

Resolución del problema dinámico directo de cadenas abiertas utilizando las ecuaciones de Gibbs-Appell

Open chains direct dynamic problem resolution using the Gibbs-Appell equations

Sebastián E. Provenzano Randazzo

Universidad de Los Andes, Facultad de Ingeniería. Escuela de Ingeniería Mecánica. Mérida, Venezuela.
prse@ing.ula.ve

Vicente Mata Amela

Departamento de Ingeniería Mecánica y de Materiales, Universidad Politécnica de Valencia. Valencia, España
vmata@mcm.upv.es

Resumen

En este trabajo se describe un novedoso método para resolver el problema dinámico directo de robots manipuladores basado en las ecuaciones de la dinámica de Gibbs-Appell. El método se basa en la obtención de la matriz de inercia del sistema mediante el hessiano de la función de Gibbs. Para obtener la mencionada matriz, se propone un algoritmo recursivo eficiente en notación tensorial y con una complejidad de orden $O(n^2)$, el cual es utilizado en la elaboración de un programa en FORTRAN, con el cual se simula el movimiento de un robot PUMA. Los resultados obtenidos de la simulación se comparan con los producidos al utilizar otros conocidos métodos.

Palabras Claves: Problema dinámico directo, simulación, robot, Gibbs-Appell.

Abstract

In this work is described a novel method to solve the direct dynamic problem of robots manipulators based on the Gibbs Appell equations of the dynamics. The method is based on the obtaining of the system inertia matrix by means of the Gibbs function hessian. To obtain this matrix, an efficient $O(n^2)$ recursive algorithm in tensor notation is proposed, which is used in the elaboration of a FORTRAN program, with this program the movement of a robot PUMA is simulated. The simulation results are compared with those other well-known methods results.

Key Words: Direct Dynamic Problem, simulation, robot, Gibbs-Appell.

1 Introducción

Durante el transcurso de las últimas tres décadas, numerosos investigadores han utilizado diferentes principios de la dinámica para obtener las ecuaciones que modelan el comportamiento dinámico de brazos robot. Las primeras formulaciones desarrolladas se basaron en la representación en forma cerrada de las ecuaciones y se utilizaron preferentemente las ecuaciones de Lagrange-Euler (L-E) para tal fin. Estas formulaciones demostraron ser poco eficientes por lo que se inició la utilización de relaciones de recurrencia entre las ecuaciones de la dinámica con el fin de reducir el número de operaciones algebraicas involucradas. Las ecuaciones de Newton-Euler (N-E) demostraron ser las

mas apropiadas para desarrollar este tipo de formulaciones, y es a partir de éstas, que se han desarrollado las formulaciones más eficientes que se conocen. Otras formulaciones, basadas en las ecuaciones de Kane han originado algoritmos con una complejidad computacional similar a la implicada en algoritmos basados en las ecuaciones de N-E. La utilización de principios de la dinámica distintos a L-E, N-E o Kane, ha sido menor, e incluso los algoritmos derivados de ellas involucran una elevada complejidad computacional. Actualmente se acepta que la utilización de diversos principios de la dinámica conduce a formulaciones similares con una complejidad computacional equivalente. Este punto ha sido parcialmente demostrado al encontrarse que si se parte de las ecuaciones de L-E y se utilizan las rela-

ciones apropiadas, es posible arribar a una formulación equivalente a la obtenida de las ecuaciones de N-E, aunque es mayor el esfuerzo requerido para llegar a las expresiones finales (Silver, 1982). Es por ello que la mayoría de las formulaciones que originan algoritmos eficientes han sido desarrolladas a partir de las ecuaciones de N-E. Featherstone y Orin (2000), hacen una detallada revisión de estos métodos y los algoritmos derivados.

