No. 2, (1982).

32. Khosla, P.K., and Neuman, C.P., "Computational Requirements of Customized Newton-Euler Algorithms", Journal of Robotic Systems, Vol. 2, No. 3, pp. 309-327, (1985).

33. Vukobratovic, M., and Kircanski, N., "A Method for Computer-Aided Construction of Analytical Models of Robotic Manipulators", Proc. of the first Int. IEEE Conf. on Robotics, Paul, R.P., ed., Atlanta, GA, pp. 519-529, March 13-15, (1984).

34. Neuman, C.P., and Murray, J.J., "Computational Robot Dynamics: Foundations and Applications", Journal of Robotics Systems, Vol. 2, No. 4, (1985).

35. Vukobratovic, M., and Kircanski, N.M., "Computer-Oriented Method for Linearization of Dynamic Models of Active Spatial Mechanisms", Journal of Mechanisms and Machine Theory, Vol. 17, pp. 21-32, (1982).

36. Wu Fei, and Zhou Oizhao, "An Improved Recursive Algorithm Based on the Vectorial Expressions of the Dynamic Equations for Robot Manipulators", Int. Journal of Robotics & Automation, Vol. 3, No. 3, pp. 127-132, (1988).

37. Stepanenko, Y., "Dynamic Robot Models: A Compound Vector Approach", Int. Journal of Robotics & Automation, Vol. 3, No. 1, (1988).

(June 1978).

10. Orin, D.E., McGhee, R.B., Vukobratovic, M., and Hartoch, G., "Kinematic and kinetic analysis of open-chain linkages utilizing Newton-Euler methods", Math. Bios. Vol. 43, pp. 107-130, (1979).
11. Luh, J.Y., Walker, M.W., and Paul, R.P., "On-line Computational Scheme for Mechanical Manipulators", Journal of Dyn. Sys., Meas., and Cont. Vol. 102, pp. 69-76, (1980).
12. Armstrong, W.W., "Recursive Solution to the Equations of Motion of an N-Link Manipulator", Proc. 5th World Congress, Theory of Machines, Mechanisms, Vol. 2, pp. 1343-1346, (July 1979).
13. Koplik, J., and Leu, M.C., "Computer Generation of Robot Dynamics Equations and the Related Issues", Journal of Robotic Systems, Vol. 3, No. 3, pp. 301-319, (1986).
14. Vukobratovic, M., "Dynamics of Active Articulated Mechanisms and Synthesis of Artificial Motion", Mechanism Machine Theory, Vol. 13 pp. 1-56, (1978).
15. Stepanenko, Y., and Vukobratovic, M., "Dynamics of Articulated Open - chain Active Mechanisms", Math. Bios., Vol. 28, pp. 137-170, (1976).
16. Craig, J.J., "Introduction to Robotics, Mechanics, and Control", Addison Wesley Pub. Co., (1986).
17. Vukobratovic, M., and Kircanski, N., "Real-Time Dynamics of Manipulation Robots", Springer-Verlag, New York, (1985).
18. Kane, T.R., "Dynamics of Nonholonomic Systems", Journal of Applied Mechanics, Vol. 28, pp. 574-578, (1961).
19. Huston, R.L., and Kelly, R.A., "The Development of Equations of Motions of Single-Arms Robots", IEEE Trans. Sys. Man. Cyber. SMC-12, pp. 259-265, March (1982).
20. Faessler, H., "Computer-Assisted Generation of Dynamical Equations for Multibody Systems", Int. Journal of Robotic Research, Vol. 5, No. 3, Fall (1986).
21. Kane, T.R., and Levinson, D.A., "The Use of Kane's Dynamical Equations in Robitics", Int. Journal of Robotics Research, Vol. 2, No. 3, pp. 3-21, (1983).
22. Huston, R.L., Passerello, C.E., and Harlow, M.W., "Dynamics of Multirigid-body Systems", ASME Trans., Journal of Applied Mechanics, Vol. 45, pp. 889-894, (1978).
23. Kane, T.R., and Levinson, D.A., "Dynamics", McGraw-Hill, N.Y., (1985).
24. Kane, T.R., and Wang, C.F., "On the Derviation of Equations of Motion", Journal of Soc. for Ind. and Applied Math., Vol. 13, pp. 487-492, (1965).
25. Huston R.L., and King, T.P., "Dynamics of Redundant Robots- Inverse Solutions", Robotica Int. Journal, Vol. 4, No. 4, pp. 263-268, (1986).
26. Paul, R., "Robot manipulators: mathematics, programming, and control", Cambridge: MIT Press, (1981).
27. Coiffet, P., "Robot technology Vol. 1: modelling and control", Paris, France: Hermes Publishing, (1983).
28. Meghdari, A., and Chiou, B.C., "A Comparative Survey of Robot Manipulator Dynamics", Proc. of the 9th Annual IEEE-ISE Exp. & Symp., pp. 50-58, Albuquerque, N.M., (May 1987).
29. Wampler, C.W., "Computer methods in manipulator kinematics, dynamics, and control: a comparative study", Ph.D. Thesis, Stanford University, Dept. of Mech. Engineering, (1984).
30. Murray, J.J., and Numan, P., "Arm: an algebraic robot dynamic modelling program", 1st Int. IEEE Conf. on Robotics, Atlanta, GA, (March 1984).
31. Silver, W.M., "On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators", The Int. Journal of Robotics Research, Vol. 1,

