

ANÁLISIS DE COLISIONES EN UN SIMULADOR DINÁMICO DE TAREAS DE TELEOPERACIÓN

Sabater, JM., Azorín, JM., García, N., Neco, R., Reinoso, O.
Universidad Miguel Hernández
Departamento de Ingeniería, Campus de Elche
Avda. del Ferrocarril s/n 03202 Elche (Alicante) España
j.sabater@umh.es

Resumen

La detección de colisiones y generación de fuerzas virtuales en simuladores robóticos es un tema de vigente actualidad gracias, entre otras causas, a los avances en la capacidad de cálculo de los computadores.

El objetivo de este artículo es presentar los primeros pasos realizados para la obtención de un simulador dinámico de tareas de teleoperación que incluya el modelo dinámico del robot. La adición de esta característica permite realizar estudios de colisiones más realistas, en los cuales se analicen los casos de colisión de algún eslabón intermedio del robot o de autocolisiones, así como el efecto de las mismas sobre las articulaciones del maestro.

Palabras Clave: Teleoperación, simulación dinámica, colisiones, generación de fuerzas.

1 INTRODUCCIÓN

La detección de colisiones y la generación de fuerzas virtuales en simuladores robóticos es un tema de vigente actualidad gracias, entre otras causas, a los avances en la capacidad de cálculo de los computadores. La utilización de tecnología *haptics* de simulación dinámica proporciona importantes ventajas adicionales dentro de una arquitectura de teleoperación [1]. Sin embargo, la mayoría de estos estudios están basados en el análisis de la dinámica de un punto virtual ficticio (acople virtual [2] o virtual proxy [3]) que interactúa con el entorno dinámico, y que se asume corresponde al extremo del robot teleoperado. Es decir, no se incluye el modelo dinámico del robot esclavo en el simulador, y por tanto, el estudio de la generación de fuerzas virtuales queda restringido a las fuerzas de contacto en el extremo del robot [4] [5].

El objetivo de este artículo es presentar los primeros pasos realizados para la obtención de un simulador dinámico de tareas de teleoperación que incluya el modelo dinámico del robot. La adición de esta característica permite realizar estudios de colisiones

más realistas, en los cuales se analicen los casos de colisión de algún eslabón intermedio o de autocolisiones, así como el efecto de las mismas sobre las articulaciones del maestro.

Se pretende construir un simulador de tareas teleoperadas independiente de los dispositivos utilizados como maestro y esclavo en el esquema. Esta característica permite implementar sobre este simulador diferentes esquemas de control a un nivel más bajo que la utilización de los propios controles de los dispositivos maestro-esclavo cerrados.

En primer lugar, en el apartado 2, se presenta un breve repaso sobre la simulación dinámica en entornos virtuales y sobre diferentes formulaciones y soluciones matemáticas del modelo de mecanismos multivariables y fuertemente acoplados. En el apartado 3 se muestra un simulador desarrollado capaz de modelar fuerzas en la interacción del robot con el entorno y que permite estudiar la influencia de las mismas en las diferentes articulaciones del robot. En el apartado 4 se presenta un esquema construido en Matlab/Simulink® con herramientas presentadas anteriormente [6], que incluye los modelos completos tanto del dispositivo maestro como del esclavo, en el que se ha implementado un control posición-posición. Además se muestran los resultados de simulaciones de colisiones del robot esclavo con su entorno y el efecto de las mismas sobre el robot maestro. Finalmente se esbozan los pasos futuros a realizar en la presente investigación.

2 DINÁMICA COMPUTACIONAL DE ENTORNOS VIRTUALES

El campo de la dinámica computacional está en continuo avance, a la vez que la mejora en las prestaciones de cálculo de los computadores actuales permite la utilización de anteriores formulaciones que permanecían en desuso. La amplísima literatura ofrece tantas vías de desarrollo para el investigador que puede ocasionar al principio un cierto desconcierto. En primer lugar se deben considerar los requerimientos temporales del simulador. Un simulador interactivo precisará de una formulación

matemática capaz de resolver las ecuaciones de movimiento en tiempo real, mientras que en un simulador off-line el compromiso tiempo de cálculo – facilidad de implementación y de modificación es menos restrictivo.

