EFFICIENT COMPUTATION OF THE GENERALIZED INERTIAL TENSOR OF ROBOTS BY USING THE GIBBS-APPELL EQUATIONS

Provenzano, S. (*), Mata, V.(**), Ceccarelli, M.(***) and Suñer, J.L. (**)

(*) Escuela de Ingeniería Mecánica
Universidad de Los Andes (VENEZUELA)

(**) Departamento de Ingeniería Mecánica y de Materiales
Universidad Politécnica de Valencia (SPAIN)

(***) Dipartimento di Meccanica, Strutture, Ambiente e Territorio
Università di Cassino (ITALY)

Abstract

In this paper, a method for computing the generalized inertial tensor of manipulator robots is described as based on the Gibbs-Appell equations. The method presents the formulation of the coefficients of the mentioned tensor by means of the Hessian of the Gibbs function. An efficient recursive algorithm is proposed, with a complexity of $O(n^2)$ order, which is used to produce a program in FORTRAN. A numerical example simulates the movement of a PUMA robot. The results obtained from the simulation are compared with those obtained by using other known methods.

1. - Introduction

The direct dynamic problem or simulation problem of robots is a process that involves the computing of accelerations, velocities and orientations or joint positions for a manipulator, when the forces exerted by the actuators are given.

The simulation problem for mechanical systems consists of two parts: the first one requires the calculation of joint accelerations from the movement equations; and the second one involves the integration of those equations. The joint accelerations can be obtained by two different approaches. In the first one a system of differential equations is proposed, obtained, and solved. In the second approach, the accelerations are calculated recursively by propagating the movement and constraint forces throughout the mechanism.

The algorithms derived from the first approach have usually a computational complexity of $O(n^3)$ order, mainly due to the fact that a linear system must be solved. Papers [1] and [2] are examples of this approach. On the other hand, the algorithms derived using the second approach have a computational complexity of $O(n)$ order, although the coefficients that appears are large (for example, [3-6]). These algorithms are superior than the first ones but only for systems with more than 9 link, as shown in [7]. The best known approach is the articulated body (ABM) [4].

The algorithms based on the first approach follow the procedure that has been developed by Walker y Orin [1], which requires the generalized inertial tensor and the so-called bias vector. In Walker and Orin´s work, they obtain the bias vector by means of an algorithm that solves the inverse dynamic problem, in which the joint accelerations became null. The claculation of the generalized inertial tensor is then the main problem for the application of this approach. Walker and Orin use the composed rigid body method (CRBM), which is based on the Newton-Euler equations and has a complexity of $O(n^2)$. An alternative method has been developed by Angeles and Ma [2], who use the natural orthogonal complement to obtain an efficient algorithm, but, of $O(n^3)$ order.

These methods have been studied and improved over time (see for example [7], [8]), even by using the spatial algebra as in [9]. It has been demonstrated that the methods ABM y CRBM, are different forms of solving the same linear system. This fact is shown in several papers, such as [10] and [11].

In this paper a new algorithm is proposed as based on the Gibbs-Appell equations in agreement with [12], in order to compute the generalized inertial tensor. The use of the Gibbs-Appell equations in the dynamics of multibody systems has been limited, in the last twenty five years, to few examples that followed mainly a closed form formulation (see [13]), so that a great number of arithmetic operations were required. The first recursive development of $O(n^2)$ order was proposed to solve the inverse dynamic problem by Desoyer and Lugner [14], who used the Jacobian matrix of the manipulator in order to avoid the explicit development of partial derivatives. Later Rudas and Toth [15], presented a recursive algorithm of $O(n)$ order, likewise, to solve the inverse problem. This algorithm is based on the minimisation of the Gibbs function by means of Lagrange´s multipliers. Recently Mata et al. [16] have solved the inverse dynamic problem for robotic manipulator and introduced an $O(n)$ efficient recursive algorithm, which takes advantage of the possibilities that the Gibbs-Appell equations offer to compute the generalized forces on robotic manipulators.

