

Diseño y desarrollo de un simulador de robots manipuladores

Daniel Librado Martínez Vázquez, Gastón H. Salazar-Silva y José Alfredo Rosas Flores

Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas,
Instituto Politécnico Nacional
ghsalazar@ipn.mx

Resumen

El problema abordado es el diseño y desarrollo de un programa computacional con interfaz gráfica llamado Simulador de Robots Manipuladores I, que simula el comportamiento cinemático y dinámico de un robot manipulador serial de hasta seis grados de libertad. Este comportamiento se puede visualizar por medio de un manipulador virtual y gráficas de las variables de mayor interés.

1. Introducción

La simulación de un sistema es el proceso de diseñar un modelo (generalmente matemático) de un sistema real o imaginario y realizar experimentos con éste [1]. La simulación tiene ventajas económicas, de ahorro en tiempo y minimización de riesgos.

El propósito de realizar experimentos mediante una simulación es el de entender el comportamiento o evaluar estrategias para la operación de un sistema sin requerir la construcción o experimentación con el sistema físico real. En una simulación por método analítico, se hacen suposiciones acerca del sistema y se derivan algoritmos y relaciones matemáticas para describir su comportamiento.

El presente trabajo plantea el diseño y desarrollo de un programa computacional con interfaz gráfica llamado *Simulador de Robots Manipuladores I*, que simule los comportamientos cinemático directo y dinámico de la estructura de robots manipuladores seriales con 6 GDL máximo. Este comportamiento se podrá visualizar en 3D y en forma de gráficas en 2D de cada variable.

Gracias a su diseño gráfico, será de gran ayuda en la docencia ya que puede funcionar como un laboratorio virtual de Robótica, ayudando a que los alumnos comprendan de mejor manera los aspectos teóricos.

También puede ser de gran utilidad en el proce-

so de diseño de robots, al conocer el comportamiento cinemático directo y dinámico del robot sin necesidad de construirlo físicamente. Ahorrándose tanto dinero como tiempo.

En esta investigación se encontraron más de 30 simuladores de manipuladores. Algunos grandes limitaciones debido a las licencias de uso y distribución [2, 3, 4]; en otros casos no se ha podido encontrar mayor información o el código fuente no esta disponible o no es ejecutable [5, 6]. El código del simulador desarrollado se encuentra disponible en forma gratuita en [7]. Más información acerca del desarrollo del simulador se encuentra en [8].

El presente trabajo se desarrolla de la siguiente manera. Después de esta Introducción, se presenta la sección 2 donde se documenta los métodos utilizados en el simulador; en la sección 3 se muestran algunos experimentos realizados con el simulador y en la sección 4 se detallan las conclusiones.

2. Desarrollo

Por simulación de un robot básicamente se entiende el análisis del comportamiento cinemático y dinámico de éste.

Para determinar la cinemática se utilizó el método convencional de Denavit–Hartenberg [9, 10, 11]. Este método no solo proporcionó la forma de describir la cinemática, si no también como describir el manipulador de una forma intuitiva. Cabe recordar que también existen otras metodologías, como el método modificado de Denavit–Hartenberg [12], el método de tornillos [13], ángulos de Euler y cuaterniones. Estas metodologías no se utilizaron debido a no considerarse adecuadas para su uso en la simulación desarrollada.

Por otro lado, para realizar una simulación se necesita conocer la dinámica del manipulador, es decir, el comportamiento de las coordenadas generalizadas en función de las fuerzas generalizadas. Este problema es

abordado por diversos métodos, tanto simbólicos como recursivos.

En el presente trabajo se utilizó el primer método recursivo de Walker-Orin que a continuación se describe [14]. Este método requiere del cálculo eficiente de la dinámica inversa. Estos autores sugieren utilizar el método de Newton-Euler para este fin, por lo cual fue utilizado en este simulador. Este método se describe en [10, 11, 15, 16, 17, 12].

2.1. Implantación del simulador

Un simulador sigue básicamente el patrón de un interprete. Un interprete es opera bajo el ciclo leer-evaluar-escribir. Un interprete debe leer las señales de entrada del modelo, las evalúa bajo el contexto del modelo y escribe los resultados ya sea en una gráfica o en un archivo.