Uno de los principios menos utilizados en la resolución de los problemas dinámicos de robots manipuladores son las ecuaciones de Gibbs-Appell (G-A). Estas ecuaciones fueron independientemente descubiertas por Gibbs en 1879 y Appell en 1899, y se asegura que “*constituyen probablemente la forma mas simple y comprensiva de las ecuaciones de movimiento jamás descubierta*” (Pars, 1972). Se resalta igualmente en la literatura especializada la forma sencilla con la que estas ecuaciones tratan los sistemas mecánicos sometidos a restricciones de tipo holónomo y no-holónomo. En vista de estas afirmaciones, resulta sorprendente el hecho de que una revisión bibliográfica en la literatura del área revele la escasa aplicación que han tenido las ecuaciones de G-A en la dinámica moderna. Hace unos años, la supuesta relación que tienen las ecuaciones de G-A con las ecuaciones de la dinámica de Kane ocasionó un buen número de trabajos y comentarios al respecto –ver por ejemplo (Desloge, 1986), (Sharf et al, 1992)–. Ya en años recientes, Udwarda y Kalaba (1998) han publicado un trabajo en el que se desarrollan nuevas formas de las ecuaciones de G-A. Más específicamente, en el campo de la robótica, Popov propuso un método, posteriormente desarrollado por Vukobratovic y Potkonjak (1985), en el cual se utilizaron las ecuaciones de G-A para desarrollar una formulación en forma cerrada de elevada complejidad computacional. Este método fue posteriormente utilizado por Desoyer y Lugner (1989) quienes resolvieron, por medio de una formulación recursiva de orden $O(n^2)$, el problema dinámico inverso utilizando la matriz jacobiana del manipulador con el fin de evitar el desarrollo explícito de las derivadas parciales. Otro enfoque fue sugerido por Vereshchagin (1974), quien planteó las ecuaciones de movimiento de manipuladores utilizando el principio de Gauss y la función de Gibbs. Recientemente Mata et al. (1999), presentaron una formulación de orden $O(n)$ para resolver el problema dinámico inverso, dicha formulación establece relaciones recursivas que involucran un número de operaciones algebraicas reducido.

Los algoritmos que modelan el comportamiento dinámico de manipuladores, se dividen en dos tipos: los algoritmos que resuelven el problema dinámico inverso y los que le dan solución al problema dinámico directo. A partir de los algoritmos que resuelven el problema dinámico inverso se obtienen, de manera algebraica, las fuerzas ejercidas por los actuadores para una determinada configuración del manipulador (posición, velocidad y aceleración). Por otro lado, en el problema dinámico directo se requiere el cómputo de las aceleraciones de los nudos del manipulador,

dadas las fuerzas que ejercen los actuadores. Este problema es parte del proceso que debe seguirse para realizar la simulación del comportamiento dinámico del manipulador. Dicho proceso se completa al calcular las velocidades y posiciones de los nudos mediante un proceso de integración numérica en el cual las aceleraciones de los nudos y la configuración inicial son datos del problema. Los algoritmos desarrollados para resolver el problema dinámico directo utilizan, sin importar el principio de la dinámica del cual se derivan, alguno de los siguientes enfoques: Cómputo de las aceleraciones de los nudos mediante la obtención, planteamiento y resolución de un sistema ecuaciones simultáneas; ó, cálculo recursivo de las aceleraciones de los nudos propagando el movimiento y las fuerzas de restricción a lo largo del mecanismo. Los algoritmos derivados de los métodos que utilizan el primer enfoque requieren de la obtención de la matriz de inercia del sistema y del vector bias. La matriz de inercia del sistema también se utiliza en esquemas de control avanzado, así como en procedimientos destinados a la estimación de parámetros; por lo que su cómputo, mediante procedimientos sencillos y eficientes, es provechoso no sólo para la simulación del movimiento del sistema. Los métodos basados en el primer enfoque requieren de la resolución de un sistema lineal, por lo cual los algoritmos derivados poseen usualmente una complejidad computacional de orden $O(n^3)$. El método más conocido que sigue este enfoque fue propuesto por Walker y Orin (1982), los cuales desarrollaron (utilizando N-E) el método de cuerpo rígido compuesto (MCRC) mediante el cual se obtiene la matriz de inercia del sistema de manera recursiva y con una complejidad de orden $O(n^2)$. Angeles y Ma (1988), propusieron otro método que sigue este enfoque, el cual se basa en la obtención del complemento natural ortogonal (CNO) de las ecuaciones de restricción cinemática del manipulador con una complejidad de orden $O(n^3)$, empleando las ecuaciones de Kane para obtener el vector bias. Por otro lado, los algoritmos derivados de métodos que utilizan el segundo enfoque poseen usualmente una complejidad de orden $O(n)$. Estos algoritmos no obtienen la matriz de inercia del sistema, por lo que su aplicación está limitada a simulación del movimiento de sistemas. El método más conocido de los que utilizan el segundo enfoque es el método del cuerpo articulado (MCA) desarrollado por Featherstone (1983). El número de operaciones algebraicas que éste involucra es inferior a las que implica el MCRC solo para sistemas que contienen nueve o más cuerpos. Las diferencias existentes entre los métodos MCRC y MCA han sido estudiadas por algunos investigadores durante la última década. Entre ellos, Ascher et al. (1997) demostraron al resolver el problema dinámico directo en un sistema compuesto por dos cuerpos con un número de condición alto, que el MCA produce resultados más precisos que el MCRC, debido a problemas de truncamiento propios del algoritmo con el que se obtiene la matriz de inercia del sistema en este método.