In the next section, we will start from the rigid body´s Gibbs function to arrive to an expression that permits to develop of efficient recursive relations which are useful to compute the terms of the generalized inertial tensor.

In the third section, a Gibbs-Appell Hessian (GAH) algorithm is proposed as based on that expression. In the forth section an example is shown, in which the results obtained from simulating the movement of a PUMA robot are compared with those obtained from ABM, CRBM and the GAH methods.

2.- Generalized inertial tensor by using the Hessian of the Gibbs function

The Gibbs function, G_i, for a rigid body i that is part of a mechanical system, can be expressed as [12]

$$G_i = \frac{1}{2} m_i \left({}^i\ddot{\vec{r}}_{G_i} \right)^T {}^i\ddot{\vec{r}}_{G_i} + \frac{1}{2} \left({}^i\dot{\vec{w}}_i \right)^T {}^i\mathbf{I}_{G_i} \cdot {}^i\dot{\vec{w}}_i + \left({}^i\dot{\vec{w}}_i \right)^T \left({}^i\vec{w}_i \times {}^i\mathbf{I}_{G_i} {}^i\vec{w}_i \right) \qquad (1)$$

where m_i is the mass of i-th link, ${}^i\ddot{\vec{r}}_{G_i}$ the acceleration of the centre of mass of i-th link, expressed in the local reference system, ${}^i\vec{w}_i$ and ${}^i\dot{\vec{w}}_i$ the angular velocity and acceleration of i-th link, both expressed in the local reference system and ${}^i\mathbf{I}_{G_i}$ the inertia tensor of link i-th, expressed in its own reference system.

On the other hand, the whole Gibbs function for a mechanical system with n bodies is given by

$$G = \sum_{i=1}^{n} G_i \qquad (2)$$

Considering the expressions (1) and (2) the terms of D, which is the generalized inertial tensor, will be given by the Hessian matrix of the Gibbs function in the form

$$D_{jk} = \frac{\partial^2 G}{\partial \ddot{q}_j \, \partial \ddot{q}_k} \qquad (3)$$

where $\ddot{q}_j$ is the generalized acceleration of j-th joint.

From the previous expression, a simple algorithm of $O(n^3)$ order can be obtained for to the computation of D, although is possible to reduce it to a recursive algorithm of $O(n^2)$ order. A procedure can be used to find the expression that permits to calculate the terms of the generalized inertial tensor through successive partial derivatives of the Gibbs function. It can be further simplified if the following expressions are taken into account, when the application of the Denavit-Hartenberg´s modified notation is used [2]

- $\frac{\partial}{\partial \ddot{q}_j} \frac{\partial {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} = \frac{\partial}{\partial \ddot{q}_j} {}^i\vec{z}_i = 0$
- $\frac{\partial}{\partial \ddot{q}_j} \frac{\partial {}^i\ddot{\vec{r}}_{o_i}}{\partial \ddot{q}_k} = \frac{\partial}{\partial \ddot{q}_j} {}^i\mathbf{R}_{i-1} \left[\frac{\partial {}^{i-1}\ddot{\vec{r}}_{o_{i-1}}}{\partial \ddot{q}_k} + \frac{\partial {}^{i-1}\dot{\vec{w}}_{i-1}}{\partial \ddot{q}_k} \times {}^{i-1}\vec{r}_{O_{i-1},O_i} \right] = 0$
- $\frac{\partial}{\partial \ddot{q}_j} \frac{\partial {}^i\ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_k} = \frac{\partial}{\partial \ddot{q}_j} \left(\frac{\partial {}^i\ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_k} + \frac{\partial {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \times {}^i\vec{r}_{O_i,G_i} \right) = 0$

in which ${}^i\vec{z}_i$ is the unit vector along the Z axis, ${}^{i-1}\ddot{\vec{r}}_{o_{i-1}}$ the acceleration of the origin of the (*i-1*)-th reference system, ${}^i\mathbf{R}_{i-1}$ the rotation matrix between two adjacent reference systems, ${}^{i-1}\vec{r}_{O_{i-1},O_i}$ is the vector from the origin of the (*i-1*)-th reference system to the origin of the i-th reference system, ${}^i\vec{r}_{O_i,G_i}$ is the vector from the origin of the i-th reference system to the centre of mass of the i-th link. The superscript denotes the reference system on which the vectors are expressed.

