipulator structure, but the technique is easily extended to other structures. One could envision a catalog of equations for the most common manipulator configurations.

Spherical wrists are the most important case for current 6-dof robots and are becoming standard. An additional simplifying kinematic criterion, namely that neighboring joint axes are oriented parallel or orthogonal to each other, is almost always followed in manipulator design and aids the solution to the two 3-dof kinematic problems.

These two kinematic criteria have a simplifying effect on the dynamic equations as well. Since in the recursive Newton-Euler equations (Luh, Walker, and Paul 1980a) the computations are carried out in internal link coordinates, the transformations between neighboring links are simplified. Angular velocities and accelerations as well as linear accelerations of the link origins are calculated more simply, even as they are in the inverse kinematics computation. Because the inverse kinematic acceleration algorithm produces some of these last-mentioned vectors as well, then a combined inverse kinematic acceleration/inverse dynamics computation results in some savings for the inverse dynamics calculation.

The implication of these algorithms is that the computational requirement for either the inverse kinematics or the inverse dynamics is low enough to warrant easy real-time implementation. Recent control algorithms based on derived hand acceleration (Luh, Walker, and Paul 1980b, Freund 1982, Hogan and Cotter 1982) are made computationally feasible, making these algorithms more competitive with schemes which close force loops around the hand to avoid the inverse kinematic problem (Khatib 1980). The recursive Newton-Euler formulation is made even more efficient; if simplifying inertial parameters are assumed, the dynamics complexity roughly equals the inverse kinematics complexity. It would seem that a general Cartesian trajectory control ability, involving full inverse kinematics and dynamics as well as sophisticated hand-based control laws, is within reach.

References

1. **Denavit, J., and Hartenberg, R.S.**, "A kinematic notation for lower pair mechanisms based on matrices," *J. Applied Mechanics* 22 (June, 1955), 215–221.

2. **Featherstone, R.**, "Position and velocity transformations between robot end effector coordinates and joint angles," *Int. J. Robotics Research* 2, 2 (June, 1983).

3. **Hogan, N., and Cotter, S. L.**, "Cartesian impedance control of a nonlinear manipulator," *ASME WAM Robotics Symp.* , 1982.

4. **Hollerbach, J.M.**, "A recursive formulation of Lagrangian manipulator dynamics," *IEEE Transactions on Systems, Man, and Cybernetics* SMC-10, 11 (1980), 730-736.

5. **Johnson, T.L.**, "Feedback Control", In: *Robot Motion: Planning and Control* (edited by Brady, J.M., Hollerbach, J.M., Johnson, T.L., Lozano-Perez, T., and Mason, M.T.) , MIT Press, Cambridge, Mass., 1983.

6. **Khatib, Oussama**, Commande Dynamique dans l'Espace Operationnel des Robots Manipulateurs en Presence d'Obstacles, Docteur Ingenieur Thesis, L'Ecole Nationale Superieure de l'Aeronautique et de l'Espace, 1980.

7. **Luh, J.Y.S., Walker, M. W., and Paul, R.P.**, "On-line computational scheme for mechanical manipulators," *Journal of Dynamic Systems, Measurement, and Control* 102 (1980a), 69-76.

8. **Luh, J.Y.S., Walker, M.W., and Paul, R.P.**, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Automatic Control* 25, 3 (1980b), 468–474.

9. **Paul, R.P.**, *Robot Manipulators: Mathematics, Programming, and Control* , MIT Press, Cambridge, Mass., 1981.

10. **Pieper, D.L.**, The Kinematics of Manipulators under Computer Control, Ph.D. Thesis, Department of Computer Science, Stanford University, 1968.

11. **Taylor, R.H.**, "Planning and execution of straight-line manipulator trajectories," *IBM J. Research and Development* 23 (1979), 424–436.

12. **Whitney, D.E.**, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Systems* MMS-10 (1969), 47–53.

13. **Whitney, D.E.**, "The mathematics of coordinated control of prostheses and manipulators," *J. Dynamic Systems, Measurement, Control* (Dec. 1972), 303–309.