Table 1. Comparison of Robot Dynamics Formulations for n = 6

Author	*Method*	*Multiplications*	*Additions*
Uicker/Kahn	Lagrangian dynamics (4X4) matraices	66, 271	51,548
Waters	Backward Recursion Lagrangian dynamics	7,051	5652
Hollerbach	Forward Recursion Lagrangian dynamics (4X4)	4388	3586
Hollerbach	Forward Recursion Lagrangian dynamics (3X3)	2,195	1,719
Newton-Euler	Recursive Newton-Euler dynamics	852	738
Kane/Levinson	Kane dynamics	646	394
Raibert/Horn	Configuration Space Method (CSM)	468	264
Yang/Tzeng	Dynamics simplification by design	72	34+4(trig. function)

REFERENCES

1. Bejczy, A.K., "Robot Arm Dynamics and Control". Jet Propulsion laboratories, Tech. Memo. 33-669, (February 1974).
2. Uicker, J.J., "On the Dynamic Analysis of Spatial Linkages Using 4x4 Matrices", Ph.D. dissertation Northwestern Univ., (August 1965).
3. Kahn, M.E., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", Standford Artificial Intelligence Project Memo. AIM 106, (December 1969).
4. Hollerbach, J.M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamic Formulations", IEEE Trans. on Systems, Man, and Cybernetics, SMC-10 pp. 730-736, (November 1980).
5. Waters, R.C.,"Mechanical arm control", M.I.T. Aritificial Intelligence Lab. Memo. 549, (October 1979).
6. Shahinpoor, M., "A Robot Engineering Text Book", Harper & Row Pub. (1987).
7. Yang, D.C.H., and Tzeng, S.W., "Simplification and linearization of manipulator dynamics by design of inertia distribution", The International Journal of Robotics Research, Vol. 5, No. 3, (1986).
8. Vukobratovic, M., and Potkonjack, V., "Dynamics of Manipulation Robots", Springer-Verlag, New York, (1982).
9. Horn, B.K.P., and Raibert, M.H., "Configuration Space Control", The Ind. Robot, pp. 69-73,

by examining the complete expanded version of Lagrange's equation to redesign the link's inertia property, including inertia, mass, and the location of the center of mass. This results in elimination of all the potential energy terms and about 90 percent of the kinetic energy terms. And further, the coefficients of most of the nonlinear terms in system's dynamic equations become zero. For a detailed design methodology the reader is refered to [7]. As a result of simple design technique, for a six link arm, 34 + 4 (trigonometrical functions) additions and 72 multiplications are required.

9. COMPUTER GENERATION OF ROBOT DYNAMICS EQUATIONS

As it has been witnessed from the brief overview of the literature, the dynamics problem of manipulators have attracted considerable research and several techniques have evolved. Recent developments in this area include the computer generation of robot dynamics equations [13, 20, 29-30]. In recent years, several algorithms for generating dynamical equations for manipulators have been developed. Most of these algorithms are based on Lagrange's equations [24-28], or the Newton-Euler method [10-12]. Furthermore, a computer oriented technique based on the Kane's equations is developed in [20] that facilitates the formulation of dynamics equations. However, since an efficient implementation most often requires explicit equations governing a specific type of manipulator, only a few programs have been developed that generate dynamics equations in symbolic form. Therefore, this area shows great potential for future research and development.

10. CONCLUSIONS

A comparative look throughout the above discussion enables future researchers to recognize the advantages and disadvantages of the various techniques. The forward and backward recursive equations reduce the number of computations for the generalzed forces. Setting up the dynamics via (4x4) matrices are inefficient because of their requirement for combining translation with rotation. Reformulation of the dynamics equations via (3x3) rotation matrices to specify the orientation of the links and displacement vectors to represent their position improves the computational efficiency a great deal as shown in Table 1.

Although, the Newton-Euler approach has attracted much interest in the formulation of manipulators dynamics equations, in 1982, Silver [29], shows that there is, in fact, no fundamental difference in computational efficiency between Lagrangian and Newton-Euler formulations. The efficiency of the New-Euler formulation is due to two factors: 1) the recursive structure of the computation, and 2) the representation chosen for the rotational dynamics. Where both of these factors can be achieved in Lagrangian formulation. In doing so, Lagrangian dynamics is formulated based on the angular velocity vector $\underset{\sim}{\omega}$ instead of the derivative of the rotation matrix $\dot{T}$. This is done due to the fact that the angular motion of a rigid body could be described equally well by either of the two.

