

Herramientas MATLAB/SIMULINK para el Control de Fuerza de Robots manipuladores

I. Ruano Ruano*, L.M. Nieto Nieto*, J. Gómez Ortega*, F.R. Rubio**

*Grupo de Robótica, Visión por computador y Automática
Escuela Politécnica Superior de la Universidad de Jaén
juango@ujaen.es

**Ingeniería de Sistemas y Automática
Escuela Superior de Ingenieros de la Universidad de Sevilla
frubio@esi.us.es

Resumen

El Conjunto de Herramientas (Toolbox) de Control de Fuerza es un paquete software que presenta una serie de modelos y bloques Simulink y funciones Matlab con los cuales se pueden estudiar, entender y aplicar, de manera muy sencilla, distintos métodos de control de fuerza en una serie de entornos sobre los que actúa un robot manipulador. En este trabajo se presenta en forma de manual la primera versión del Toolbox y se muestran ejemplos de las distintas técnicas aplicadas mediante la simulación de un robot manipulador RX60 (Stäubli).

Palabras Clave: Control de Fuerza, Control de Rigidez, Control Híbrido, Control de Impedancia, Toolbox de Simulink.

1 INTRODUCCIÓN

En el presente trabajo se presentan una serie de herramientas (funciones, bloques y modelos) que se han desarrollado para la simulación del control de fuerza de robots manipuladores basado en los programas Matlab/Simulink.

Al adquirir el software de Matlab/Simulink existe la posibilidad de instalar una serie de colecciones de funciones altamente optimizadas y específicas de la aplicación llamadas toolboxes. Las toolboxes permiten ampliar la capacidad de Matlab/Simulink en campos específicos (Comunicaciones, Sistemas de control, Bases de datos, Adquisición de datos, Diseño de filtros, etc). Tradicionalmente, la toolbox más usada en el campo de la robótica ha sido y es la creada por Peter Corke [1]. Otros autores ([5]) han desarrollado toolboxes que, basándose en ésta, han desarrollado campos más específicos dentro de la robótica. Sin embargo no dejan de tener un interés e

importancia clave en su campo. La toolbox desarrollada también está incluida en el campo de la robótica, y se enmarca dentro del control de robots manipuladores mediante estrategias de control de fuerza.

1.1. CONTROL DE ROBOTS MANIPULADORES

Tradicionalmente, los métodos de control de los robots manipuladores industriales más empleados han sido y son los basados en el control de posición. Por otro lado, existen otras estrategias de control que no tienen porque tener en cuenta la posición sino que (independientemente de que la utilicen) emplean información basada en la fuerza ejercida en el entorno por parte del robot manipulador para elaborar un control del mismo. A estos métodos se les llama estrategias de control de Fuerza. El toolbox al que se refiere el presente artículo está centrado en este tipo de control. Antes de entrar de lleno en el contenido del mismo es necesario tener presente una serie de ideas sobre este tipo de control que se expondrán en los siguientes apartados, según se vayan explicando los distintos conceptos se irán introduciendo sucesivamente los elementos del toolbox que los modelan o simulan.

2 CONTROL DE FUERZA

Cuando se debe aplicar un control en aplicaciones en las que el manipulador debe mantener contacto o aplicar fuerzas sobre el entorno, suele ser necesario aplicar estrategias de control de fuerza. Con ellas se puede determinar la fuerza que se quiere que el robot ejerza sobre un entorno. Por este motivo, el entorno, y todo lo que ello conlleva, cobra una importancia vital que no tiene cuando el control se realiza en posición. Los elementos que hay que tener en cuenta a la hora de aplicar estos métodos de control son:

- El propio **Robot Manipulador**. En este sentido se han aprovechado modelos y funciones de otros autores [1] y [5].
- La **Estrategia de Control de Fuerza** elegida. Existen distintos métodos de control de fuerza, habiendo implementado en la toolbox las más importantes.
- El **Entorno**. Al ser el medio el que recibe el contacto del robot manipulador
- El **Sistema de Referencia de la Tarea**. De su correcta elección dependerá en gran medida la sencillez de la aplicación.

A continuación se pasa a describir estos elementos y las funciones, bloques y modelos que los implementan en el toolbox.