Appendix I

The wrist velocity is given by

$$\dot{\mathbf{p}}_4 = \underline{\omega}_2 \times a_2\mathbf{x}_2 + \underline{\omega}_3 \times s_4\mathbf{z}_3 \tag{106}$$

where

$$\underline{\omega}_3 = \underline{\omega}_2 + \mathbf{z}_2\dot{\theta}_3 \tag{107}$$

Substituting above,

$$\dot{\mathbf{p}}_4 = \underline{\omega}_2 \times (a_2\mathbf{x}_2 + s_4\mathbf{z}_3) + \dot{\theta}_3\mathbf{z}_2 \times \mathbf{z}_3 s_4 \tag{108}$$

Noting that $\mathbf{p}_w = a_2\mathbf{x}_2 + s_4\mathbf{z}_3$ and $\mathbf{x}_3 = -\mathbf{z}_2 \times \mathbf{z}_3$,

$$\dot{\mathbf{p}}_4 = \underline{\omega}_2 \times \mathbf{p}_w - s_4\dot{\theta}_3\mathbf{x}_3 \tag{109}$$

where

$$\underline{\omega}_2 = \mathbf{z}_0\dot{\theta}_1 + \mathbf{z}_1\dot{\theta}_2 \tag{110}$$

The wrist velocity is best evaluated in joint 2 coordinates, a reflection of the regular kinematic structure of the first 3 joints. Since the rotation axes are either perpendicular or parallel to each other, then a rotation axis in one coordinate system will also be a major axis in the next coordinate system. Link 2 coordinates are the most convenient because they are situated in the middle of the coordinate systems. Starting first with the angular velocity ${}^2\underline{\omega}_2$,

$${}^2\mathbf{z}_0 = {}^2\mathbf{y}_1 = \begin{bmatrix} s\theta_2 \\ c\theta_2 \\ 0 \end{bmatrix} \tag{111}$$

$${}^2\mathbf{z}_1 = {}^2\mathbf{z}_2 \tag{112}$$

$${}^2\underline{\omega}_2 = \begin{bmatrix} \dot{\theta}_1 s\theta_2 \\ \dot{\theta}_1 c\theta_2 \\ \dot{\theta}_2 \end{bmatrix} \tag{113}$$

The vector $\mathbf{p}_w$ in link 2 coordinates is

$${}^2\mathbf{p}_w = \begin{bmatrix} a_2 - s_4 s\theta_3 \\ s_4 c\theta_3 \\ 0 \end{bmatrix} \tag{114}$$

Thus the cross product term in (109) evaluates to

$$
{}^2\underline{\omega}_2 \times {}^2\mathbf{p}_w = \begin{bmatrix} -\dot{\theta}_2\,{}^2p_{wy} \\ \dot{\theta}_2\,{}^2p_{wx} \\ \dot{\theta}_1({}^2p_{wy}s\theta_2 - {}^2p_{wx}c\theta_2) \end{bmatrix} = \begin{bmatrix} -\dot{\theta}_2(s_4c\theta_3) \\ \dot{\theta}_2(a_2 - s_4s\theta_3) \\ -\dot{\theta}_1\,{}^1p_{wx} \end{bmatrix} \tag{115}
$$

where it is noticed that ${}^1p_{wx} = -({}^2p_{wy}s\theta_2 - {}^2p_{wx}c\theta_2)$. Finally,

$$
{}^2\dot{\mathbf{p}}_4 = {}^2\underline{\omega}_2 \times {}^2\mathbf{p}_w - s_4\dot{\theta}_3\,{}^2\mathbf{x}_3 \tag{116}
$$

$$
{}^2\mathbf{x}_3 = (c\theta_3, s\theta_3, 0) \tag{117}
$$

Collecting all terms, (79) becomes

$$
{}^2\dot{\mathbf{p}}_4 = \begin{bmatrix} -s_4c\theta_3(\dot{\theta}_2 + \dot{\theta}_3) \\ a_2\dot{\theta}_2 - s_4s\theta_3(\dot{\theta}_2 + \dot{\theta}_3) \\ -\dot{\theta}_1\,{}^1p_{wx} \end{bmatrix} \tag{118}
$$