Existe toda una taxonomía [18, 19] para describir los diferentes tipos de modelos dinámicos que se pueden desarrollar en un simulador. En la figura ?? se muestra un diagrama de flujo simplificado del simulador.

Existe toda una taxonomía [18, 19] para describir los diferentes tipos de modelos dinámicos que se pueden desarrollar en un simulador. En este caso se utilizó la descripción por medio de un sistema de ecuaciones diferenciales, el cual se resuelve por medio del método de Walker-Orin [14] que se describe 2.4.

2.2. Cinemática directa de manipuladores seriales

Se entiende por modelo cinemático directo a la relación que determina la posición y orientación del efector final con respecto a la base.

Un robot manipulador serial consta de cuerpos llamados eslabones, conectados por uniones de tipo rotacional o translacional, donde cada una de estas tiene un GDL [10, 11]. Para un manipulador con N uniones numeradas de 1 a N , se tienen $N + 1$ eslabones, numerados de 0 a N . El eslabón 0 es la base del manipulador y generalmente se encuentra fijo, del otro extremo, el eslabón N carga al efector final. La unión i conecta a los eslabones $i - 1$ e i , por lo tanto, la unión 1 es el punto de conexión entre la base y el eslabón 1. Cada eslabón está conectado, cuando mas, con otros dos eslabones, es decir, el eslabón i conecta a los eslabones $i - 1$ e $i + 1$ de tal forma que no se forman cadenas cerradas.

Cada eslabón puede considerarse como un cuer-

po rígido, de esta manera, se puede definir la relación entre dos ejes de unión consecutivos. Cada eslabón puede ser especificado por dos parámetros: la longitud del eslabón, a_i , y la torsión del eslabón, α_i . Generalmente para el primer y último eslabones estos parámetros carecen de importancia, por lo que pueden ser definidos de forma arbitraria como cero. Las uniones pueden ser descritas por dos parámetros: el desplazamiento del eslabón, d_i , y el ángulo de la unión, θ_i .

Para facilitar la descripción de la posición de los eslabones, se fija un *sistema coordenado* diestro y unitario (marco de referencia) a cada uno de éstos, asignando el sistema coordenado i ($o_i x_i y_i z_i$) al eslabón i . Denavit y Hartenberg [9] propusieron un método operacional que sistemáticamente asigna un marco de referencia a cada eslabón de la cadena serial del robot. El eje de la unión $i + 1$ esta alineado con z_i . El origen i (o_i) está localizado en la intersección de z_{i-1} y z_i o en la intersección entre z_i y la normal entre z_{i-1} y z_i . El eje x_i está dirigido a lo largo de la normal común a los ejes z_{i-1} y z_i , (para ejes que se intersectan, el eje x_i es paralelo al plano $z_{i-1} \times z_i$).

Si la articulación i es rotacional (revoluta), θ_i es la variable de articulación y d_i es constante. Por otro lado, si esta articulación es translacional (prismática), d_i es la variable y θ_i es constante. Para poder representar tanto desplazamientos angulares como lineales, se introduce el término de *coordenadas generalizadas* q_i , definidas como:

$$q_i = \begin{cases} \theta_i & \text{si la articulación es revoluta,} \\ d_i & \text{si la articulación es prismática.} \end{cases}$$

Para obtener la expresión con respecto al marco $i - 1$ de un vector expresado en el marco i se utiliza la matriz de transformación homogénea:

$$A_{i-1}^i = \begin{pmatrix} R & a_i \cos \theta_i \\ & a_i \sin \theta_i \\ & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

donde $R_{i-1}^i \in \mathbb{R}^{3 \times 3}$ es una matriz de rotación de la forma

$$R_{i-1}^i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{pmatrix}.$$

Para especificar la posición y rotación del origen del marco N con respecto al marco 0 se hace una concatenación de matrices de transformación de la forma:

$$A_0^N = A_0^1 A_1^2 \cdots A_{N-1}^N$$

Esta matriz es llamada también *Matriz del brazo*, gracias a ésta, la solución a la cinemática directa puede ser

obtenida para cualquier manipulador serial, sin importar el número de uniones o la estructura cinemática.