2.1 EL ROBOT MANIPULADOR

El manipulador utilizado en las simulaciones es un robot Stäubli RX60, que tiene 6 grados de libertad. Para incluir en la simulaciones el modelo cinemático y dinámico del robot RX60 debe definirse la matriz *dyn* del bloque *robot* disponible en la toolbox Hemero [5] y Corke [1], y que se compone de los subsistemas mostrados en la figura 1.

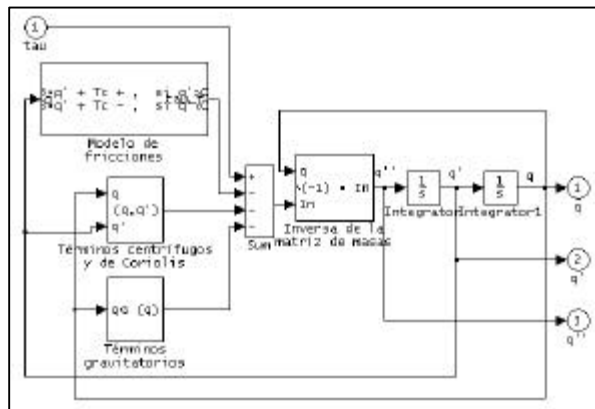


Figura 1: Esquema del bloque *robot*

En esta matriz han de especificarse los parámetros de Denavit-Hartenberg modificados (Craig [2]), tensor de inercia, fricción viscosa, etc. Con este bloque se obtienen los vectores de posición, velocidad y aceleración articulares (q , \dot{q} , \ddot{q}) del robot en función del vector de fuerzas y pares aplicados (τ). Otro de los parámetros que admite el bloque *robot* es la posición y velocidad iniciales.

2.2 ESTRATEGIAS DE CONTROL DE FUERZA

A la hora de hablar de estrategias de control de fuerza se pueden encontrar distintas clasificaciones. Según la forma de obtener la fuerza ejercida se dice que hay métodos de control de fuerza implícita (no existen sensores de esfuerzo y la fuerza se calcula en base a otros parámetros y no directamente) y

métodos de control de fuerza explícita (existe un sensor de esfuerzos que proporciona un valor de la fuerza ejercida en el entorno). La clasificación que aquí se ha utilizado es la siguiente:

- Control de Rigidez (Stiffness Control).
- Control de Fuerza Híbrido.
- Control de Impedancia.

A continuación se pasa a describir brevemente cada uno de los métodos expuestos y a mostrar las herramientas desarrolladas para su uso en Matlab/Simulink.

2.2.1 Control de Rigidez (Stiffness Control).

Controla la rigidez del efector final del robot manipulador mediante un control en posición: la fuerza ejercida depende, de forma implícita, de una referencia de posición (x_d) definida en el interior del entorno. Para determinar con precisión la fuerza aplicada es imprescindible el conocimiento del modelo del entorno, que en este esquema se supone elástico. Una explicación más completa a este tipo de control se puede obtener de [4]. En la figura 2 se puede ver un esquema de este tipo de control.

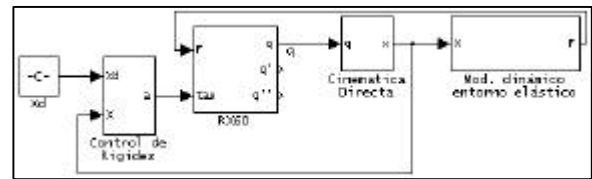


Figura 2: Esquema del control de rigidez

El bloque de control aplica una ley PD que tiene como parámetros la **constante proporcional** (K_p) y la **constante derivativa** (K_d), dadas ambas en forma matricial.

Otras entradas y parámetros de este tipo de esquema son los siguientes:

Vectores de posición y velocidad articulares iniciales del robot manipulador, q_0 y \dot{q}'_0 .

Sirven para situar al RX60 con la posición y velocidad inicial deseadas en el momento inicial. Se introducen en el bloque RX60 y se encuentran en todos los esquemas de control de la toolbox.

Referencia de posición, x_d . Determina la fuerza que se aplica en el entorno de la forma:

$$F \approx K_p(x_d - x_e) \quad (1)$$

Donde K_p es la constante proporcional antes comentada y x_e la posición del entorno. Se introduce directamente tal y como se puede ver en la figura 2, esta entrada aparece en la mayoría de los esquemas del Toolbox.

Posición del entorno, x_e . Sirve para indicar la situación del entorno respecto al sistema de referencia de la tarea, esta entrada se

introduce en el bloque del entorno y debe utilizarse en todos los esquemas de control de la toolbox.