Thus, the expression for the terms of the generalized inertial tensor looks like

$$D_{jk} = \sum_{i=j}^{n} \left[m_i \left(\frac{\partial\, {}^i\ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_k} \right)^T \frac{\partial\, {}^i\ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_j} + \left(\frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \right)^T {}^i\mathbf{I}_{G_i} \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_j} \right] \tag{4}$$

The utilisation of this last expression can elaborate a computationally efficient algorithm applicable to manipulators, but it needs the study of the recurrent relations that among its elements, since the direct application of the expression would produce an algorithm with a high number of operations. The following expressions are useful for a recursive calculation of the expression (4):

$$\frac{\partial\, {}^i\vec{r}_{G_i}}{\partial \ddot{q}_k} = \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \times {}^i\vec{r}_{O_k,G_i}\,, \quad \frac{\partial\, {}^k\dot{\vec{w}}_k}{\partial \ddot{q}_k} = {}^k\mathbf{R}_i \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k}$$

In addition, replacing the vectorial product by the product of an antisymetric tensor and a vector. The following expression can be obtained

$$\left(\frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \times {}^i\vec{r}_{O_k,G_i} \right)^T = \left(\frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \right)^T \cdot {}^i\tilde{r}_{O_k,G_i}\,, \quad {}^i\vec{r}_{O_k,G_i} \times \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} = {}^i\tilde{r}_{O_k,G_i} \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k}$$

Applying the properties of the vectorial product, the equation **Errore. L'argomento parametro è sconosciuto.** can be written as

$$D_{jk} = \sum_{i=j}^{n} \left[-m_i \left({}^i\vec{r}_{O_k,G_i} \times \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \right)^T \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_j} \times {}^i\vec{r}_{O_j,G_i} + \left(\frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_k} \right)^T {}^i\mathbf{I}_{G_i} \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_j} \right]$$

By using the antisymetric tensor as previously defined, the expression takes the form

$$D_{jk} = \sum_{i=j}^{n} \left[\left({}^i\mathbf{R}_k \frac{\partial\, {}^k\dot{\vec{w}}_k}{\partial \ddot{q}_k} \right)^T \left({}^i\mathbf{I}_{G_i} - m_i\, {}^i\tilde{r}_{O_i,G_i} \cdot {}^i\tilde{r}_{O_j,G_i} - m_i\, {}^i\tilde{r}_{O_k,O_i} \cdot {}^i\tilde{r}_{O_j,G_i} \right) \frac{\partial\, {}^i\dot{\vec{w}}_i}{\partial \ddot{q}_j} \right] \tag{5}$$

3.- GAH Algorithm

A recursive algorithm of $O(n^2)$ order can be formularted as, based on the expression (5) in order to compute the generalized inertia tensor for a robotic manipulator. In particular, the computation can be provided by means of a step by step procedure as in the following.

Step 1

The compound mass, the compound inertial tensor and other quantities that do not vary with the time, can be calculated off-line by means of the following items, once $M_n = m_n$ is given

For $i = 1$ to $n - 1$, do

$M_i = m_i + M_{i+1}$ 0M, 0A (6)

${}^{i}\vec{g}_i = m_i\, {}^{i}\vec{r}_{O_i,G_i} + M_{i+1}\, {}^{i}\vec{r}_{O_i,O_{i+1}}$ 0M, 0A (7)

For i = 1 to n - 1, do

${}^{i}\mathbf{I}_{O_i} = {}^{i}\mathbf{I}_{G_i} - m_i\, {}^{i}\tilde{r}_{O_i,G_i}\, {}^{i}\tilde{r}_{O_i,G_i}$ 0M, 0A (8)