2.3. Dinámica inversa de cuerpo rígido para manipuladores seriales

En el modelo dinámico inverso las ecuaciones de movimiento para el manipulador se resuelven para ciertas coordenadas generalizadas y se determinan las fuerzas generalizadas que ocasionarán dicho movimiento [10].

Al igual que en cinemática se utilizan coordenadas generalizadas, en dinámica se definen las *fuerzas generalizadas* como:

$$\tau_i = \begin{cases} n_i & \text{si la articulación es revoluta,} \\ f_i & \text{si la articulación es prismática,} \end{cases}$$

donde n_i es un par y f_i es una fuerza, ambos ejercidos por un actuador sobre la articulación.

Las ecuaciones de movimiento para un manipulador de N GDL pueden ser descritas por la representación:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) \quad (2)$$

donde $q(t) \in \mathbb{R}^N$ es el vector de coordenadas generalizadas y describe la configuración del manipulador, $\dot{q}(t) \in \mathbb{R}^N$ es el vector de las velocidades de las articulaciones; $\ddot{q}(t)$ es el vector de aceleraciones de las articulaciones; M es la matriz de inercia o el tensor de inercia del manipulador; C describe los efectos de las fuerzas centrífugas y de Coriolis; F describe la fricción viscosa y de Coulomb y generalmente no es considerada como parte de la dinámica del cuerpo rígido; G es el vector de fuerza derivado de la energía potencial; y $\tau(t) \in \mathbb{R}^N$ es el vector de fuerzas generalizadas asociado con las coordenadas generalizadas $q(t)$.

La ecuación (2) puede ser deducida por varias técnicas, como por ejemplo los métodos de Euler-Lagrange, Newton-Euler, d'Alembert o de Kane. Para mayor explicación del método Euler-Lagrange, se puede consultar [10, 11, 12]. En [10] se abarca el tema de las coordenadas generalizadas d'Alembert, dando un algoritmo concreto para desarrollar su modelo conceptual. En [4] se encuentra la mención al método de Kane y se pueden encontrar tablas comparativas entre los diferentes métodos en [4, 10, 20].

Por ésta exposición de características, se ha tomado la decisión de utilizar la metodología recursiva de Newton-Euler para resolver el problema de la dinámica inversa de manipuladores seriales.

2.3.1. Método recursivo de Newton-Euler

El método recursivo de Newton-Euler (RNE), calcula la dinámica inversa del manipulador, es decir, las fuerzas generalizadas requeridas en las uniones para un conjunto de coordenadas, velocidades y aceleraciones generalizadas [15]. La recursión directa propaga información cinemática (como velocidades angulares, aceleraciones angulares, aceleraciones lineales) desde el marco base (eslabón 0) hasta el efector final. La recursión inversa propaga las fuerzas y los momentos que se encuentran en los eslabones, desde el efector final hasta el marco base.

Versiones de este algoritmo con notación modificada pueden ser encontradas en [10, 11, 15, 16, 17]. Asimismo, en [12] se deduce este método para los parámetros modificados de Denavit-Hartenberg.

2.4. Dinámica directa de cuerpo rígido para manipuladores seriales

Para realizar una simulación se necesita conocer la dinámica directa, es decir, el comportamiento de las coordenadas generalizadas en función de las fuerzas generalizadas. Este problema es abordado por diversos métodos, tanto simbólicos como recursivos y para fines de este trabajo, se utiliza el primer método recursivo de Walker-Orin que a continuación se describe [14].

2.4.1. Primer método recursivo de Walker-Orin

Walker y Orin presentaron una serie de métodos para la resolución de la dinámica directa, los cuales implican del cálculo eficiente de la dinámica inversa. Estos autores sugieren utilizar el método de Newton-Euler para este fin. El primer método de Walker-Orin parte de que la dinámica inversa puede ser descrita como en la ecuación (2) y se define un vector de compensación

$$B = C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q).$$

Este vector de compensación corresponde a las fuerzas generalizadas requeridas para mantener el estado actual del manipulador sin ninguna aceleración. Es obtenido al aplicar el método RNE, solo que igualando a cero las aceleraciones articulares.