Este apartado trata de resumir brevemente algunos aspectos que deben analizarse previamente a la implementación de un simulador dinámico. Para ello se consideran los siguientes pasos para la creación de un simulador dinámico:

2.1 ELECCIÓN DE PARÁMETROS Y VARIABLES: MODELO MATEMÁTICO

Los sistemas articulados están caracterizados por la presencia de restricciones, o relaciones impuestas entre las variables de movimiento y/o la configuración cinemática del modelo. La elección de la formulación matemática, de las variables y de los parámetros del modelo puede hacerse teniendo en cuenta las siguientes subdivisiones:

- Modelo dinámico. Su resolución proporcionará la posición y velocidad de los eslabones del mecanismo.
- Modelo geométrico. La manipulación de la información geométrica permite la detección de colisiones.
- Modelo de contacto. Calcula las fuerzas de interacción con el entorno.

2.1.1 Construcción del modelo: Formulaciones básicas.

La base del modelo del entorno dinámico reside en la elección de los parámetros, variables y las relaciones entre ellos que definen el comportamiento físico de los objetos modelados. Aunque la formulación de las leyes de Newton en términos de posición produce ecuaciones diferenciales de segundo orden, estas mismas leyes pueden escribirse en términos de posición y velocidad, eligiendo a éstas como variables de estado, resultando así ecuaciones diferenciales de primer orden [7].

La tabla 1 resume la elección de las variables de estado en la formulación de Newton-Euler (NE), en la formulación de Lagrange (LE) y la formulación de Kane (K):

Tabla 1: Opciones para el vector de estado

	Variables de posición	Variables de velocidad	Número de restricciones
NE	$(x, y, z, \theta_1, \theta_2, \theta_3)_i$ (i=1,...,v)	$(\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)_i$ (i=1,...,v)	M+m
LE	q_i (i=1,...,n)	\dot{q}_i (i=1,...,n)	m
K	q_i (i=1,...,n)	u_i (i=1,...,p)	0

Donde v son los cuerpos presentes, M las restricciones holonómicas y m las restricciones no holonómicas, por lo tanto $n=6v-M$ y $p=n-m$ corresponde al número de velocidades generalizadas independientes.

En la literatura de dinámica multicuerpo se encuentran múltiples análisis y relaciones entre estas formulaciones básicas, como el método de partición de coordenadas [8].

2.2 PLANTEAMIENTO DE LAS ECUACIONES DE MOVIMIENTO

Cada una de las formulaciones anteriores y sus derivadas producen ecuaciones de movimiento con una estructura diferente.

Las fuerzas y pares que aparecen en las ecuaciones de NE incluyen, no solo las fuerzas aplicadas, sino también las fuerzas de contacto y las fuerzas articulares. La forma más simple de determinar la fuerza de interacción entre dos cuerpos consiste en unir ambos cuerpos mediante un muelle o un conjunto muelle-amortiguador, calculándose la fuerza de interacción como una función del desplazamiento del muelle-amortiguador de acuerdo a las leyes de Newton. Estos métodos son conocidos como métodos de penalización, debido a que penalizan la interferencia entre objetos. La simplicidad de estos métodos ha hecho que la mayoría de simuladores dinámicos pertenezcan a esta categoría. Sin embargo, si la constante de rigidez del acople se aumenta demasiado en un intento de simular un contacto con un cuerpo rígido, el sistema de ecuaciones se vuelve numéricamente mal condicionado.

La utilización de coordenadas generalizadas, tanto dependientes (método de Lagrange) como independientes únicamente (método de Kane) permite escribir ecuaciones de movimiento de menor dimensión.

