Proceedings of the 11th World Congress in Mechanism and Machine Science
April 1-4, 2004, Tianjin, China
China Machine Press, edited by Tian Huang

A Combined Penalty and Semi-Recursive Formulation for Closed-Loops in MBS

J. Cuadrado and D. Dopico

Dept. of Mechanical Engineering, University of La Coruña, Ferrol 15403, Spain
e-mail: javicuad@cdf.udc.es

Abstract: The continuously improved performance of personal computers enables the real-time motion simulation of complex multi-body systems, such as the whole model of an automobile, on a conventional $1,200 PC, provided the adequate formulation is applied. There exist two big families of dynamic formulations, depending on the type of coordinates they use to model the system: global and topological. The former leads to a simple and systematic programming while the latter is very efficient. In this work, a hybrid formulation is presented, obtained by combination of one of the most efficient global formulation and one of the most systematic topological formulation. In this way, it is developed a new formulation which shows, at the same time, easiness of implementation and a high level of efficiency. In order to verify the advantages that the new formulation has over its predecessors, the analysis of three examples is carried out using the three formulations and the corresponding results are compared: a planar mechanism which goes through a singular position, a car suspension with stiff behaviour, and a 6-dof robot with changing configurations.

Keywords: Multi-body systems, Real-time dynamics, Penalty formulations, Semi-recursive formulations

1 Introduction

Some years ago, the dynamic simulation of complex multibody systems in real-time was an objective difficult to achieve. Not only the fastest formulations available had to be applied, but also powerful and expensive hardware platforms (over $30,000) were needed. Nowadays, thanks to the enormous improvement experimented by the PC performance, either in calculation as well as in graphics, such complex multibody systems as, for example, the full model of a car, can be simulated in real-time on low cost PCs ($1,200). However, the field of multibody dynamics incessantly evolves and, if some years ago the objective was to simulate the motion of a mechanical system consisting of several rigid bodies, today the goal has been put farther, and flexible bodies as well as contact and impact effects should be considered, to mention just some examples. Hence, new super-efficient dynamic algorithms are required, capable of reducing at the minimum the calculation time needed, so that those simulations of complex systems modeled in a realistic way can be achievable.

Methods developed so far for the dynamic analysis of multibody systems can be grouped into two big families: global and topological.

Global methods[1,2] are characterized by the use of a set of coordinates that perfectly defines the position of each body. Due to this fact, the proper dynamic terms (applied and inertia forces) can be independently calculated for each body and, later on, be assembled to form the corresponding terms of the whole mechanism. On the other hand, the kinematic terms (constraint equations which relate the variables) are established in a systematic way for each body and kinematic pair. Consequently, this family of methods leads to simple and general algorithms of easy implementation but not very efficient.

Topological methods[1,2] make use of relative coordinates in order to model the mechanism, so that the position of each body is defined with respect to the previous one in the kinematic chain. This fact invites to take profit of the chain topology to produce algorithms in which the kinematic as well as the dynamic terms are calculated by means of efficient recursive procedures. However, these kind of formulations are usually rather involved and difficult to generalize.

This work is aimed at obtaining a hybrid formulation as combination of one global and another topological, so that the advantages of both types of formulations are kept, while their inherent drawbacks avoided.

2 Starting Formulations

Unlike the usual cases described above, two formulations, one global and another topological, have been developed recently that, possessing the advantages of their respective families, avoid the most part of the corresponding drawbacks.

The global method[3] uses natural (global and dependent) coordinates to model the multibody system. It consists of an index-3 augmented Lagrangian formulation, which is combined with the numerical integrator known as the trapezoidal rule, to produce a non-linear algebraic system of equations with the dependent positions as unknowns. Such system is solved through the Newton-Raphson iteration. Once convergence is attained into the time-step, velocities and accelerations are cleaned by means of mass-damping-stiffness-orthogonal projections. The result is a robust and efficient algorithm.