The Configuration Space Method (CSM) is the most efficient formulation for $n < 9$ joints. Although there is still an n^3 dependence for the additions and multiplications in the scheme, the much smaller coefficients on the ploynomial terms represent a greatly reduced computational cost. In application to large arms with powerful actuators and large link masses, configuration space memory (CSM) can increase the maximum velocity without adversely affecting the stability control resulted by uncompensated time varying inertial terms or non-linear velocity dependent terms.

equations involving the joint force and moment components which are clearly a convenience for solving the inverse problem of computing these components when the system kinematics is known. For a six link robot manipulator this technique has shown to require only 394 additions, and 646 multiplications, which shows a great computational efficiency in comparison with the previously discussed methods.

7. THE RAIBERT/HORN CONFIGURATION SPACE METHOD

A method of evaluating the equations of motion that makes real time compensation for all important terms possible is presented by Raibert and Horn[9]. They proposed a control method that explicitly compensates for configuration - dependent gravity, acceleration, and velocity forces which become especially important during rapid simultaneous motions of a number of joints. The configuration space method (CSM) is derived from the Lagrangian dynamics formulated as:

$$\tau_i = G_i(\underset{\sim}{q}) + \sum_{j=1}^{n} D_{ij}(\underset{\sim}{q})\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} C_{ijk}(\underset{\sim}{q})\dot{q}_j\dot{q}_k \quad (16)$$

Where, τ_i represents the motor torque acting on the ith joint. And G_i is the gravitational force acting on the ith joint. A coefficient of the form D_{ij} is the inertial force acting on the ith joint as a result of accelerations of the jth joint (coupling inertia). Finally, coefficients of the form C_{ijk} are velocity dependent forces acting on the ith joint as a result of rotations about the jth and kth joints (coriolis coefficients). Equation (16) can also be written in matrix form as:

$$\underset{\sim}{\tau} = \underset{\sim}{G}(\underset{\sim}{q}) + \underset{\sim}{J}(\underset{\sim}{q}).\ddot{\underset{\sim}{q}} + \begin{bmatrix} \dot{\underset{\sim}{q}}^t . C_1(\underset{\sim}{q}) . \dot{\underset{\sim}{q}} \\ \dot{\underset{\sim}{q}}^t . C_2(\underset{\sim}{q}) . \dot{\underset{\sim}{q}} \\ \vdots \\ \dot{\underset{\sim}{q}}^t . C_n(\underset{\sim}{q}) . \dot{\underset{\sim}{q}} \end{bmatrix} \quad (17)$$

Where, τ and $\underset{\sim}{G}$ are n-vectors, $\underset{\sim}{J}$ and each C_i are (nxn) matrices. Although these equations are very complicated, but it is noticable that each of the coefficients in the equations (17), $\underset{\sim}{G}$, $\underset{\sim}{J}$, and the C_i's, is only dependent on the configuration of the arm. Rather than computing these coefficients each time they are needed, the (CSM) approach is to look them up in a pre-defined, multi-dimensional memory organized by these positional variables. Furthermore, a space that has one dimension for each joint of the manipulator is defined where each point in this space corresponds to a configuration. Then a multi-dimensional memory corresponding to this configuration space is created where the values of the above coefficients can be computed and stored for use whenever the equations of motion are to be evaluated. Formulations based on (CSM) offer an efficient way of dynamics calculations for a manipulator with $n < 9$ joints. For a six link arm formulated by (CSM) 264 additions, and 468 multiplications are required.

8. THE YANG/TZENG SIMPLE DESIGN METHOD

In 1986 Yang, and Tzeng [7] introduced a new approach for designing simple manipulators with better dynamic behavior. This technique obviously differs very much from the previously discussed methods. Development of this is based on elimination of coefficients of nonlinear terms in the robotic system's potential and kinetic energy equations. Furthermore, a set of design criteria regarding the links inertia distribution may be established for different types of manipulators. As a result, a robot design based upon these criteria will have much simplified dynamics characteristics. They have shown that for some configurations of three and four link manipulators, it is possible to design the robotic structure so that completely linearized dynamic equations are the results. They further improved the robotic structure n and then by applying the Newton-Euler equations, the total force vector $\underline{F}_i$ and the total moment vector $\underline{N}_i$ acting on each link i are computed (backward recursion). In the second step, forces and torques of interaction and joint actuator torques are determined recursively from link n to the base of the arm (forward recursion). The general algorithem for the case where all joints are revolute is given below [4], and for sliding joints the reader may refer to Shahinpoor [6]. This formulation has shown to require 852 multiplications and 738 additions for a six link manipulator arm. Because of its efficiency, it is widely used in the formulation of robot manipulator dynamics.