En este trabajo se propone la resolución del proble-

ma dinámico directo de robots manipuladores utilizando las ecuaciones de G-A. Para ello se emplea una formulación que sigue el primer enfoque explicado anteriormente. Para obtener la matriz de inercia del sistema se ha desarrollado un novedoso algoritmo recursivo de orden $O(n^2)$ en notación tensorial. Los resultados numéricos obtenidos de este método son comparados con los producidos de los métodos MCA y MCRC.

El trabajo se encuentra organizado de la siguiente manera: en la sección 2 se desarrolla una formulación para resolver el problema dinámico directo; el algoritmo, derivado de las formulación propuesta se expone en la sección 3; en la sección 4 se analiza, mediante un ejemplo, el error numérico que arroja el método para resolver el problema dinámico directo propuesto.

2 Método del hessiano de la función de Gibbs

Las ecuaciones de la dinámica de G-A (Pars, 1972) parten de la definición de la función de Gibbs, la cual, para un sistema mecánico compuesto por n cuerpos y expresada en coordenadas generalizadas, se puede escribir de la siguiente manera (Vukobratovic y Potkonjak, 1985):

$$G = \frac{1}{2} \ddot{\mathbf{q}}^T \mathbf{D} \ddot{\mathbf{q}} + \bar{\varphi}^T \ddot{\mathbf{q}} + f(\bar{\mathbf{q}}, \dot{\mathbf{q}}) \quad (1)$$

donde \mathbf{D} es la matriz de inercia del sistema, $\bar{\varphi}$ es el vector bias y $\bar{\mathbf{q}}, \dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$ son los vectores de las coordenadas, velocidades y aceleraciones generalizadas respectivamente. Las ecuaciones de G-A se obtienen al derivar parcialmente la función de Gibbs con respecto a las aceleraciones generalizadas:

$$\frac{\partial G}{\partial \ddot{\mathbf{q}}} \equiv \bar{\tau} = \mathbf{D}(\bar{\mathbf{q}}) \ddot{\mathbf{q}} + \bar{\varphi} \quad (2)$$

Esta ecuación modela el comportamiento dinámico del sistema y coincide con la obtenida utilizando otros principios de la dinámica (Craig, 1986). En esta expresión la matriz de inercia del sistema (\mathbf{D}) y el vector bias ($\bar{\varphi}$) pueden ser calculados mediante algoritmos recursivos desarrollados de manera separada, y $\bar{\tau}$ representa el vector de fuerzas generalizadas (conocido). El método para resolver el problema dinámico directo que se desarrolla en este trabajo se basa en la resolución de esta expresión. En este trabajo, el vector bias es calculado mediante la adaptación de un algoritmo desarrollado para resolver el problema dinámico inverso, en el cual los términos correspondientes a las aceleraciones de los nudos se han eliminado, según el método propuesto por Walker y Orin (1982). El algoritmo inverso utilizado se basa en las ecuaciones de G-A y es una versión en notación tensorial del propuesto por Mata et al. (1999). Para obtener la matriz de inercia del sistema se

desarrolla un nuevo método basado en las ecuaciones de G-A. Para ello se deriva nuevamente de manera parcial la Ec. (2) con respecto a cada una de las aceleraciones generalizadas, obteniéndose la matriz hessiana de la función de Gibbs, la cual coincide con la matriz de inercia del sistema:

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

De la expresión anterior se obtiene de manera natural un algoritmo sencillo de orden $O(n^3)$ para el cómputo de \mathbf{D} , aunque es posible reducirlo a un algoritmo recursivo de orden $O(n^2)$. El procedimiento empleado para hallar la expresión que permite calcular los coeficientes de la matriz de inercia del sistema por medio de un proceso de derivación sucesiva parcial de la función de Gibbs, se simplifica si se toman en cuenta la siguiente serie de cuestiones, las cuales resultan de la aplicación de la notación de Denavit-Hartenberg (D-H) modificada (Angeles y Ma, 1988):

$$\frac{\partial}{\partial \ddot{q}_j} \frac{\partial^i \dot{\omega}_i}{\partial \ddot{q}_k} = \frac{\partial}{\partial \ddot{q}_j} {}^i \ddot{z}_i = 0 \quad (4)$$