The topological method, called semi-recursive[4], defines a double set of coordinates in the modeling: six coordinates (three translations plus three rotations) for each body, and the relative coordinates of the whole mechanism. The dynamic equations are expressed in the coordinates of the bodies and, then, a velocity projection is carried out which leads to a set of motion equations in the relative coordinates. In order to calculate the leading matrix and the right-hand-side of that set of equations, a recursive technique which accumulates forces and inertias is used. However, as it happens with any topological method, closed-loops should be opened and, later on, the corresponding constraints imposed. Is in this step where the semi-recursive method finds problems, as it chooses to perform a second velocity projection (in order to arrive at a

set of motion equations in independent coordinates), which suffers from the usual drawbacks of this technique: range of validity of the independent set of coordinates selected and lack of robustness in singular positions. To avoid differences coming from the integration scheme, the same procedure as for the global formulation is used: the integrator (trapezoidal rule) equations are combined with the motion equations, thus obtaining a non-linear algebraic system of equations where the independent positions are the unknowns, which is solved by means of the Newton-Raphson iteration. The result is an easy and general algorithm.

3 The Proposed Formulation

In the proposed approach, the dynamic equations are stated according to the index-3 augmented Lagrangian formulation[3] in the form,

$$M\ddot{z} + \Phi_z^t \alpha \Phi + \Phi_z^t \lambda^* = Q \qquad (1)$$

where z are the relative coordinates, M is the mass matrix of the mechanism expressed in terms of the relative coordinates, Φ is the constraints vector due to the closure conditions of the loops, Φ_z is the Jacobian matrix of the constraints, α is the penalty factor, Q is the vector of applied and velocity-dependent forces, and λ^* is the vector of Lagrange multipliers obtained from the following iteration process (given by sub-index i, while sub-index n stands for the time-step):

$$\lambda_{i+1}^* = \lambda_i^* + \alpha \Phi_{i+1} \ , \quad i = 0,1,2,... \qquad (2)$$

where the value of λ_0^* is taken equal to the λ^* worked out in the previous time-step.

In order to determine the dynamic terms M and Q, a second set of coordinates is defined. It can be expressed at velocity level for each body of the system in the form,

$$Z = \begin{Bmatrix} \dot{s} \\ \omega \end{Bmatrix} \qquad (3)$$

being $\dot{s}$ the velocity of the point of the body which in that particular time is coincident with the fixed frame origin, and ω the angular velocity of the body.

When expressed in terms of such coordinates, the dynamic terms for a single body are[4],

$$\overline{M} = \begin{bmatrix} m I_3 & -m\tilde{g} \\ m\tilde{g} & J - m\tilde{g}\tilde{g} \end{bmatrix} \qquad (4)$$

$$\overline{Q} = \begin{Bmatrix} f - \omega \times (\omega \times mg) \\ n - \omega \times J\omega + g \times (f - \omega \times (\omega \times mg)) \end{Bmatrix} \qquad (5)$$

wherein m is the body mass, I_3 is the 3x3 identity matrix, g is the global position of the mass center of the body, $\tilde{g}$ is the dual anti-symmetric matrix of g, J is the inertia tensor of the body with respect to a reference frame parallel to the global one at the mass center of the body, f is the vector of forces applied to the body, and n is the vector of applied moments with respect to the mass center of the body.
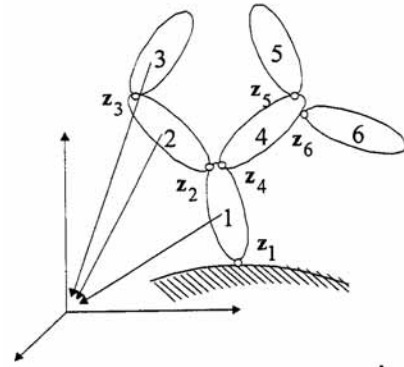

Figure 1 Example of mechanism topology

A matrix R can be defined so that the following relationship stands,

$$Z = R\dot{z} \qquad (6)$$

where now Z includes the body coordinates of all the bodies of the mechanism. Due to the body coordinates adopted, the form of matrix R is rather special, as shown in what follows for the example illustrated in Figure 1.

$$R = TH \qquad (7)$$

with,

$$T = \begin{bmatrix} I_6 & 0 & 0 & 0 & 0 & 0 \\ I_6 & I_6 & 0 & 0 & 0 & 0 \\ I_6 & I_6 & I_6 & 0 & 0 & 0 \\ I_6 & 0 & 0 & I_6 & 0 & 0 \\ I_6 & 0 & 0 & I_6 & I_6 & 0 \\ I_6 & 0 & 0 & I_6 & 0 & I_6 \end{bmatrix} \qquad (8)$$

and,

$$H = \begin{bmatrix} R_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_6 \end{bmatrix} \qquad (9)$$

where I_6 is the 6x6 identity matrix and R_i are 6x1 vectors.