${}^{i}\mathbf{E}_i = {}^{i}\mathbf{I}_{O_i} - M_{i+1}\, {}^{i}\tilde{r}_{O_i,O_{i+1}}\, {}^{i}\tilde{r}_{O_i,O_{i+1}}$ 0M, 0A (9)

Step 2

Vectors ${}^{j}\vec{r}_{O_i,O_j}$ can be computed by

For i = 1 to $n - 1$,

For j = i + 1 to n, do

${}^{j}\vec{r}_{O_i,O_{i+1}} = {}^{j}\mathbf{R}_{j-1}\, {}^{j-1}\vec{r}_{O_i,O_{i+1}}$ 8M, 4A (10)

For i = $n - 2$ to 1,

For j = n to i + 2, do

${}^{j}\vec{r}_{O_i,O_j} = {}^{j}\vec{r}_{O_i,O_{i+1}} + {}^{j}\vec{r}_{O_{i+1},O_j}$ 0M, 3A (11)

Step 3

Vectors $\frac{\partial\, {}^{j}\dot{\vec{w}}_j}{\partial \ddot{q}_k}$ .can be evaluated by the recursive computations in the form

For i = 1 to n, set

$\frac{\partial\, {}^{j}\dot{\vec{w}}_j}{\partial \ddot{q}_j} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ 0M, 0A (12)

For i = 2 to n, set

$\frac{\partial\, {}^{j}\dot{\vec{w}}_j}{\partial \ddot{q}_{j-1}} = \begin{bmatrix}{}^{j}R_{j-1}^{(1,3)} & {}^{j}R_{j-1}^{(2,3)} & {}^{j}R_{j-1}^{(3,3)}\end{bmatrix}^T$ 0M, 0A (13)

For i = 1 to $n - 1$,

For j = i + 2 to n, do

$\frac{\partial\, {}^{j}\dot{\vec{w}}_j}{\partial \ddot{q}_i} = {}^{i}\mathbf{R}_{i-1} \frac{\partial\, {}^{j-1}\dot{\vec{w}}_{j-1}}{\partial \ddot{q}_i}$ 8M, 4A (14)

Step 4

Vectors $^{i}\vec{f}_{i}$ y $^{i-1}\vec{f}_{i}$ (the product $m_{i}\,{}^{i}\vec{r}_{O_i,G_i}$ can be obtained off-line) can be obtained, once $^{n}\vec{f}_{n} = m_{n}\,{}^{n}\vec{r}_{O_n,G_n}$ is given , by means of the computation

For $i = n - 1$ to 1, do

$$^{i}\vec{f}_{i+1} = {}^{i}\mathbf{R}_{i+1}\,{}^{i+1}\vec{f}_{i+1} \qquad \text{8M, 4A (15)}$$

$$^{i}\vec{f}_{i} = {}^{i}\vec{g}_{i} + {}^{i}\vec{f}_{i+1} \qquad \text{0M, 3A (16)}$$

Step 5

For the computation of Tensors $^{i}\boldsymbol{\o}_{i}$. The resemblance transformation is calculated in an efficient way by doing the decomposition of the product $^{i}\mathbf{R}_{i+1}\,{}^{i+1}\boldsymbol{\o}_{i+1}\,{}^{i+1}\mathbf{R}_{i}$, once $^{n}\boldsymbol{\o}_{n} = {}^{n}\mathbf{I}_{O_n}$ is given, and the items are calculated as

For $i = n - 1$ to 2, do

$$^{i}\boldsymbol{\o}_{i} = {}^{i}\mathbf{E}_{i} - {}^{i}\tilde{r}_{O_i,O_{i+1}}\,{}^{i}\vec{f}_{i+1} - \left({}^{i}\tilde{r}_{O_i,O_{i+1}}\,{}^{i}\vec{f}_{i+1}\right)^{T} + {}^{i}\mathbf{R}_{i+1}\,{}^{i+1}\boldsymbol{\o}_{i+1}\,{}^{i+1}\mathbf{R}_{i} \qquad \text{31M, 40A (17)}$$