Backward Recursions:

$$\underline{\omega}_i = \underline{\omega}_{i-1} + \underline{Z}_{i-1}\dot{q}_i$$
$$\dot{\underline{\omega}}_i = \dot{\underline{\omega}}_{i-1} + \underline{Z}_{i-1}\ddot{q}_i + \underline{\omega}_{i-1} + \underline{Z}_{i-1}\dot{q}_i$$
$$\ddot{\underline{p}}_i = \dot{\underline{\omega}}_i \times \underline{p}_i^* + \underline{\omega}_i \times (\underline{\omega}_i \times \underline{p}_i^*) + \ddot{\underline{p}}_{i-1}$$
$$\ddot{\underline{r}}_i = \underline{\omega}_i \times (\underline{\omega} \times \underline{r}_i^*) + \dot{\underline{\omega}}_i \times \underline{r}_i^* + \ddot{\underline{p}}_i$$
$$\underline{F}_i = M_i \ddot{\underline{r}}_i$$
$$\underline{N}_i = I_i^c \dot{\underline{\omega}}_i + \underline{\omega}_i \times (I^c_i \underline{\omega}_i)$$

Where:

- $\underline{\omega}_i$: is the angular velocity vector of link i
- $\dot{\underline{\omega}}_i$: is the angular acceleration vector of link i
- $\underline{F}_i$: is the total external force vector on link i
- $\underline{N}_i$: is the total external torque vector on link i
- I_i^c : is the inertia tensor of link i about its center of mass

Forward Recursion:

$$^{i-1}\underline{f}_i = \underline{F}_i + {}^{i}\underline{f}_{i+1}$$
$$^{i-1}\underline{n}_i = {}^{i}\underline{n}_{i+1} + \underline{N}_i + (\underline{P}_i^* + \underline{r}_i^*) \times \underline{F}_i + \underline{p}_i^* \times {}^{i}\underline{f}_{i+1}$$
$$\tau_i = \underline{Z}_{i-1} \cdot {}^{i-1}\underline{n}_i$$

Where:

- $^{i-1}\underline{f}_i$: the force exerted on link i by link i-1 refered to base frame
- $^{i-1}\underline{n}_i$: the moment exerted on link i by link i-1, refered to base frame
- τ_i : is the input actuator torque at the joint i.

A more efficient formulation of this technique is obtained by referring to local link coordinates rather than the global base coordinates [4] & [6].

6. THE KANE'S METHOD

In 1982 and 1983, Huston and Kelly [19], and Kane and Levinson [21] applied the oroginal Kane's dynamical equations [23] to robot manipulators and generated sets of dynamics equations. Their technique employs the Euler parameters and relative coordinates [22], partial velocities and partial angular velocities [23], and generalized speeds [24] in the development of the governing manipulators dynamic equations. This is an "algorithmic" approach extensively discussed in [19] and [20] leading to governing equations whose coefficients are readily evaluated by computer subroutines. It has the advantage of automatically eliminating "nonworking" internal constraint forces, without inclusion of tedious, and often unwieldy, differentiation of scalar energy functions and other similar manipulations. Relative coordinates are used to conveniently and efficiently define the system configuration, whereas the Euler parameters are used to define the robotic system orientation. The use of partial velocities and partial angular velocities leads to efficient computation of the generalized forces needed for Kanes dynamical equations in which their components also form the elements of the coefficients of the governing differential equations. And finally, the implementation of generalized speeds leads to dynamic equations which are in "canonical" form, and can readily be adapted to commonly used numerical integrators. Furthermore, it will decouple the

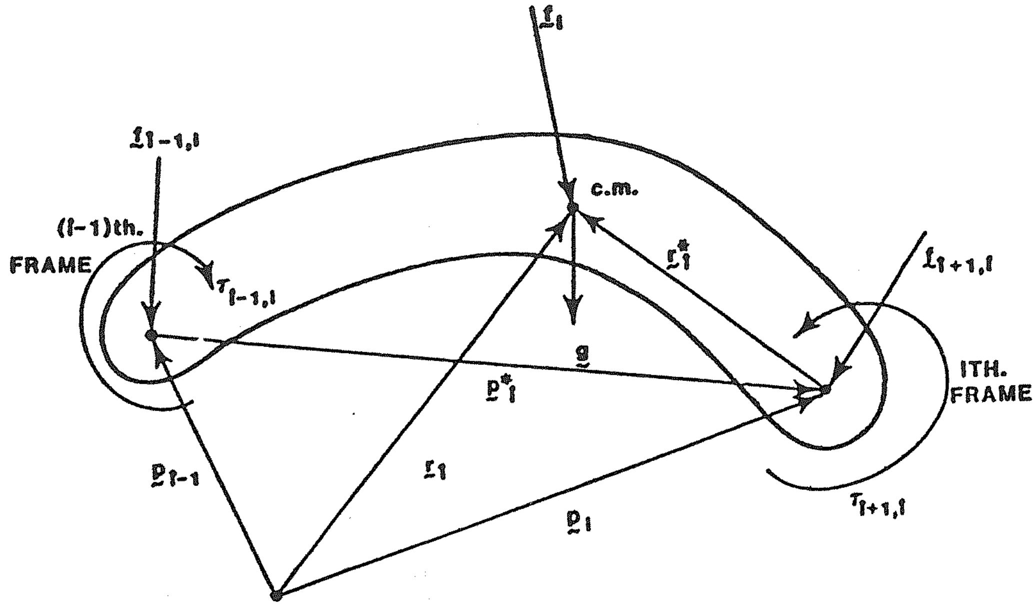

Fig. 3. Kinematical notations on a single link

$\underset{\sim}{r}_i^* \equiv$ a vector connecting the origin of frame i to mass center of link i,

$\underset{\sim}{P}_i \equiv$ a vector from origin of the base frame to the origin of frame i,

$\underset{\sim}{P}_i^* \equiv$ a vector connecting the origin of frame i-1 to origin of frame i,

$\ddot{T}_j \equiv$ same expression as equation (6), but with (3 x 3) rotation matrices,

$\underset{\sim}{n}_i \equiv \underset{\sim}{r}_i^* / m_i$,

$\ddot{\underset{\sim}{P}}_j \equiv \ddot{\underset{\sim}{P}}_{j-1} - \ddot{T}_j {}^j\underset{\sim}{P}_j^*$.