$$\frac{\partial}{\partial \ddot{q}_j} \frac{\partial^i \vec{r}_{o_i}}{\partial \ddot{q}_k} = \frac{\partial}{\partial \ddot{q}_j} {}^i \mathbf{R}_{i-1} \left[\frac{\partial^{i-1} \ddot{\vec{r}}_{o_{i-1}}}{\partial \ddot{q}_k} + \frac{\partial^{i-1} \dot{\omega}_{i-1}}{\partial \ddot{q}_k} \wedge {}^{i-1} \vec{r}_{o_{i-1}, o_i} \right] = 0 \quad (5)$$

$$\frac{\partial}{\partial \ddot{q}_j} \frac{\partial^i \vec{r}_{G_i}}{\partial \ddot{q}_k} = \frac{\partial}{\partial \ddot{q}_j} \left(\frac{\partial^i \ddot{\vec{r}}_{o_i}}{\partial \ddot{q}_k} + \frac{\partial^i \dot{\omega}_i}{\partial \ddot{q}_k} \wedge {}^i \vec{r}_{o_i, G_i} \right) = 0 \quad (6)$$

$$\frac{\partial^i \dot{\omega}_i}{\partial \ddot{q}_j} = \begin{cases} [0 \ 0 \ 0]^T & i < j \\ {}^i \mathbf{R}_{i-1} \cdot {}^{i-1} \mathbf{R}_{i-2} \cdots {}^1 \mathbf{R}_j \cdot \frac{\partial^j \dot{\omega}_j}{\partial \ddot{q}_j} & i > j \\ {}^i \ddot{z}_i & i = j \end{cases} \quad (7)$$

$$\frac{\partial^i \ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_k} = \frac{\partial^i \dot{\omega}_i}{\partial \ddot{q}_k} \wedge {}^i \vec{r}_{o_i, G_i} \quad \text{ó en notación tensorial}$$

$$\frac{\partial^i \ddot{\vec{r}}_{G_i}}{\partial \ddot{q}_k} = \left(\frac{\partial^i \dot{\omega}_i}{\partial \ddot{q}_k} \right)^T \cdot {}^i \tilde{\vec{r}}_{o_i, G_i} \quad (8)$$

donde ${}^i \dot{\omega}_i$ es el vector aceleración angular de la barra i , ${}^i \ddot{\vec{r}}_{o_i}$ y ${}^i \ddot{\vec{r}}_{G_i}$ son los vectores aceleración del origen y del centro de masas de la misma barra respectivamente, y ${}^i \vec{r}_{o_i, G_i}$ es el vector posición de su centro de masas; todos ellos expresados en el sistema coordenado local. Mientras que ${}^{i-1} \vec{r}_{o_{i-1}, o_i}$ es el vector posición del origen de la barra i con respecto al origen de la barra $i-1$ y expresado en el sistema coordenado

local a la barra $i-1$. La matriz de rotación ${}^i\mathbf{R}_{i-1}$ transforma los vectores que premultiplica desde el sistema coordenado local $i-1$ al i . En la notación D-H utilizada, los sistemas coordenados ligados a las barras se sitúan en el inicio de cada una de ellas, con lo cual, el sistema coordenado correspondiente al cuerpo i está situado sobre la articulación i , en consecuencia ${}^i\bar{\mathbf{z}}_i = [0 \ 0 \ 1]^T$. La utilización de notación tensorial en estas expresiones, no sólo permite obtener expresiones más compactas y elegantes, sino que también reduce el número de operaciones aritméticas involucradas en algunas de ellas –ver detalles en (Balafoutis y Patel, 1991) por ejemplo–. Empleando las Ec. (4-6), la expresión para el cómputo de los coeficientes de la matriz de inercia del sistema, toma la siguiente forma:

$$D_{jk} = \sum_{i=j}^n \left[m_i \left(\frac{\partial^i \ddot{\mathbf{r}}_{G_i}}{\partial \ddot{q}_k} \right)^T \frac{\partial^i \ddot{\mathbf{r}}_{G_i}}{\partial \ddot{q}_j} + \left(\frac{\partial^i \dot{\hat{\omega}}_i}{\partial \ddot{q}_k} \right)^T {}^i\mathbf{I}_{G_i} \frac{\partial^i \dot{\hat{\omega}}_i}{\partial \ddot{q}_j} \right] \quad (9)$$