It is to note that For $i = 1$ only the element $^{1}\boldsymbol{\o}_{1}^{(3,3)}$, is necessary 13M, 11A (18)

Step 6

The elements of principal diagonal of the generalized inertial tensor can be computed by

For $i = 1$ to n, do

$$D_{ii} = \left(\frac{\partial\,{}^{i}\dot{\vec{w}}_{i}}{\partial \ddot{q}_{i}}\right)^{T} {}^{i}\boldsymbol{\o}_{i}\,\frac{\partial\,{}^{i}\dot{\vec{w}}_{i}}{\partial \ddot{q}_{i}} \qquad \text{0M, 0A (19)}$$

Step 7

Computation of the rest of elements of the generalized inertial tensor

For $j = n$ to 2,

For $k = j - 1$ to 1, do

$$D_{jk} = \left(\frac{\partial\,{}^{j}\dot{\vec{w}}_{j}}{\partial \ddot{q}_{k}}\right)^{T} \left[{}^{j}\boldsymbol{\o}_{j} - {}^{j}\tilde{r}_{O_k,O_j} \cdot {}^{j}\tilde{f}_{j}\right] \frac{\partial\,{}^{j}\dot{\vec{w}}_{j}}{\partial \ddot{q}_{j}} \qquad \text{7M, 6A (20)}$$

Table 1.- Comparison of complexities

Authors	Method	Products M (n = 6)	Additions A(n =6)
Walker and Orin	CRBM	$12n^2 + 56n - 27$ (741)	$7n^2 + 67n - 53$ (601)
Angeles and Ma	Nat. Ortog. Comp.	$n^3 + 17n^2 - 21n + 8$ (710)	$n^3 + 14n^2 - 16n + 5$ (629)
This work	GAH	$11.5n^2 + 19.5n - 49$ (482)	$8.5n^2 + 31.5n - 69$ (426)

Once the generalized inertia tensor is determined, a numerical simulation of dynamic behaviour of a multibody system can be straightforward obtained.

4.- Example

In order to execute an example useful for comparison and validation, a simulation program has been written in FORTRAN as based on the methodology proposed by Walker and Orin [1], in which for the generalized inertial tensor that is calculated by using the proposed algorithm . In addition, the bias vector is calculated by using a modified version of the algorithm that solves the inverse dynamic problem by using the Gibbs-Appell equations, proposed in [16].

The program has been written in double precision and is executed on a computer PC Pentium II 400 MHz. The Gauss-Jordan elimination method has been used to solve the linear system, , and the system of differential equations is integrate by the Runge-Kutta´s five order technique. In the integration process, the tolerance is 1×10^{-6} and the time interval of 0.1 s. In this example we simulate the movement of the PUMA robot, whose terminal element describes a straight trajectory, with constant orientation and the velocity of 0.1 m/s. The simulation starts from the position ${}^{0}\vec{r}_{O_0,TCP}\big|_{t=0s} = \begin{bmatrix}0.60 & 0.175 & 0.250\end{bmatrix}^T$, the orientation (Z, Y, Z) of the local system fixed on the link 6 is $\begin{bmatrix}45 & 60 & 90\end{bmatrix}$ deg. After 5 seconds the movement lasts, the terminal element reaches the position given by ${}^{0}\vec{r}_{O_0,TCP}\big|_{t=5s} = \begin{bmatrix}0.244 & 0.527 & 0.250\end{bmatrix}^T$. Figure 1 shows the robot in the initial position and the planned trajectory. The inverse dynamic problem has been solved for each instant by using as input data the results of the resolution of the inverse kinematic problem, through the procedure that is outlined in the flowchartof Figure 2. The superscript "*" denotes the generalized coordinates, velocities and accelerations, obtained from the numerical integration of the proposed formulation.