Sin embargo un factor importante en la simulación de entornos dinámicos es la facilidad para reformular las ecuaciones de movimiento dependiendo de los contactos producidos. NE permite una implementación sencilla de restricciones unilaterales que sólo tienen efecto cuando se ha producido la colisión, pero LE necesita de la utilización del método de los multiplicadores de Lagrange para modelar contactos intermitentes. Por último, la formulación de Kane necesita de una reformulación completa de las ecuaciones de movimiento on-line, al no estar preparada para incorporar nuevas restricciones que modifiquen el número de variables linealmente independientes.

3 ELECCIÓN DEL SIMULADOR: GENERACIÓN DE FUERZAS DE CONTACTO EN ENTORNOS VIRTUALES

Debido a que se pretende estudiar el efecto de las colisiones en eslabones intermedios del robot esclavo en simulaciones off-line, se ha optado por la simplicidad de la formulación de NE y un método de penalización para estudiar el contacto producido.

Como se ha dicho anteriormente, utilizando esta formulación no se puede analizar el contacto con sólidos fijos e indeformables cuyo modelo de contacto implique una constante de rigidez muy alta. Sin embargo es sencillo utilizar un conjunto muelle-amortiguador para modelar un impacto que provoque un cambio en la cantidad de movimiento en los eslabones del robot. El modelo de contacto utilizado es el siguiente:

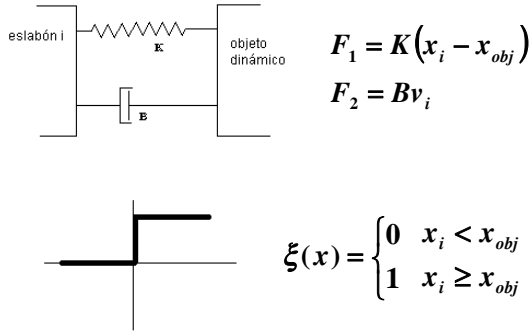


Figura 1: modelo de penalización

Por tanto, la variación brusca de la cantidad de movimiento que se produce en una colisión, se traduce en la aparición de un vector de fuerzas T generalizado ($F_x, F_y, F_z, M_x, M_y, M_z$) aplicado sobre el eslabón i que ha sufrido la colisión.

Para poder calcular las fuerzas que aparecen en las articulaciones del robot debido a la colisión se utiliza la jacobiana del robot. A partir de la definición de jacobiana de un robot:

$$\dot{x} = J(q)\dot{\phi} \quad (1)$$

y realizando un simple balance de potencia, es decir, igualando la potencia expresada en el espacio cartesiano y expresada en el espacio articular:

$$T^T \dot{x} = \tau^T \dot{\phi} \quad (2)$$

es inmediato llegar a la conocida expresión:

$$\tau = J^T T \quad (3)$$

que relaciona los pares generalizados en las n articulaciones del robot (τ) con los pares generalizados en el extremo del robot (T).

Por tanto, en el caso de que la colisión no se produzca en el extremo del robot, sino que sea un eslabón intermedio el que colisiona, bastará con suponer como un único sólido rígido todos los eslabones desde el que colisiona hasta el extremo final, trasladar la fuerza de impacto a una fuerza en el extremo final de ese eslabón ficticio y aplicar la ecuación (3) para el nuevo robot. Al utilizar el algoritmo recursivo de NE en la formulación, la propagación hacia delante de las velocidades trasladará el efecto de la colisión a los eslabones que han formado parte del sólido rígido ficticio, y en el resto de las articulaciones aparecerán unos pares dados por τ .