La utilización de esta última expresión para elaborar un algoritmo aplicable a manipuladores que sea computacionalmente eficaz, pasa por estudiar las relaciones de recurrencia que existen entre sus elementos, ya que la aplicación directa de la expresión produciría un algoritmo con un elevado número de operaciones. Tomando en cuenta las Ec. (7-8) y propiedades del producto vectorial, la expresión (9) puede escribirse:

$$D_{jk} = \sum_{i=j}^n \left[\left({}^i\mathbf{R}_k \frac{\partial^k \dot{\hat{\omega}}_k}{\partial \ddot{q}_k} \right)^T \left({}^i\mathbf{I}_{G_i} - m_i {}^i\tilde{\mathbf{r}}_{O_i,G_i} {}^i\tilde{\mathbf{r}}_{O_i,G_i} - m_i {}^i\tilde{\mathbf{r}}_{O_i,O_i} {}^i\tilde{\mathbf{r}}_{O_i,G_i} \right) \frac{\partial^i \dot{\hat{\omega}}_i}{\partial \ddot{q}_j} \right] \quad (10)$$

En la siguiente sección se presenta un algoritmo para la obtención de la matriz de inercia del sistema basado en esta expresión.

3 Algoritmo gah

A continuación se muestra un algoritmo recursivo de orden $O(n^2)$ para el cómputo de la matriz de inercia del, basado en la formulación propuesta en la sección anterior.

Paso 1

$$M_n = m_n$$

Para $i = n - 1$ hasta 1, hacer:

$$M_i = m_i + M_{i+1}$$

$${}^i\tilde{\mathbf{r}}_i = m_i {}^i\tilde{\mathbf{r}}_{O_i,G_i} + M_{i+1} {}^i\tilde{\mathbf{r}}_{O_i,O_{i+1}}$$

$${}^i\mathbf{I}_{O_i} = {}^i\mathbf{I}_{G_i} - m_i {}^i\tilde{\mathbf{r}}_{O_i,G_i} {}^i\tilde{\mathbf{r}}_{O_i,G_i}$$

$${}^i\mathbf{E}_i = {}^i\mathbf{I}_{O_i} - M_{i+1} {}^i\tilde{\mathbf{r}}_{O_i,O_{i+1}} {}^i\tilde{\mathbf{r}}_{O_i,O_{i+1}}$$

Paso 2

Para $i = 1$ hasta $n - 1$, y $j = i + 1$ hasta n , hacer

$${}^j\tilde{\mathbf{r}}_{O_i,O_{i+1}} = {}^j\mathbf{R}_{j-1} {}^{j-1}\tilde{\mathbf{r}}_{O_i,O_{i+1}}$$

Para $i = n - 2$ hasta 1 y $j = n$ hasta $i + 2$, hacer

$${}^j\tilde{\mathbf{r}}_{O_i,O_j} = {}^j\tilde{\mathbf{r}}_{O_i,O_{i+1}} + {}^j\tilde{\mathbf{r}}_{O_{i+1},O_j}$$

Paso 3

Para $i = 1$ hasta n , asignar: $\frac{\partial^j \dot{\hat{\omega}}_j}{\partial \ddot{q}_j} = [0 \ 0 \ 1]^T$

Para $i = 1$ hasta $n - 1$ y $j = i + 2$ hasta n , hacer:

$$\frac{\partial^j \dot{\hat{\omega}}_j}{\partial \ddot{q}_i} = {}^i\mathbf{R}_{i-1} \frac{\partial^{j-1} \dot{\hat{\omega}}_{j-1}}{\partial \ddot{q}_i}$$

Paso 4

Para $i = n, n - 1$ hasta 1, hacer

$${}^i\tilde{\phi}_{i+1} = {}^i\mathbf{R}_{i+1} {}^{i+1}\tilde{\phi}_{i+1}$$

$${}^i\tilde{\phi}_i = {}^i\tilde{\gamma}_i + {}^i\tilde{\phi}_{i+1}$$

Paso 5

$${}^n\boldsymbol{\Psi}_n = {}^n\mathbf{I}_{O_n}$$

Para $i = n - 1$ hasta 2, hacer

$${}^i\boldsymbol{\Psi}_i = {}^i\mathbf{E}_i - {}^i\tilde{\mathbf{r}}_{O_i,O_{i+1}} {}^i\tilde{\phi}_{i+1} - \left({}^i\tilde{\mathbf{r}}_{O_i,O_{i+1}} {}^i\tilde{\phi}_{i+1} \right)^T + {}^i\mathbf{R}_{i+1} {}^{i+1}\boldsymbol{\Psi}_{i+1} {}^i\mathbf{R}_i$$