The final recurrence relation of the equation (10) is written as:

$$F_i = tr\,[\frac{\partial T_i}{\partial q_i} D_i] - \underset{\sim}{g}^t \frac{\partial T_i}{\partial q_i} d_i + Ia_i \ddot{q}_i \qquad (11)$$

Where:

$$D_i = A_{i+1} D_{i+1} + {}^i\underset{\sim}{p}_{i+1} \underset{\sim}{e}_{i+1} + {}^i\underset{\sim}{n}_i \ddot{\underset{\sim}{P}}_i^t + J_i \ddot{T}_i^t \qquad (12)$$

and,

$$\underset{\sim}{e}_i = \underset{\sim}{e}_{i+1} + m_i \ddot{p}_i^t + {}^i\underset{\sim}{n}_i^t \ddot{T}_i^t \qquad (13)$$

and d_i has the same expression as equation (9), but with (3 x 3) rotation matrices. With this formulation (3 x 3) matrices, the size of the coefficients of the complexity polynomial is reduced to more than 50% of the forward recursive lagrangian formulation with (4 x 4) matrices. This will result in 2195 multiplications and 1719 additions for the case of a six link manipulator arm.

5. THE RECURSIVE NEWTON- EULER DYNAMICS APPROACH

The Newton-Euler formulation of Manipulator Dynamics have been studied by many authors in the past [10-16]. In this formulation each link of a manipulator is treated as a rigid body with a known center of mass and inertia tensor. Then the Newton's second law:

$$\underset{\sim}{F} = m\underset{\sim}{a}^c \qquad (14)$$

along with its rotational analog, Euler's equation:

$$\underset{\sim}{N} = I^c . \dot{\underset{\sim}{\omega}} + \underset{\sim}{\omega} \times (I^c . \underset{\sim}{\omega}) \qquad (15)$$

are used to describe how forces, inertias, and accelerations relate. To apply the Newton-Euler equations to a set of kinematic chain, we first iteratively compute the velocities and accelerations from the base to link of link numbering direction can be expressed in a more compact form:

$$F_i = \sum_{j=i}^{n} \left\{ tr\left[\frac{\partial T_j}{\partial q_i} J_j \ddot{T}_j^t \right] - m_j \underline{g}^t \frac{\partial T_j}{\partial q_i} \underline{^j r_j} \right\} + I_{ai} \ddot{q}_i \quad (4)$$

Where:

$$\ddot{T}_j = \sum_{k=1}^{j} \frac{\partial T_j}{\partial q_k} \ddot{q}_k + \sum_{k=1}^{j} \sum_{m=1}^{j} \frac{\partial^2 T_j}{\partial q_k \partial q_m} \dot{q}_k \dot{q}_m \quad (5)$$

with this formulation, Velocity and acceleration relations $\dot{T}_j$, and $\ddot{T}_j$ are computed by straight forward differentiation of $T_j = T_{j-1} A_j$ and therefore:

$$\ddot{T}_j = \ddot{T}_{j-1} A_j + 2\dot{T}_{j-1} \frac{\partial A_j}{\partial q_j} \dot{q}_j + T_{j-1} \frac{\partial^2 A_j}{\partial q_j^2} \dot{q}_j^2 + T_{j-1} \frac{\partial A_j}{\partial q_j} \ddot{q}_j \quad (6)$$

with the formulation represented by equations (4) and (6), the real time computation of the required forces is proportional to the second power of the number of manipulator's links [4,5]. Therefore, for a six-link manpulator arm, there exist 7051 multiplications and 5652 additions to compute all torques. The primary reason for this improvement is the more efficient computation of the Coriolis and centrifugal forces, (i.e. it requires that only $(\partial^2 T_j / \partial q_j^2)$ be calculated rather than all the matrices $(\partial^2 T_j / \partial q_k \partial q_m)$