Appendix II

The wrist acceleration in link 2 coordinates is

$$
{}^2\ddot{\mathbf{p}}_4 = {}^2\dot{\underline{\omega}}_2 \times {}^2\mathbf{p}_w + {}^2\underline{\omega}_2 \times {}^2\dot{\mathbf{p}}_4 - s_4\ddot{\theta}_3\,{}^2\mathbf{x}_3 - s_4\dot{\theta}_3\,{}^2\underline{\omega}_3 \times {}^2\mathbf{x}_3 \tag{119}
$$

Starting with the first term on the right,

$$
{}^1\dot{\underline{\omega}}_2 = (\dot{\theta}_1\dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2) \tag{120}
$$

$$
{}^1\mathbf{p}_w = ({}^1p_{wx}, {}^1p_{wy}, 0) \tag{121}
$$

$$
{}^1\dot{\underline{\omega}}_2 \times {}^1\mathbf{p}_w = \begin{bmatrix} -\ddot{\theta}_2\,{}^1p_{wy} \\ \ddot{\theta}_2\,{}^1p_{wx} \\ \dot{\theta}_1\dot{\theta}_2\,{}^1p_{wy} - \ddot{\theta}_1\,{}^1p_{wx} \end{bmatrix} \tag{122}
$$

$$
\mathrm{A}_2^T({}^1\dot{\underline{\omega}}_2 \times {}^1\mathbf{p}_w) = \begin{bmatrix} \ddot{\theta}_2({}^1p_{wx}s\theta_2 - {}^1p_{wy}c\theta_2) \\ \ddot{\theta}_2({}^1p_{wx}c\theta_2 + {}^1p_{wy}s\theta_2) \\ \dot{\theta}_1\dot{\theta}_2\,{}^1p_{wy} - \ddot{\theta}_1\,{}^1p_{wx} \end{bmatrix} \tag{123}
$$

This expression can be further simplified by noting that

$$
{}^1p_{wx}s\theta_2 - {}^1p_{wy}c\theta_2 = -s_4c\theta_3 \tag{124}
$$

$$
{}^1p_{wx}c\theta_2 + {}^1p_{wy}s\theta_2 = a_2 - s_4s\theta_3 \tag{125}
$$

The last term on the right is evaluated as follows:

$$ {}^3\underline{\omega}_3 \times {}^3\mathbf{x}_3 = \begin{bmatrix} 0 \\ {}^3\omega_{3z} \\ -\,{}^3\omega_{3y} \end{bmatrix} \tag{126} $$

$$ \mathrm{A}_3({}^3\omega_3 \times {}^3\mathbf{x}_3) = \begin{bmatrix} s\theta_3\, {}^3\omega_{3y} \\ -c\theta_3\, {}^3\omega_{3y} \\ -\,{}^3\omega_{3z} \end{bmatrix} \tag{127} $$

Collecting all unknowns on one side,

$$ \begin{bmatrix} -s_4 c\theta_3 \ddot{\theta}_2 - s_4 c\theta_3 \ddot{\theta}_3 \\ (a_2 - s_4 s\theta_3)\ddot{\theta}_2 - s_4 s\theta_3 \ddot{\theta}_3 \\ -\,{}^1p_{wx}\ddot{\theta}_1 \end{bmatrix} = {}^2\ddot{\mathbf{p}}_4 - {}^2\underline{\omega}_2 \times {}^2\dot{\mathbf{p}}_4 + \begin{bmatrix} s_4 \dot{\theta}_3 s\theta_3\, {}^3\omega_{3y} \\ -s_4 \dot{\theta}_3 c\theta_3\, {}^3\omega_{3y} \\ -s_4 \dot{\theta}_3\, {}^3\omega_{3z} - {}^1p_{wy}\dot{\theta}_1\dot{\theta}_2 \end{bmatrix} \tag{128} $$