Paso 6

Para $i = 1$ hasta n ,

$$D_{ii} = \left(\frac{\partial^i \dot{\hat{\omega}}_i}{\partial \ddot{q}_i} \right)^T {}^i\boldsymbol{\Psi}_i \frac{\partial^i \dot{\hat{\omega}}_i}{\partial \ddot{q}_i}$$

Para $j = n$ hasta 2 y $k = j - 1$ hasta 1, hacer

$$D_{jk} = \left(\frac{\partial^j \dot{\hat{\omega}}_j}{\partial \ddot{q}_k} \right)^T \left[{}^j\boldsymbol{\Psi}_j - {}^j\tilde{\mathbf{r}}_{O_j,O_i} {}^j\tilde{\phi}_j \right] \frac{\partial^j \dot{\hat{\omega}}_j}{\partial \ddot{q}_j}$$

La resolución del problema dinámico directo se completa, como ya se ha mencionado, con la obtención del vector bias y la resolución del sistema lineal implicado al despejar las aceleraciones generalizadas en la Ec. (2). La complejidad total que involucra este proceso se indica en la Tabla 1. En la Tabla 2 se compara el número de operaciones que efectúan el método propuesto y otros conocidos métodos para resolver el problema dinámico directo de un manipulador de seis grados de libertad.

Tabla 1.- Complejidad computacional total del método propuesto

	Productos ($n = 6$)		Sumas ($n = 6$)	
Cómputo de D	$11.5n^2 + 19.5n - 49$	(482)	$8.5n^2 + 31.5n - 69$	(426)
Cómputo del vector bias	$96n - 108$	(468)	$83n - 105$	(393)
Resolución del sist. lineal	$1/6n^3 + 1.5n^2 + 1/3n$	(92)	$1/6n^3 + 1.5n^2 - 2/3n$	(66)
Total	$1/6n^3 + 13n^2 + 695/6n - 157$	(1042)	$1/6n^3 + 10n^2 + 683/6n - 174$	(905)

Tabla 2.- Número de operaciones necesarias para resolver el problema dinámico directo utilizando diversos métodos ($n = 6$).

Autores	Método	Productos	Sumas
Walker y Orin (1982)	MCRC	1627	1261
Angeles y Ma (1988)	CNO	1353	1165
Featherstone (1987)	MCA	1533	1415
Este trabajo	GAH	1042	905

4 Validación y ejemplo de aplicación

En esta sección se efectúa la simulación del movimiento de un robot PUMA, cuyo elemento terminal describe una trayectoria recta con inclinación constante y con una velocidad igualmente constante de 0.1 m/seg. La simulación parte de la posición ${}^0\vec{r}_{O_0, TCP}|_{t=0s} = [0.50 \ 0.00 \ 0.05]^T$ con una inclinación (Z, Y, Z) del sistema local ligado a la barra 6 de $[0^\circ \ 90^\circ \ 180^\circ]$. Luego de 5.0 segundos, tiempo de duración del movimiento, el elemento terminal debe alcanzar la posición ${}^0\vec{r}_{O_0, TCP}|_{t=5s} = [0.62 \ 0.30 \ 0.11]^T$. Para efectuar la simulación propuesta en este ejemplo, se ha escrito en lenguaje FORTRAN un programa de simulación basado en el método propuesto. El programa se ha escrito en doble precisión y es ejecutado en un PC Pentium II de 400 MHz. Para resolver el sistema lineal se ha utilizado el método de eliminación basado en la descomposición de Cholesky, y para integrar el sistema de ecuaciones diferenciales se utiliza la técnica de Runge-Kutta de quinto orden. En el proceso de integración se emplea una tolerancia de 1×10^{-6} y un intervalo de 0.1 s. En la Fig. 1 se muestra el robot en la posición inicial, la trayectoria planeada y la posición final. Para obtener las fuerzas generalizadas que ocasionan que el elemento terminal describa la trayectoria programada, se ha resuelto el problema dinámico inverso para cada instante de tiempo considerado, utilizando como datos de entrada los resultados de la resolución del problema cinemático inverso. En la Fig. 2 se muestra un diagrama de bloques del procedimiento empleado, donde el

superíndice “*” denota las coordenadas, velocidades y aceleraciones generalizadas obtenidas de la integración numérica del método desarrollado en este trabajo.

En la Tabla 3 se muestran los errores promedio resultantes en coordenadas, velocidades y aceleraciones generalizadas de los nudos, comparados con los resultantes del problema cinemático inverso.