4 MODELADO DE UN ESQUEMA DE CONTROL POSICIÓN- POSICIÓN EN MATLAB

Un sistema teleoperado permite la realización de tareas remotas sin riesgo para el operario. Básicamente un sistema teleoperado se compone de un maestro y un esclavo. El maestro es el dispositivo sobre el que actúa el operador para realizar la tarea en el entorno remoto. El maestro genera las órdenes que son enviadas al esclavo, que es un robot que interactúa con el entorno remoto reproduciendo los mandatos recibidos del maestro. Utilizando las herramientas presentadas en trabajos anteriores [6] y resumidas en la tabla 2, se implementa el modelo posición-posición de 2 gdl con maestro y esclavo de dinámicas idénticas. La razón de utilizar dinámicas idénticas es la facilidad para obtener, mediante prueba y error, los valores de los reguladores PID incluidos en el modelo, para que el sistema completo tenga un comportamiento estable cuando se crucen las informaciones del maestro y del esclavo.

Tabla 2: resumen de las herramientas Matlab

Función Matlab®	Breve descripción
Denavit.m	Devuelve la matriz de transformación homogénea 4x4 a partir de DH.
Directkine.matic.m	Resuelve la cinemática directa con matrices de transformación homogénea
Inversekine.matic.m	Resuelve la cinemática inversa por métodos geométricos
newtoneule.r.m	Resuelve la dinámica inversa calculando el vector de pares/fuerzas en las articulaciones del robot tomando como entrada las coordenadas articulares y sus derivadas. [9]
Walkerorin.m	Calcula las aceleraciones en las articulaciones a partir de los datos de las posiciones articulares, las velocidades y los pares aplicados, utilizando el tercer método de Walker y Orin [10]
jacobiana.m	Calcula la matriz jacobiana en función de las coordenadas articulares actuales. [11]
pares.m	Función que convierte las fuerzas en el extremo del manipulador en fuerzas articulares.

Las figuras 2 y 3 muestran el esquema general de teleoperación utilizado y el modelo del robot esclavo. Como se observa, la señal de referencia de posiciones articulares deseadas se compara con la señal realimentada de las posiciones articulares del esclavo obteniéndose una señal de error que se suma a la señal de referencia, generándose así la entrada del maestro. Esto último permite modelar de una forma sencilla pero realista el comportamiento del operador, que tratará de generar la señal de entrada necesaria para cumplir con su objetivo de posición.

También se ha añadido una entrada de pares perturbadores en el esclavo, de forma que se pueda

utilizar para simular las fuerzas de reacción generadas en la interacción del esclavo con el entorno. Estos pares se añaden a los generados por los motores del esclavo y conforman la señal de entrada al modelo dinámico directo del robot (bloque Robot2GDL). El bloque completo del modelo del dispositivo maestro o esclavo permite obtener en cada instante de simulación, tanto los pares en las articulaciones del mecanismo, como las posiciones (coordenadas articulares) del mismo, siendo inmediata la implementación de otro esquema de teleoperación diferente al estudiado aquí.