If the virtual power principle is applied, the dynamic equations of the system can be written as,

$$Z^{*t}(\overline{M}\dot{Z} - \overline{Q}) = 0 \qquad (10)$$

and substituting in Equation (10) the result of Equation (6) together with its derivative yields

$$R^t \overline{M} R \ddot{z} = R^t (\overline{Q} - \overline{M}\dot{R}\dot{z}) \qquad (11)$$

which means that the mass matrix of the mechanism expressed in terms of the relative coordinates becomes,

$$M = R^t \overline{M} R \qquad (12)$$

and, analogously, the corresponding vector of applied and velocity-dependent forces is,

$$Q = R^t \left(\overline{Q} - \overline{M} R \dot{z} \right) \qquad (13)$$

If now the special form of matrix R described in Equations (7-9) is considered, M and Q can be rewritten as,

$$M = H^t \left(T^t \overline{M} T \right) H \qquad (14)$$

$$Q = H^t \left(T^t \left(\overline{Q} - \overline{M} T \dot{H} \dot{z} \right) \right) \qquad (15)$$

Due to the particular structure of matrices T and H, the mass matrix of Equation (14) and the force vector of Equation (15) can be calculated through a very efficient recursive procedure. Hence, for the example of Figure 1,

$$T^t M T = \begin{bmatrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 \\ & M_2 & M_3 & 0 & 0 & 0 \\ & & M_3 & 0 & 0 & 0 \\ & & & M_4 & M_5 & M_6 \\ & sym & & & M_5 & 0 \\ & & & & & M_6 \end{bmatrix} \qquad (16)$$

where the sub-matrices are obtained as,

$$\begin{aligned} M_6 &= \overline{M}_6 \\ M_5 &= \overline{M}_5 \\ M_3 &= \overline{M}_3 \\ M_2 &= \overline{M}_2 + M_3 \\ M_4 &= \overline{M}_4 + M_5 + M_6 \\ M_1 &= \overline{M}_1 + M_2 + M_4 \end{aligned} \qquad (17)$$

and,

$$T^t \left(\overline{Q} - \overline{M} T \dot{H} \dot{z} \right) = \begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{Bmatrix} \qquad (18)$$

where the sub-vectors are obtained as,

$$\begin{aligned} Q_6 &= \overline{Q}_6 - \overline{M}_6 \left(T \dot{H} \dot{z} \right)_6 \\ Q_5 &= \overline{Q}_5 - \overline{M}_5 \left(T \dot{H} \dot{z} \right)_5 \\ Q_4 &= \overline{Q}_4 - \overline{M}_4 \left(T \dot{H} \dot{z} \right)_4 + Q_5 + Q_6 \\ Q_3 &= \overline{Q}_3 - \overline{M}_3 \left(T \dot{H} \dot{z} \right)_3 \\ Q_2 &= \overline{Q}_2 - \overline{M}_2 \left(T \dot{H} \dot{z} \right)_2 + Q_3 \\ Q_1 &= \overline{Q}_1 - \overline{M}_1 \left(T \dot{H} \dot{z} \right)_1 + Q_2 + Q_4 \end{aligned} \qquad (19)$$

So far, the calculation of the M and Q terms of Equation (1) has been addressed. The remaining term in that equation is the Jacobian matrix of the constraints Φ_z, which implies the differentiation of the constraints vector with respect to the relative coordinates z. This can be easily done by applying the chain differentiation rule as,

$$\Phi_z = \Phi_q q_z \qquad (20)$$

In the proposed method, natural coordinates[2] are used at the cut-points to impose the closure conditions of the loops. This means that the constraints are expressed in terms of such coordinates, named q in Equation (20). Therefore, the term Φ_q is the traditional Jacobian matrix of the constraints, when natural coordinates are used[2], while the term q_z simply represents the velocities of the natural coordinates q, when unit velocities are successively given to the relative coordinates z.