In order to validate and compare the results, simulation codes are written in FORTRAN as based on the composed rigid body methods (CRBM) and of articulated body (ABM). Those codes have been executed under identical conditions also use for the code that is based on the algorithm GAH. The differences among the obtained results from the three compared methods are subtle, so that is possible to observe differences in generalized coordinates, velocities and accelerations, from the tenth decimal. In the other hand, if the results are expressed regarding to position, velocity and acceleration of the end-effector, the differences can be observed more clear , due to the accumulation errors produced in the different joints that compose the studied system. In the Table 2, the maximum errors with respect to the prescribed trajectory and the difference average are depicted which take place in the modules of the vector position, velocity and acceleration of the element terminal, obtained from the simulation with each one of the methods.

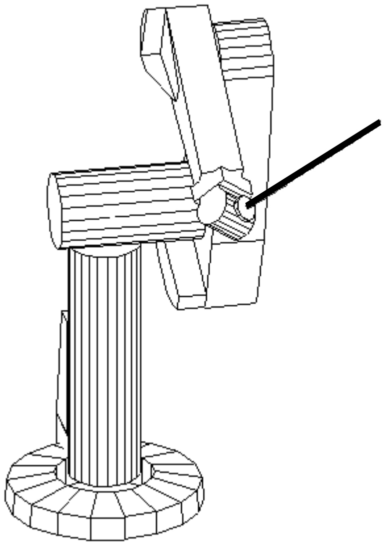

Figure 1.- Initial position of the robot and trajectory planned.

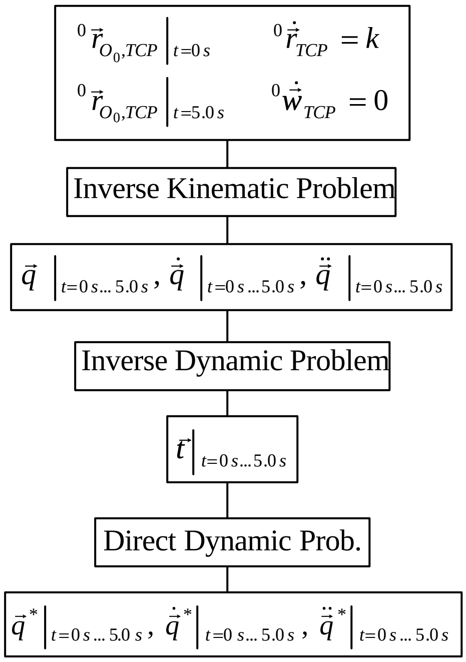

Figure 2.- Flow chart of the procedure for the numerical simulation

Table 2.- Maximum Errors and processing time in the simulation for each method.

		CRBM	ABM	GAH
$\Delta\left\|{}^{0}\vec{r}_{O_0,TCP}\right\|$	Max.	2.24416×10^{-3}	2.24506×10^{-3}	2.24446×10^{-3}
	Prom.	1.98474×10^{-4}	1.98552×10^{-4}	1.98500×10^{-4}
$\Delta\left\|{}^{0}\dot{\vec{r}}_{TCP}\right\|$	Max.	1.48353×10^{-3}	1.48402×10^{-3}	1.48370×10^{-3}
	Prom.	1.01924×10^{-4}	1.01960×10^{-4}	1.01937×10^{-4}
$\Delta\left\|{}^{0}\ddot{\vec{r}}_{TCP}\right\|$	Max.	1.96068×10^{-2}	1.96293×10^{-2}	1.96184×10^{-2}
	Prom.	1.73333×10^{-3}	1.73432×10^{-3}	1.73377×10^{-3}
Processing time (s.)		0.57	0.65	0.56

5.- Conclusion

In this paper, the Gibbs-Appell equations have been used for the formulation of a new recursive algorithm for the computation of the generalized inertial tensor. The algorithm is of $O(n^2)$ order and because of the arithmetic number of involved operations, it is computationally efficient, as shown in the Table 1. In the algorithm, the terms of the generalized inertial tensor are obtained by means of the Hessian matrix of the Gibbs function. This approach takes the advantage of the Gibbs equations for a novel suitable formulation of the generalized inertia tensor of multi-body systems, particularly robotic manipulators. A numerical example has been carried out in a PUMA robot using the proposed algorithm (GAH) and the algorithms better known of the literature (CRBM and ABM). The results of these simulations are shown in the Table 2, where it can be observed that the method GAH produces errors smaller than the CRBM method, but a little bit greater than the ABM, which is, according to several authors (see for example [11] and [7]), the method that has better numerical response. In addition from Table 2, the processing time of the GAH method is the smallest one, since the proposed algorithm has smaller computational complexity with respect to traditional the algorithms derived from the CRBM and ABM methods.