A forward link to link recursion of equation (4) leads to further efficiency of the Lagrangian dynamics. Therefore, rewriting equation (4) and noting that $\partial T_j / \partial q_i = (\partial T_i / \partial q_i)^i T_j$, and $^iT_i = I$, we have:

$$F_i = tr\left[\frac{\partial T_i}{\partial q_i} \sum_{j=i}^{n} {^iT_j} J_j \ddot{T}_j^t\right] - \underline{g}^t \frac{\partial T_i}{\partial q_i} \sum_{j=i}^{n} m_j {^iT_j} \underline{^j r_j} + I_{ai}\ddot{q}_i = tr\left[\frac{\partial T_i}{\partial q_i} D_i\right] - \underline{g}^t \left[\frac{\partial T_i}{\partial q_i}\right] \underline{d}_i \quad (7)$$

where:

$$D_i = \sum_{j=i}^{n} {^iT_j} J_j \ddot{T}_j^t = J_i \ddot{T}_i^t + A_{i+1} D_{i+1} \quad (8)$$

and,

$$\underline{d}_i = \sum_{j=i}^{n} m_j {^iT_j} \underline{^j r_j} = m_i \underline{^i r_i} + A_{i+1} d_{i+1} \quad (9)$$

In this formulation, the acceleration terms $\ddot{T}_i^t$ are first computed starting from the first to the nth link. And then the D_i and d_i terms are computed in the opposite direction starting from the nth link to the first link. In this formulation the number of multiplications and additions is only proportional to the first power of the number of links, resulting in 4388 multiplications and 3586 additions for a six link manipulator arm.

4. THE HOLLERBACH FORWARD RECURSIVE LAGRANGIAN DYNAMICS WITH (3 x 3) MATRICES

In 1980, Hollerbach [4] realized that the reformulation of Lagrangian dynamics through the use of (3 x 3) pure rotation matrices rather than (4 x 4) rotation- translation matrices provides the greatest reduction in computation. This is due to the fact that (3 x 3) matrix multiplications require 27 multiplications whereas (4 x 4) matrix multiplications require 64. The result would be a reduction of more than 50% in the coefficients of the computational terms.

In his analysis, Hollerbach used a rotation matrix T_i to represent the orientation of link i of the robot kinematic chain. The matrix T_i transforms the components of a vector with respect to a fixed frame. The angular motion of link i is therefore specified by $\dot{T}_i$, and $\ddot{T}_i$. Figure 3 clearly shows the kinematical notation used by Hollerbach to obtain the generalized forces:

$$F_i = \sum_{j=i}^{n} \left[tr\left[m_j \frac{\partial \underline{p}_j}{\partial q_i} \ddot{\underline{p}}_j^t + \frac{\partial \underline{p}_j}{\partial i} {^j\underline{n}_j}^t \ddot{T}_j^t + \frac{\partial T_j}{\partial q_i} {^j\underline{n}_j} \ddot{\underline{p}}_j^t + \frac{\partial T_j}{\partial q_i} J_j \ddot{T}_j^t \right] - m_j \underline{g}^t \frac{\partial T_j}{\partial q_i} \underline{^j r_j} \right] + I_{ai}\ddot{q}_i \quad (10)$$

where:

$\underline{r}_i$ ≡ a vector from origin of the base frame to mass center of link i, adjacent links. However, like any other mechanical system, an open-loop kinematic chain contains a natural set of generalized coordinates $\{q_i\}$ which completely specifies its position. For the case where the joint is revolute (rotary), the joint variable is described by θ_i whereas for prismatic (sliding) type joint, d_i will represent the corresponding joint variable. With these notation's Uicker/Kahn employed the general Lagrange equation:

$$\frac{d}{dt}\left[\frac{\partial L}{\partial q_i}\right] - \left[\frac{\partial L}{\partial \dot{q}_i}\right] = F_i \qquad (2)$$

where:

F_i = The generalized forces or torques,

q_i = The generalized coordinates,

L = The Lagrangian function $\equiv K - P$

K = The kinetic energy, and

P = The potential energy.

in conjunction with Denavit-Hartenberg (4 x 4) translation/rotation matrices T_i to represent the position and motion of the kinematic chain. The matrix T_i actually transforms the components of the position vector of a point P fixed with respect to the coordinate frame i (moving with it) to its components in a fixed base coordinate system. After expressing the kinetic and potential energies of the kinematic chain in terms of T_i's and their derivatives, an expression for the generalized forces F_i of an n-link manipulator arm has been derived [2–4]:

$$F_i = \sum_{j=i}^{n}\left\{\sum_{k=1}^{j}\left[tr\left[\frac{\partial T_j}{\partial q_i} J_j \frac{\partial T_j^t}{\partial q_k}\right]\right]\ddot{q}_k + \sum_{k=1}^{j}\sum_{m=1}^{j}\left[tr\left[\frac{\partial T_j}{\partial q_i} J_j \frac{\partial^2 T_j^t}{\partial q_k \partial q_m}\right]\dot{q}_k \dot{q}_m\right] - m_j \underline{g}^t \frac{\partial T_j}{\partial q_i}\, {}^{j}\underline{r}_j\right\} + Ia_i \ddot{q}_i \qquad (3)$$

where:

m_j : mass of link j,

Ia_i : actuator inertias at the joint (equivalent mass for the case of a prismatic joint),

$\underline{g}$: Gravity vector,

J_j : Inertial tensor, and

${}^{j}\underline{r}_j$: Coordinate of a point (center of mass) on link j described in link j's coordinate frame.

real time computation of equation (3) has been shown to be very slow due to the fact that the number of multiplications and additions involved in the operation is proportional to the forth power of the number of links [4]. Ignoring actuator inertias, in 1980 Hollerbach [4] has shown that for a six link manipulator arm, Uicker/Kahn formulation will result in 66271 multiplications, and 51548 additions in order to compute all torques. Furthermore, Luh, Walker, and Paul [11] have estimated that the time required to evaluate the torques for a point in the trajectory is about 9 seconds on a PDP–11/45. However, the ineffectiveness of this technique in real-time control of manipulator arms, have led the researchers to investigate various formulations of manipulator arms dynamics.