Once the calculation of all the terms appearing in Equation (1) has been explained, the main thread of the proposed formulation can be taken again.

As integration scheme, the implicit single-step trapezoidal rule has been adopted. The corresponding difference equations in velocities and accelerations are:

$$\dot{z}_{n+1} = \frac{2}{\Delta t} z_{n+1} + \hat{\dot{z}}_n \qquad (21)$$

$$\ddot{z}_{n+1} = \frac{4}{\Delta t^2} z_{n+1} + \hat{\ddot{z}}_n \qquad (22)$$

being Δt the time-step and,

$$\hat{\dot{z}}_n = -\left(\frac{2}{\Delta t} z_n + \dot{z}_n \right) \qquad (23)$$

$$\hat{\ddot{z}}_n = -\left(\frac{4}{\Delta t^2} z_n + \frac{4}{\Delta t} \dot{z}_n + \ddot{z}_n \right) \qquad (24)$$

Dynamic equilibrium can be established at time step $n+1$ by introducing the difference equations (21) and (22) into the equations of motion (1), leading to a non-linear system of algebraic equations, where the z_{n+1} are the unknowns,

$$f(z_{n+1}) = 0 \qquad (25)$$

Such system can be solved by the Newton-Raphson iteration, where the approximated tangent matrix is:

$$f_z = M + \frac{\Delta t}{2} C + \frac{\Delta t^2}{4} \left(\Phi_z^t \alpha \Phi_z + K \right) \qquad (26)$$

and the residual vector:

$$f = \frac{\Delta t^2}{4} \left(M\ddot{z} + \Phi_z^t \alpha \Phi + \Phi_z^t \lambda^* - Q \right) \qquad (27)$$

where C and K represent the contribution of damping and elastic forces of the system provided they exist.

The procedure explained above yields a set of positions z_{n+1} that not only satisfies the equations of motion (1), but also the constraint conditions $\Phi = 0$. However, it is not expected that the corresponding sets of velocities and accelerations satisfy $\dot{\Phi} = 0$ and $\ddot{\Phi} = 0$, because these conditions have not been imposed in the solution process. To overcome this difficulty, mass-damping-stiffness -orthogonal projections in velocities and accelerations are performed. It can be seen that the projections leading matrix is the same tangent matrix appearing in equation (26). Therefore, triangularization is avoided and projections in velocities and accelerations are carried out with just forward reductions and back substitutions.

If $\dot{z}^*$ and $\ddot{z}^*$ are the velocities and accelerations obtained after convergence has been achieved in the Newton-Raphson iteration, their cleaned counterparts $\dot{z}$ and $\ddot{z}$ are calculated from,

$$f_z \dot{z} = \left[M + \frac{\Delta t}{2} C + \frac{\Delta t^2}{4} K \right] \dot{z}^* - \frac{\Delta t^2}{4} \Phi_z^t \alpha \Phi_t \quad (28)$$

for the velocities, and,

$$f_z \ddot{z} = \left[M + \frac{\Delta t}{2} C + \frac{\Delta t^2}{4} K \right] \ddot{z}^* - \frac{\Delta t^2}{4} \Phi_z^t \alpha \left(\dot{\Phi}_z \dot{z} + \dot{\Phi}_t \right) (29)$$

for the accelerations.

4 Examples
To show the advantages of the new proposed formulation, the following three examples have been solved through three approaches: the new formulation and the two starting ones. The implementation of all the cases has been carried out in the computing environment Matlab, which means that the reported CPU times would be drastically reduced if the same programs were written in Fortran or C languages.

4.1 Planar mechanism
The first example is a one degree-of-freedom assembly of two four-bar linkages, illustrated in Figure 2, which has been proposed[5] as an example to test the performance in cases where the mechanism undergoes singular configurations. When the mechanism reaches a horizontal position, the number of degrees of freedom increases instantaneously from 1 to 3. All the links have a uniformly distributed unit mass and a unit length. The gravity force acts in the negative vertical direction. At the beginning, the links pinned to the ground are in vertical position and upwards, receiving a unit clockwise angular velocity.
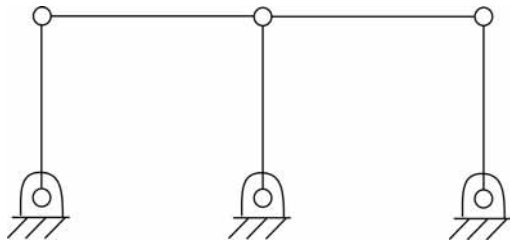