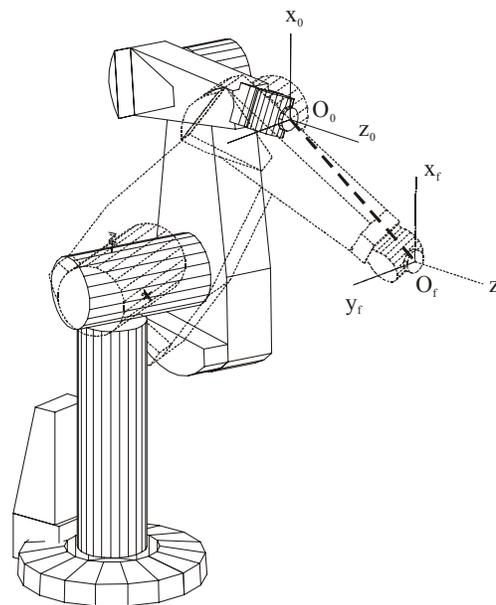


Fig. 1.- Posición inicial, final y trayectoria.

Tabla 3.- Diferencias promedio en los resultados de la simulación.

nudo	$prom q_i - q_i^* $ (rad)	$prom \dot{q}_i - \dot{q}_i^* $ (rad/s)	$prom \ddot{q}_i - \ddot{q}_i^* $ (rad/s ²)
1	$6.91878269 \times 10^{-4}$	$1.28209533 \times 10^{-3}$	$2.84439095 \times 10^{-3}$
2	$1.73972101 \times 10^{-2}$	$1.66087380 \times 10^{-3}$	$9.22318761 \times 10^{-2}$
3	$8.54693990 \times 10^{-4}$	$1.66087459 \times 10^{-3}$	$3.13261587 \times 10^{-3}$
4	$1.73972067 \times 10^{-2}$	$4.05637323 \times 10^{-2}$	$9.22318581 \times 10^{-2}$
5	$1.78259578 \times 10^{-2}$	$4.17630699 \times 10^{-2}$	$9.54221869 \times 10^{-3}$
6	$6.91878269 \times 10^{-4}$	$1.21908065 \times 10^{-3}$	$2.43724278 \times 10^{-3}$

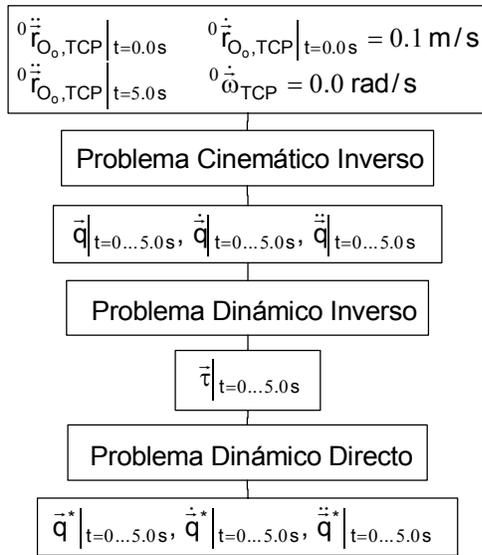


Fig. 2.- Diagrama de bloques del procedimiento.

Esta misma simulación ha sido efectuada por medio de programas basados en los métodos MCRC y MCA, produciéndose en ambos casos prácticamente los mismos resultados que con el método GAH, lo cual valida los resultados obtenidos del método propuesto en este trabajo. La principal diferencia entre métodos se observa en el tiempo empleado en realizar la simulación y en el número de veces que es evaluada la función de integración. En la Tabla 4 se muestra una comparación de dichas variables, observándose que el método propuesto en este trabajo efectúa la simulación en menor tiempo que los métodos MCRC y MCA, lo cual era de esperarse debido a que el GAH efectúa menor número de operaciones aritméticas que los otros dos métodos en este ejemplo. En cuanto al número de evaluaciones de la función de integración, se observa que el GAH efectúa menor cantidad de evaluaciones de la función que el MCRC, acercándose al número de evaluaciones que efectúa el MCA. El hecho de que el MCRC efectúe más evaluaciones de la función que el MCA, confirma las observaciones hechas por Ascher et al. (1997), los cuales observaron que el MCRC posee una formulación rígida que ocasiona más evaluaciones de la función de integración, debido a la dificultad de la rutina de integración para cumplir con la tolerancia exigida. En este apartado también se destaca el método propuesto en este trabajo, el cual efectúa prácticamente el mismo número de evaluaciones que el MCA, pero en un tiempo considerablemente menor.