3. THE WATERS/HOLLERBACH LAGRANGIAN DYNAMICS WITH BACKWARD/FORWARD RECURSION

In 1979, Waters introduced five techniques to speed up the calculation of the required forces. Reformulating the dynamic equations so that they could be evaluated faster, simplifying the dynamic equations by eliminating insignificant terms, using direct acceleration servoing of the arm, using exterapolation and interpolation, and using precomputed values of the equations are some of the methods first noticed by Waters [5].

Bacward Recursion from link to link formulation of the generalized forces of equation (3) in the sense method [10–17]. Other algorithms based on Kane's method [17–25] have also been published in the literature. Recent works on robot dynamics are oriented toward the vectorial, computational, and symbolic derivations of dynamic equations [26–37], and a new design concept [7] which would result in manipulators with much simplified dynamics. In the following sections we shall briefly elaborate on each of the classical algorithms.

2. THE UICKER/KAHN FORMULATION OF MANIPULATOR DYNAMICS VIA LAGRANGIAN METHOD

In 1965, Uicker [2] used (4 x 4) matrices to develop a general Lagrangian based dynamic algorithm which included arbitarary spatial closed linkages. Later in 1969, Kahn [3] was the first who applied the Uicker's formulation to open-loop articulated kinematic chains. Considering the generalized adjacent links situation for manipulators shown in Fig. 2, one can readily show that the (4 x 4) Denavit - Hartenberg homogeneous transformation matrix representing the position and orientation of coordinate frame i with respect to frame i-1 is given by:

$$^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where:

a_i : Link Length
d_i : Link Offset
α_i : Link Twist, and
θ_i : Joint Angle.

are the four neccessary and sufficient parameters which completely describe the relative position of two

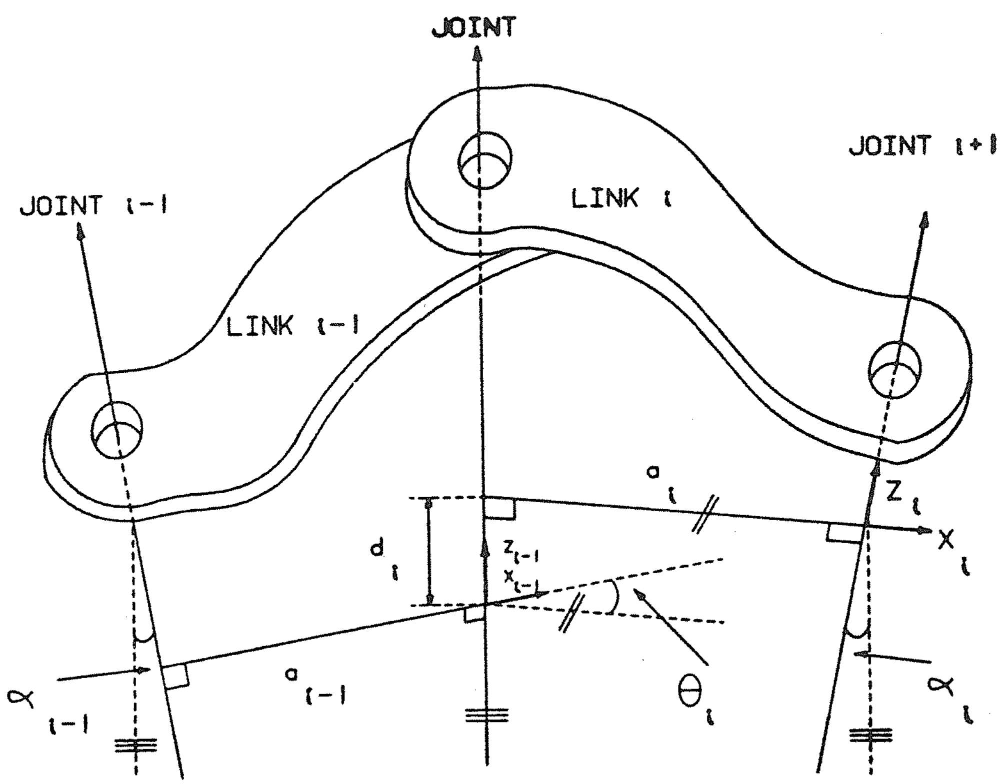

Fig. 2. Generalized manipulators link representation

nisms with two or three degrees-of-freedom, the complexity of the dynamic model is of minor importance in the computations, but for advanced control and design of typical industrial robots with more than three degrees-of-freedom, the development of an efficient mathematical representation of the manipulator dynamics is essential. This complication is due to the dynamic coupling among the joints and the nonlinearity arising from the continual variations in the configuration and payloads of the manipulator arm. However, it is important to note that on-line computations are normally required for the real-time control of robot manipulators. And furthermore, since the forces and torques should be updated frequently at a sampling rate of about 60 hertz, computational efficiency is of a major concern in the formulation of the dynamical equations. From the point of view of computer simulation, the dynamics equations derived by classical dynamic methods are too slow for real-time computation. Consequently due to the importance of the subject, the dynamics problem of manipulator arms has attracted considerable research and several techniques have evolved.