Figure 2 Double four-bar mechanism

The simulation lasts for 10 s, during which the mechanism goes through the singular position ten times. Table 1 shows the obtained results by using the three methods for different time-steps (Δt). The error is measured as the maximum deviation, in J, suffered by the system energy. The time is the CPU time, in s, required to carry out the simulation.

Table 1 Results for the double four-bar mechanism

	Global		Topological		Hybrid	
Δt	Error	Time	Error	Time	Error	Time
0.01	-0.21	0.86	no convergent		-0.01	3.20
0.03	-0.83	0.37	no convergent		-0.12	1.32
0.05	-9.03	0.26	no convergent		-0.34	0.83
0.1	wrong results		no convergent		-1.34	0.52

First of all, it can be seen that the topological method cannot accomplish the integration. It is due to the second velocity projection, which cannot be performed at the singularity. It can also be observed that the global method is more efficient than the hybrid one for the same time-step, while the hybrid method is found to be more accurate and robust, as it shows an smaller error for the same time-step and, moreover, works for larger time-steps than the global one. Therefore, if the efficiency of both methods is compared for the same level of error, similar performances would be obtained. It should be noted that, in this example, the number of global coordinates is 6, while the number of relative coordinates is 5. It is not then a case in which the hybrid method can get a great advantage due to the reduced number of coordinates with respect to the global method (this will occur in large examples).

4.2 Car suspension
The second example, shown in Figure 3, is the suspension system of the Iltis vehicle[6], which was proposed as a benchmark problem by the European automobile industry to check multibody dynamic codes.
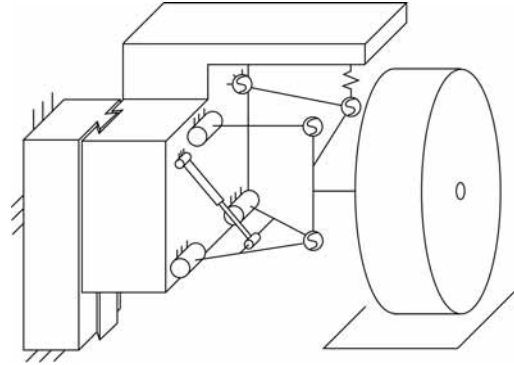

Figure 3 Iltis suspension

The suspension starts moving from at rest conditions at a speed of 5 m/s, and its position does not correspond to the static equilibrium. Thus, it freely oscillates until the static equilibrium position is reached. After three seconds, the suspension drops down a kerb of 10 cm height, and afterwards oscillates until the equilibrium is reached again. The complete analysis lasts for 7 s. It is carried out using the three methods, the performance results being depicted in Table 2. This time the error is obtained as the difference, in J, between the lost gravitational energy and the energy dissipated by the shock-absorber.

Table 2 Results for the Iltis suspension

	Global		Topological		Hybrid	
Δt	Error	Time	Error	Time	Error	Time
0.001	-43.13	17.8	-43.49	45.7	-42.05	43.2
0.01	-43.45	2.04	-44.06	6.04	-42.80	5.44
0.03	-57.96	0.84	no convergent		no convergent	

In this case, unlike the previous one, the system shows a stiff behaviour, since the block representing one fourth of the whole vehicle has great inertia and, therefore, moves at slow frequency, while the wheel possess little mass and, subsequently, moves at high frequency. Firstly, it can be seen that the topological method is the least efficient and robust. On the other hand, the global method is less

accurate but more robust than the hybrid one, since it works for larger time-steps. Regarding efficiency, the global method keeps the same advantage with respect to the hybrid one than in the previous example, since 25 global coordinates have been used face to 8 relative coordinates. It is supossed that many more variables are needed, so that the global method requires a huge effort to solve the equation system, thus balancing its advantage.