Agrupando términos y reordenando, la ecuación 2 se describe como

$$M(q)\ddot{q} = \tau - B. \quad (3)$$

Para el cálculo de la matriz de inercias generalizadas $M(q)$ se necesita modificar el método RNE, en donde

el nuevo método RNE varia en que se deben despreciar todos los términos que involucran velocidad, efectos gravitacionales y efectos debidos a fuerzas y pares externos.

Numéricamente, la j -ésima columna de $M(q)$ se encuentra aplicando el método RNE con las coordenadas generalizadas actuales y un vector e_j (vector $N \times 1$ con el j -ésimo elemento igual a uno y los demás iguales a cero).

Al poder calcular $M(q)$, se conocen todos los elementos involucrados en 3, excepto las aceleraciones generalizadas y se resuelve para ellas de la forma:

$$\ddot{q} = (M(q))^{-1}(\tau - B).$$

En el caso de que $M(q)$ sea de 6×6 y puesto que nunca es singular, la inversión se puede calcular. Si $M(q)$ está mal condicionada numericamente, se puede aplicar métodos de solución de sistemas de ecuaciones lineales, como descomposición triangular.

Hasta este punto, se conocen las aceleraciones generalizadas, pero el objetivo primordial es conocer las coordenadas generalizadas. Para éste fin se utilizan un métodos de integración numérica o resolución de sistemas de ecuaciones diferenciales ordinarias (SE-DO's) para obtener las coordenadas generalizadas.

Una explicación mas detallada a cerca de la dinámica directa por este método, se puede encontrar en [16, 17, 21, 22].

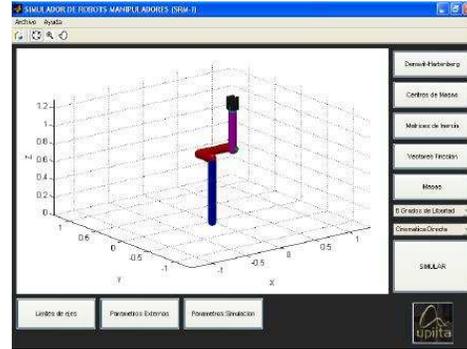
2.4.2. Resolución de sistemas de ecuaciones diferenciales ordinarias

En el modelado de la dinámica directa, se necesita solucionar un sistema de ecuaciones diferenciales ordinarias. Debido a su naturaleza no lineal, es requerido un algoritmo que realice la integración numérica en forma robusta. La literatura sugiere el algoritmo de Runge–Kutta de cuarto orden, sin embargo, se advierte que hay ocasiones en donde este método no converge, por lo que también se describe el algoritmo de Dormand–Prince RK5(4)7M [23].

3. Resultados

Hasta este punto se planteron las bases teóricas para el desarrollo del *Simulador de Robots Manipuladores I* y se hizo un resumen de los puntos desarrollados hasta el momento para la implementación del simulador.

Figura 1: Interfaz del simulador



Cuadro 1: Parámetros de Denavit–Hartenberg

Eslabón	d_i	a_i	α_i	θ_i
1	0.700	0	-90	θ_1
2	0.2435	0.4318	0	θ_2
3	-0.0934	-0.0203	90	θ_3
4	0.4331	0	-90	θ_4
5	0	0	90	θ_5
6	0.0640	0	0	θ_6

En la figura 1 se puede ver el interfaz gráfico del simulador desarrollado.

Este simulador tiene parámetros iniciales, es decir, aquellos con los que el simulador comienza a funcionar. Se han definido parámetros de Denavit–Hartenberg de un manipulador articular de 6 grados de libertad, de tal forma que la visualización gráfica en 3D sea adecuada; estos parámetros aparecen en el cuadro 1.