6.- References

[1] M. W. Walker, D.E. Orin, “Efficient Dynamic Computer Simulation of Robotic Mechanisms”, *Journal of Dynamic Systems, Measurement and Control*, vol 104, pp. 205-211, 1982.

[2] J. Angeles, O. Ma, “Dynamic Simulation of n-Axis Serial Robotic Manipulators Using a Natural Orthogonal Complement”, *The International Journal of Robotic Research*, vol. 7, no. 5, pp. 32-47, 1988.

[3] W. W. Armstrong, “Recursive solution to the equations of motion of an n-link manipulator”, *Proc. Fifth World Congress on Theory of Machines and Mechanisms*, Montreal, 1979.

[4] R. Featherstone, “The Calculation of Robot Dynamics using Articulated-Body Inertias”, *Int. J. Robotics Research*, vol. 2, no. 1, pp. 13-30, 1983.

[5] G. Rodriguez, “Kalman Filtering, Smoothing, and Recursive Robot Arm Forward and Inverse Dynamics”, *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 6, pp. 624-639, 1987.

[6] S. K. Saha, “A decomposition of the manipulator inertia matrix”, *IEEE Trans. on Robotics and Automation*, vol. 13, no. 2, 1997.

[7] R. Featherstone, D. Orin, “Robot Dynamics: Equations and Algorithms”, *Proc. Of the 2000 IEEE Int. Conf. On Robotics & Automation*, pp. 826-834, San Francisco, Abril 2000.

[8] C. A. Balafoutis, R. V. Patel, *Dynamic Analysis of Robot Manipulators: A Cartesian Tensor Approach*, Kluwer Academic Press, Boston, 1991.

[9] G. Rodriguez, A. Jain, K. Kreutz-Delgado, “A Spatial Operator Algebra for Manipulator Modelling and Control”, *Int. J. Robotics Research*, vol. 10, no. 4, pp. 371-381, 1991.

[10] D. K. Pai, U. M. Ascher, P. G. Kry, “Forward dynamics algorithms for multibody chains and contact”, *Proc. 2000 IEEE Int. Conf. on Robotics & Automation*, San Francisco, April 2000.

[11] U. M. Ascher, D. K. Pai, B. P. Cloutier, "Forward Dynamics Elimination Methods, And Formulation Stiffness in Robot Simulation", *The International Journal of Robotic Research*, vol. 16, no. 6, pp. 749-758, 1997.
[12] L. Pars, *A Treatise on Analytical Dynamics*, Ox Bow Press, Connecticut, 1972.
[13] M. Vukobratovic y N. Kircanski, *Real Time Dynamics of Manipulation Robots*. Springer-Verlag, Berlin, 1985.
[14] K. Desoyer, P. Lugner, H. Springer, "Dynamic effects of active elements in manipulators and their influence upon the controlling drives, *IUTAM/IFFToMM Symposium*, pp. 44-51, Udine, 1985.
[15] I. Rudas, A. Toth, "Efficient Recursive Algorithm for Inverse Dynamics", *Mechatronics*, vol 3, no 2, pp 205-214, 1993.
[16] V. Mata, S. Provenzano, J. I. Cuadrado, F. Valero, "An $O(n)$ Algorithm for Solving the Inverse Dynamic Problem in Robots by Using The Gibbs-Appell Formulation", *Proc. of Tenth World Congress IFToMM*, vol. 3, pp 1208-1215, Oulu, June 1999.