4.3 Serial robot

The PUMA robot, designed by Unimation & Co. and shown in Figure 4, is an example of 6 degrees-of-freedom serial manipulator. It has been often used by different authors[7] to illustrate methods and procedures in several areas of robotics. In this work, the robot has been taken as an example of multibody system undergoing changing configurations.
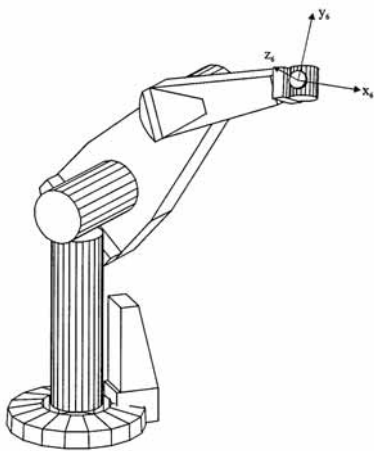

Figure 4 PUMA robot

Starting from rest, torques at the six hinges of the robot are provided so that, in a time of 2 s, it arrives at a new position in the space, again in rest conditions. Once the new position has been reached, a point of the hand is attached to the ground, so that the robot loses 3 degrees-of-freedom. In this new configuration, torques are applied to the three rotational pairs of the hand, and a second manoeuvre, which lasts 4 s and ends with rest conditions, is performed. Finally, the robot is released from its attachment and returned in 2 s to the initial position by torques acting at the six revolute joints, once more finishing the manoeuvre at rest conditions. Therefore, the total simulation time is 8 s. Table 3 illustrates the results obtained when applying the three methods, where the error has been calculated as the distance between the hand positions at initial and final times (which, ideally, should be coincident).

Table 3 Results for the PUMA robot

Δt	Global		Topological		Hybrid	
	Error	Time	Error	Time	Error	Time
0.01	0.53	6.25	0.26	3.05	0.34	4.02
0.05	3.4	1.92	1.5	1.20	1.8	1.74
0.1	no convergent		3.2	0.78	no convergent	

This time, the global method shows the worst behaviour either in efficiency, accuracy or robustness, probably because the number of global coordinates defined to model the robot is 45 (sparse matrix techniques have not been used in this case), far from the 6 corresponding relative coordinates needed. Comparing the topological and the hybrid methods, some advantage is attained by the former with respect to the latter. It must be pointed out that the integration scheme used for the three methods is a single-step one. This fact favours the topological method, since every time the configuration changes, also does the number of integrable variables when using that method (it is not the case with the two others). If a multi-step integrator had been used, a considerable extra-effort would have been needed when applying the topological method, in order to restart the integration whenever a change in the configuration had place.

5 Conclusions

Based on the results obtained for the three studied examples, the conclusions can be drawn as follows:

(1) A new real-time formulation for the dynamics of multibody systems has been presented, which encompasses high ranks of efficiency, accuracy, robustness and easiness of implementation.

(2) The method combines a semi-recursive formulation based on velocities projection and a penalty formulation for closed-loops consideration.

(3) The method is more robust that its semi-recursive predecessor, as it can handle singular positions.

(4) The method is more efficient that its global ancestor for a similar level of accuracy when the number of global coordinates becomes large.

References

1. Shabana, A.A., Dynamics of multibody systems, Cambridge University Press, Cambridge, 1998
2. García de Jalón, J., Bayo, E., Kinematic and dynamic simulation of multibody systems, Springer-Verlag, New York, 1994
3. Cuadrado, J., Gutiérrez, R., Naya, M.A., Morer, P., A comparison in terms of accuracy and efficiency between a MBS dynamic formulation with stress analysis and a non-linear FEA code, *Int. J. for Numerical Methods in Engineering*, 51(9): 1033-1052, 2001
4. Rodríguez, J.I., Analisis eficiente de mecanismos 3D con metodos topologicos y tecnologia de componentes en Internet, Ph. D. Thesis, University of Navarre, Spain, 2000
5. Bayo, E., Avello, A., Singularity free augmented Lagrangian algorithms for constraint multibody dynamics, *Nonlinear Dynamics*, 5: 209-231, 1994
6. Iltis Data Package, *IAVSD Workshop*, Herbertov, Czechoslovakia, 1990
7. Angeles, J., Ma, O., Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement, *Int. J. of Robotic Research*, 7(5): 32-47, 1988