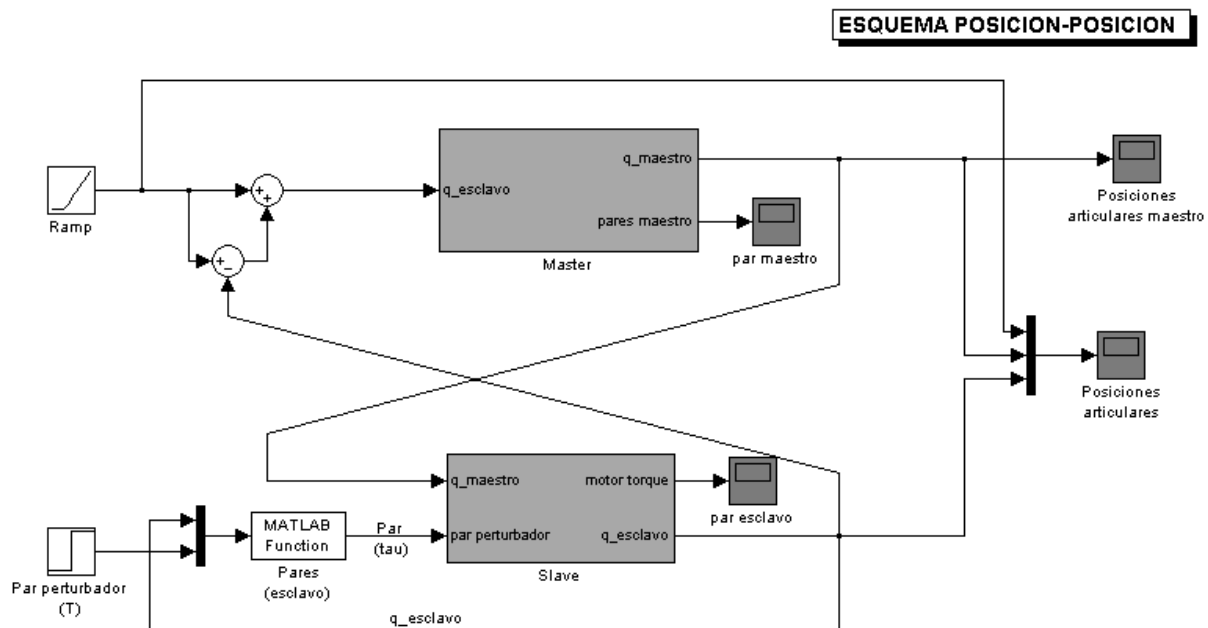


Figura 2: Esquema global posición-posición 2 gdl

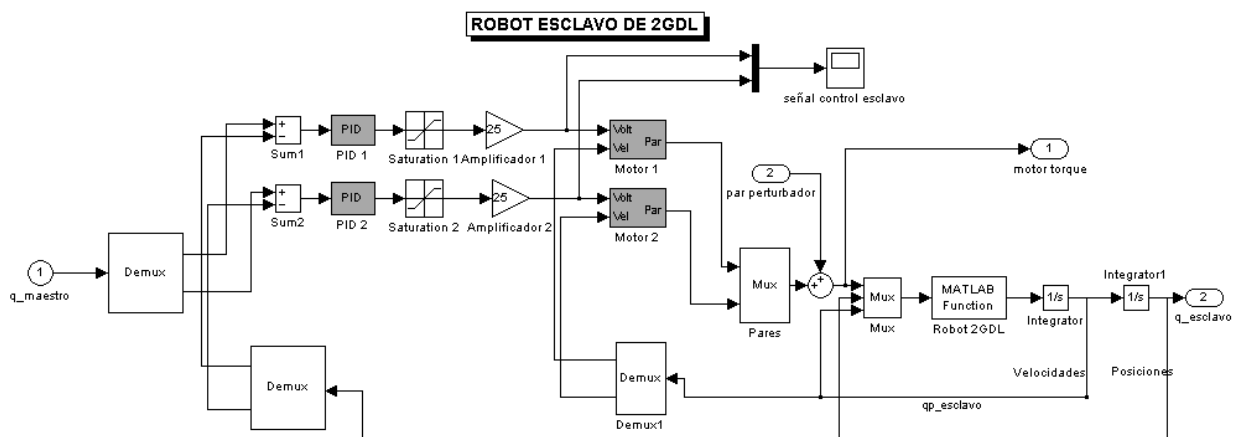


Figura 3: Esquema completo del dispositivo esclavo de 2 gdl

4.1 SIMULACIONES

Para la solución numérica de las simulaciones se ha utilizado un integrador de paso variable (ode45 de Simulink®).

Se ha simulado el seguimiento de una rampa de distinta pendiente en cada una de las coordenadas articulares del maestro y el efecto de la aparición en un instante determinado ($t = 0,05$ seg.) de un vector de fuerzas generalizado (\mathbf{T}) que produzca unas fuerzas de reacción en las articulaciones del esclavo que puedan ser vencidas por los actuadores de las articulaciones del robot. Este caso corresponde al contacto del robot con un objeto dinámico móvil que a partir de ese instante se mueve con el robot. Por tanto corresponde a una variación repentina de la constante de reflexión del entorno. Las gráficas muestran que la señal de control de salida de los reguladores tanto del esclavo como del maestro sufre un cambio brusco provocando en los actuadores pares que tratan de compensar los efectos del par perturbador. Es interesante observar la aparición de pares en las articulaciones del maestro como respuesta a la colisión ocurrida en el entorno remoto.