Tomando como referencia [24], se han definido los parámetros dinámicos tomando las tolerancias superiores y modificando aquellos parámetros espaciales para que coincidan con los parámetros de Denavit–Hartenberg propuestos. Los parámetros dinámicos iniciales para el simulador estan dados por las masas m_i de los eslabones, los centro de masa r_i , y las matrices de inercia $I_i = \text{diag}(I_{11}, I_{22}, I_{33})$, que aparecen en los cuadros 2, 3 y 4. Además, los índices de fricción viscosas y de Couloumb son todos cero.

Cabe mencionar que las coordenadas generalizadas, necesarias para la cinemática directa y la dinámica inversa, se calculan por una función llamada *trayectoria.m*. Esta función debe tener como entradas: el número de grados de libertad n , el tiempo de muestreo t_s y el tiempo final de la simulación t_f . Como salida, se obtiene un arreglo tridimensional de $n \times (t_f/t_s) \times 3$, donde se tienen las coordenadas, las velocidades y las aceleraciones generalizadas.

Cuadro 2: Masas de los eslabones

Eslabón	Masa
1	0
2	17.40
3	6.04
4	1.02
5	0.42
6	0.11

Cuadro 3: Centros de masas

Eslabón	r_x	r_y	r_z
1	0	0	0
2	-0.0364	0.0060	-0.0160
3	0.0203	0.0140	0.1430
4	0	-0.0190	0
5	0	0	0
6	0	0	-0.0320

Las fuerzas generalizadas están dadas en Newton-metro (si la unión es rotacional) o en Newton (si la unión es translacional). Las fuerzas generalizadas iniciales para el simulador son igual a cero, es decir, suponiendo que el simulador no tiene actuadores. Para poder cambiarlas, se tendrá que modificar el archivo fuerza.m.

3.1. Experimentación con el simulador

Para evaluar el comportamiento del simulador, se han hecho una serie de experimentos. Estos experimentos tienen como parámetros iniciales los anteriormente descritos, a menos que se indique lo contrario. A continuación se presentan los experimentos realizados y sus resultados.

3.1.1. Experimento: Cinemática directa.

En este experimento, se aprecia el comportamiento cinemático directo del robot manipulador y los parámetros extras para esta simulación son: $t_s = 0,1$ y $N = 6$. Las coordenadas cartesianas de la posición del

Cuadro 4: Inercia

Eslabón	i_{11}	i_{22}	i_{33}
1	0	0.35	0
2	0.134	0.550	0.55
3	0.200	0.220	0.016
4	$2,7 \times 10^{-3}$	$1,95 \times 10^{-3}$	$2,7 \times 10^{-3}$
5	$0,45 \times 10^{-3}$	$0,60 \times 10^{-3}$	$0,45 \times 10^{-3}$
6	$0,23 \times 10^{-3}$	$0,23 \times 10^{-3}$	$0,06 \times 10^{-3}$

Figura 2: Coordenadas cartesianas de la posición del efector final.

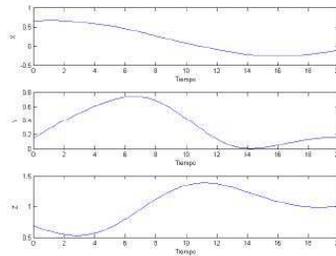
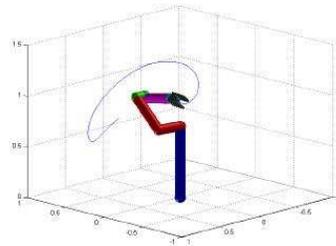


Figura 3: Visualización 3D del robot y trayectoria generada.



efector final se presenta en la figura 2, mientras que la visualización 3D del robot y la trayectoria seguida por el efector final se aprecia en la figura 3.

4. Conclusiones

En el presente trabajo se muestra el desarrollo del *Simulador de Robots Manipuladores I*. Se han implementado todas las funciones necesarias para el desarrollo de los algoritmos de cinemática directa, dinámica inversa y dinámica directa, además, se cuenta con una interfaz gráfica con visualización en 3D y es posible manejar las funciones del simulador desde la línea de comandos en Matlab.