Tabla 4.- Tiempo de procesamiento y número de evaluaciones de la función para cada método.

	MCRC	MCA	GAH
Tiempo de Proc. (seg.)	0.65	0.71	0.59
Nº de ev. de función	1122	1089	1092

5 Conclusiones

En este trabajo se ha propuesto un método basado en las ecuaciones de la dinámica de Gibbs-Appell con el cual se resuelve eficientemente el problema dinámico directo de manipuladores. El método calcula la matriz de inercia del sistema mediante un novedoso algoritmo que se basa en la matriz hessiana de la función de Gibbs. El algoritmo se ha planteado en notación tensorial lo que ha permitido obtener una formulación elegante que además involucra un reducido número de operaciones aritméticas. El método ha sido validado por medio de un ejemplo de aplicación, comparando los resultados obtenidos con los producidos por otros dos programas basados en los conocidos métodos MCRC y MCA. Los resultados además han demostrado que el método propuesto posee una formulación numéricamente menos rígida que la del MCRC.

6 Referencias

- Angeles, J. y Ma, O. 1988. "Dynamic Simulation of n-Axis Serial Robotic Manipulators Using a Natural Orthogonal Complement." *Int. Journal of Robotic Research*, Vol. 7, Nº 5, pp. 32-47.
- Ascher, U. M., Pai, D. K., y Cloutier, B. P. 1997. "Forward Dynamics Elimination Methods, And Formulation Stiffness in Robot Simulation." *Int. Journal of Robotic Research*, Vol. 16, Nº 6, pp. 749-758.
- Balafoutis, C. A. y Patel, R. V. 1991. *Dyn. An. of Robot Man.: A Cart. Tensor App.*, Kluwer Ac. Press, Boston.
- Craig, J. J. 1986. *Introduction to robotics: Mechanics and control*, Addison-Wesley, Reading.
- Desloge, E. 1986. "Relationship between Kane's Equations and the Gibbs-Appell Equations." *Journal of Guidance, Control, and Dynamics*, Vol 10, Nº 1.
- Desoyer, K. y Lugner, P. 1989. "Recursive Formulation for the Analytical or Numerical Application of the Gibbs-Appell Method to the Dynamics of Robots." *Robotica*, Vol 7, pp. 343-347.
- Featherstone, R. 1983. "The Calculation of Robot Dynamics using Articulated-Body Inertias." *Int. Journal of Robotic Research*, Vol. 2, Nº 1, pp. 13-30.
- Featherstone, R. 1987. *Robot dynamics algorithms*, Kluwer Academic, Boston.
- Featherstone, R. y Orin, D. E. 2000. "Robot Dynamics: Equations and Algorithms." *Proc. Of the 2000 IEEE Int. Conf. On Robotics & Automation*, pp. 826-834.
- Mata, V., Provenzano, S., Cuadrado, J. I., y Valero, F. 1999. "An O(n) Algorithm for Solving the Inverse Dynamic Problem in Robots by Using The Gibbs-Appell Formulation." *Proc. of Tenth World Congress IFToMM*, pp. 1208-1215.
- Pars, L. 1972. *A Treatise on Analytical Dynamics*, Ox Bow Press, Connecticut.
- Sharf, I., D'eleuterio, G. M. T., y Hughes, P. C. 1992. "On the dynamics of Gibbs, Appell, and Kane." *European Jour-*

nal of Mechanics. A/Solids, Vol. 11, N° 2, pp. 145-155.

Silver, W. M. 1982. "On the equivalence of Lagrangian and Newton-Euler dynamics of manipulators." Int. Journal of Robotic Research, Vol. 1, pp. 60-70.

Udwadia, F. E. y Kalaba, R. E. 1998. "The Explicit Gibbs-Appell Equation and Generalised Inverse Forms." Quarterly of Applied Mathematics, Vol. LVI, N° 2, pp. 277-288.

Vereshchagin, A. F. 1974. "Computer simulation of the

dynamics of complicated mechanisms of robotic manipulators." Engineering Cybernetics, Vol. 6, pp. 65-70.

Vukobratovic, M. y Potkonjak, V. 1985. Applied dyn. and CAD of man. robots, Springer-Verlag, Berlin.

Walker, M. W. y Orin, D. E. 1982. "Efficient Dynamic Computer Simulation of Robotic Mechanisms." Journal of Dynamic Systems, Measurement and Control, Vol. 104, pp. 205-211.