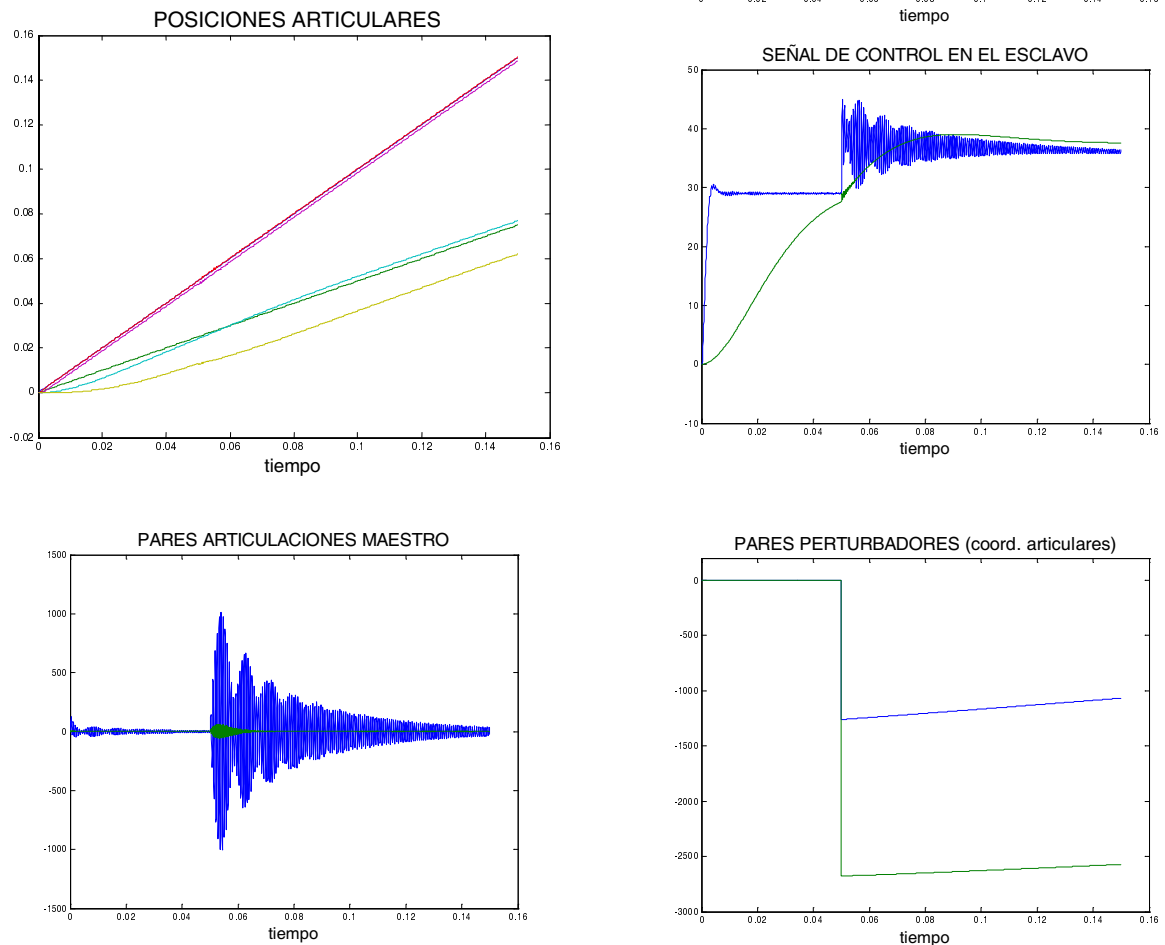


Figura 4: resultados de las simulaciones

También se ha simulado el caso correspondiente a la colisión con un muro rígido que provoque unos pares de respuesta incapaces de ser vencidos por los actuadores del esclavo. El resultado obvio obtenido ya es conocido en la literatura especializada [12]. Con el modelo sencillo de operador utilizado, en el cual se intenta seguir la rampa de posiciones articulares, la aparición de una impedancia muy alta (tendiente a ∞) en el entorno remoto provoca la inestabilidad del sistema.