In recent years, several computational algorithms for generating the manipulators equations of motion for the sake of control and simulation have been developed (Fig. 1). Most of which are based on either Lagrange's equation [1–9] or the Newton-Euler

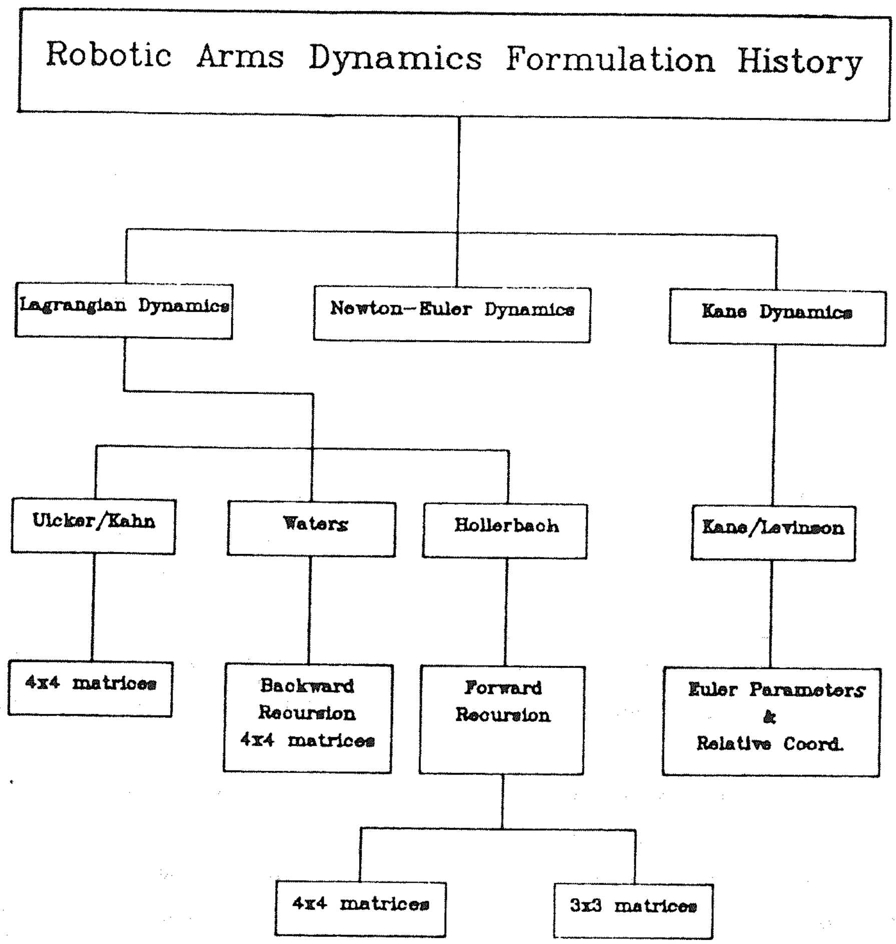

Fig. 1. Robotic arms dynamics formulation history

REVIEW

Dynamics Simulation Algorithms of Industrial Robots

A. Meghdari, Ph.D.

Mech. Eng. Dept. Sharif Univ. of Tech.

ABSTRACT

Presented are an investigation of mathematical algorithms pertaining to formulation of the governing dynamical equations of motion for industrial robots. Formulations based upon Lagrangian Dynamics, Newton - Euler Dynamics, and Kane Dynamics are specifically discussed in accordance to their computational complexities and the number of arithmetic operations involved.

1. INTRODUCTION

The efficient mathematical formulation of dynamic equations is of utmost importance in the analysis and control of robot manipulators. Hence, the first step one takes in analysing, designing, or identifying a robotic mechanism is to derive the system's equations of motion, because robot dynamics are concerned with the relationship between the motions of the mechanical kinematic chain of linkages and the forces or torques applied by its actuators.

In general, there exists two problems related to the dynamics of a manipulator arm. The first problem can be stated as evaluating the resulting motion of the manipulator, θ, $\dot{\theta}$, and $\ddot{\theta}$ (i.e. Position, Velocity, and Acceleration) under the application of a set of joint torques τ, which is obviously useful for simulating the mechanism. In robotics, this is refered to as the Forward Dynamics Problem. The second problem is to compute the joint torques required to produce given joint positions, velocities, and accelerations. The latter case named as Inverse Dynamics Problem is useful for the purpose of real-time control of the robotic arm which in turn is a computationally complex task. For mecha-