La validez de los algoritmos utilizados en este simulador se ha constatado en literatura especializada. El código se ha verificado totalmente, y el funcionamiento del programa se ha acreditado a través de una serie de experimentos, llegándose a un análisis de resultados satisfactorio.

5. Agradecimientos

Este proyecto fue financiado por el Instituto Politécnico Nacional por medio del apoyo CGPI 20070788.

Referencias

- [1] R. D. Smith, *Simulation*, cuarta ed. New York, New York: Grove's Dictionaries, July 2000.
- [2] "Camelot - robot offline programming." [Online]. Available: <https://www.camelot.dk/>
- [3] "Easy-rob: 3d robot simulation tool." [Online]. Available: <http://www.easyrob.de/>
- [4] P. I. Corke, "A robotics toolbox for matlab," *Robotics & Automation Magazine, IEEE*, vol. 3, no. 1, pp. 24–32, Mar. 1996.
- [5] G. H. Salazar-Silva, J. C. Martinez-Garcia, and R. Garrido, "Enhancing basic robotics education on the web," in *American Control Conference, 1999. Proceedings of the 1999*, vol. 2, 1999, pp. 1470–1471 vol.2.
- [6] H. Morales and C. Cruz, "Edisim: editor y simulador de robots manipuladores," in *2º Congreso Nacional de Electrónica*, Puebla, México, Sep. 2002.
- [7] D. L. Martínez Vázquez. (2007, Aug.) Simulador de robots manipuladores (SRM-I). [Online]. Available: <http://simulador.robot.manipulador.googlepages.com>
- [8] —, "Simulador de robots manipuladores (SRM-I)," Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, México, D. F., Reporte de trabajo terminal para obtener el título de Ingeniero en Mecatrónica, July 2007.
- [9] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," *Trans ASME J. Appl. Mech*, vol. 23, pp. 215–221, 1955.
- [10] K. S. Fu, R. C. González, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. Singapore: Mc Graw-Hill International Editions, 1987.
- [11] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. Canada: John Wiley & Sons, 1989.
- [12] J. Craig, *Introduction to Robotics: Mechanics and Control*, segunda ed. E.U.A.: Addison-Wesley Publishing Company, 1989.
- [13] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. E.U.A.: CRC Press, 1993.
- [14] D. Orin and M. Walker, "Efficient dynamic computer simulation of robotic mechanisms," *ASME Journal of Dynamic Systems, Measurement and Control*, 1982.
- [15] J. Luh, M. Walker, and R. Paul, "On-line computational scheme for mechanical manipulators," *Trans. ASME, J. Dynamic Systems, Measurements and Control*, vol. 120, pp. 69–76, 1980.
- [16] N. M. Swartz, "Arm dynamics simulationtech," Report CMU-RI-TR-82-17, Carnegie Mellon University, Tech. Rep., November 1982.
- [17] R. Featherstone and D. Orin, "Robot dynamics: Equations and algorithms," in *Robotics and Automation, Proc. of the IEEE International Conference on*, San Francisco, CA, 2000, pp. 826–834.
- [18] K. Lee and P. A. Fishwick, "Dynamic model abstraction," in *WSC '96: Proceedings of the 28th conference on Winter simulation*. ACM Press, 1996, pp. 764–771.
- [19] P. A. Fishwick, "A taxonomy for simulation modeling based on programming language principles," *IIE Transactions*, vol. 30, no. 9, pp. 811–820, 1998.
- [20] J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms*. E.U.A.: Springer, 1997.
- [21] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, Massachusetts: The MIT Press, 1990.
- [22] M. W. Walker and D. E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *Dynamic Systems, Measurement and Control, ASME Journal on*, vol. 104, pp. 205–211, 1982.
- [23] J. R. Dormand and P. J. Prince, "A family of embedded runge- kutta formulae," *Computational and Applied Mathematics, Journal of*, vol. 6, pp. 19–26, 1980.
- [24] B. Armstrong, O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the puma 560 arm," in *Robotics and Automation, Proc. of the IEEE International Conference on*, vol. 3, Apr. 1986, pp. 510–518.