5 CONCLUSIONES

Se han utilizado herramientas previamente desarrolladas para modelar un entorno de teleoperación completamente abierto sobre el cual se puede simular el efecto de la aparición de colisiones en eslabones intermedios de un robot serial. Se hace patente en las simulaciones la necesidad de “limitar” la impedancia que puede reflejarse en el sistema bilateral. Varios investigadores trabajan en el diseño de nuevas y mejoradas arquitecturas de control bilateral que permitan “acotar” el sistema e impedir la inestabilidad del mismo. La implantación de estos esquemas en herramientas como la presentada en este artículo puede ser de gran utilidad.

También se ha empezado a trabajar en la obtención de un modelo del robot más completo utilizando formulaciones de dinámica multicuerpo presentadas en el apartado 2 que permita la generalización de las herramientas a robots no seriales.

Agradecimientos

El trabajo presentado en este artículo se ha realizado gracias a la financiación aportada por el MCyT para el desarrollo del proyecto de investigación ‘Modelado de espacios virtuales para entrenamiento de sistemas teleoperados en entornos dinámicos’ - DPI2001-3827-C02-02

Referencias

- [1] Burdea, G.C. (1999) “The Sinergy Between Virtual Reality and Robotics” *IEEE Transactions on Robotics and Automation*. Vol. 15, n°3, June 1999
- [2] Brown, J.M, Colgate, E. (1997) “ Passive implementation of multibody simulations for haptic display” *Proceedings of the 1997 ASME International Mechanical Engineering Congress and Exhibition*.
- [3] Ruspini,D. Khatib,O. (1998) “Dynamic models for haptic rendering systems” *Advances in Robot Kinematics: ARK'98, June 1998, Strobl/Salzburg, Austria*, pp523-532.
- [4] Ruspini, D., Krasimir K., Khatib O. “The Haptic Display of Complex Graphical Environments.” *SIGGRAPH 97 Proceedings, (August 1997)*, pp. 345-352.
- [5] Dupont, P. Schulteis, T. Millman, P. Howe, R (1999) “Automatic identification of environment haptic properties” *PRESENCE:Teleoperators and virtual environments*, 8(4), August, 392-409.
- [6] Sabater,JM (2001) “Issues on the simulation of teleoperated systems” *International Conference on Robotics and Automation 2001*.
- [7] Gillespie, R.B., Colgate, E.(1997) “A survey of multibody dynamics for virtual environments” *Proceedings of the IMECE 1997*
- [8] Wehage, R.A., Haug, E.J. “Dynamic analysis of mechanical systems with intermittent motion” *Transactions of the ASME vol 104, pp 778-784, October 1982*
- [9] Saltarén, R. Azorín, JM, Almonacid, M. Sabater, JM. (2000) “Prácticas de robótica utilizando Matlab”, UMH-Limencop, España.
- [10] Walker, M. W., Orin, D. E., (1982) “Efficient Dynamic Computer Simulation of Robotic Mechanisms”, *Journal of Dynamic Systems, Measurement, and Control*, pp. 205-211
- [11] Asada, H., Slotine, J.J.E. “Robot analysis and control” *John Wiley Interscience Publication (1986)*
- [12] Colgate, J.E. Brown, J.M “Factors affecting the Z-Width of a haptic display” *IEEE 1994 International Conference on Robotics and Automation* pp.3205-10.