Angulo de inclinación del entorno, α . Este parámetro será descrito en el apartado 2.3 que trata el entorno en el que se desarrolla la aplicación. Debe ser modificado en el bloque cinemática directa y en los bloques de cálculo de jacobianos, utilizados en todos los esquemas de control de la toolbox.

Constante de Rigidez del entorno, K_e . Matriz que determina el comportamiento elástico del entorno. Este parámetro se introduce en el bloque del entorno, cuando un entorno presente características elásticas debe introducirse este parámetro.

Fuerzas de rozamiento de las componentes tangenciales a la superficie, F_{Rv} y F_{Rw} . Que se suponen constantes. Deben ser introducidas en el bloque del entorno, se puede usar en todos los esquemas de control.

2.2.2 Control de Fuerza Híbrido.

Con este esquema se consigue aplicar una fuerza en una dirección mientras se mueve el efector final en otras. Esto se consigue gracias a dos controladores (de fuerza y de posición) que actúan simultáneamente en direcciones espaciales complementarias, de forma que en unas direcciones se realiza control de fuerza y en el resto control de posición, de acuerdo con una matriz de selección. Este método exige un conocimiento preciso de la geometría del entorno y una estrategia adecuada para planificar las tareas de control. Para más información sobre el mismo consúltese [4]. En la figura 3 se puede ver un esquema de este tipo de control.

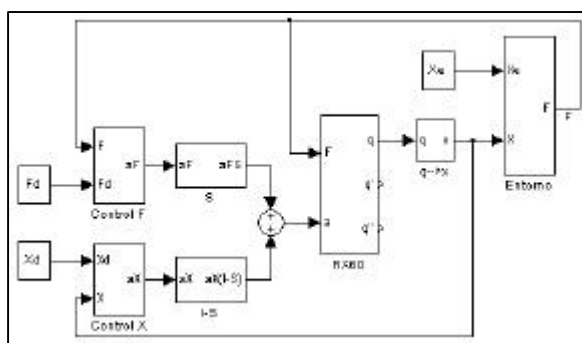


Figura 3: Esquema del control híbrido

Lo más destacable del esquema que muestra la figura 3 es la existencia de 2 matrices de selección complementarias en el espacio de tareas: I-S y S. Son las encargadas de determinar en que direcciones se va a realizar el control en posición y en cuales se hará control en fuerza. Nótese que en este tipo de control la referencia no sólo se da en posición (x_d) sino que también se da en fuerza (F_d). El bloque de control de

Fuerza aplica una ley PD con parámetros matriciales K_p y K_d . El entorno modelado es de tipo elástico (como el descrito para el control de rigidez en el apartado 2.2.1).

2.2.3 Control de Impedancia.

Este tipo de métodos tienen como finalidad controlar la relación entre el movimiento del manipulador y la fuerza que ejerce sobre el entorno, lo que se conoce como impedancia mecánica:

$$Z = \frac{F}{V}, \quad (2)$$

con F, Fuerza ejercida por el robot en el entorno, y V, velocidad del robot.

Dependiendo del objetivo de control el esquema debe incluir una realimentación de fuerza o de posición, denominándose control de admitancia o de impedancia. Se trata de ajustar la impedancia que ofrece el robot manipulador ante un determinado entorno que, a su vez, deberá ser modelado como una impedancia mecánica. En el control de impedancia la magnitud medida es la posición, a partir de la cual se obtiene la velocidad que a su vez se aplicará al controlador. El esquema correspondiente puede verse en la figura 4.

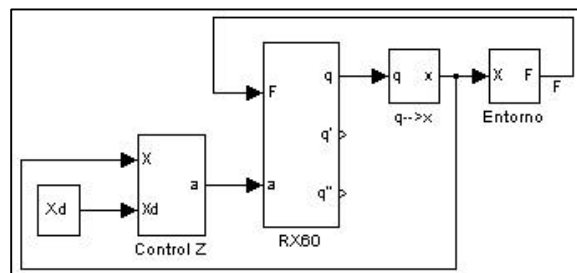


Figura 4: Esquema del control de Impedancia.

Variantes muy conocidas del control de Impedancia son el stiffness control (control de rigidez) y el control híbrido de impedancia. En la figura 5 se muestra un esquema de un control de admitancia, en este caso la magnitud realimentada es la fuerza.

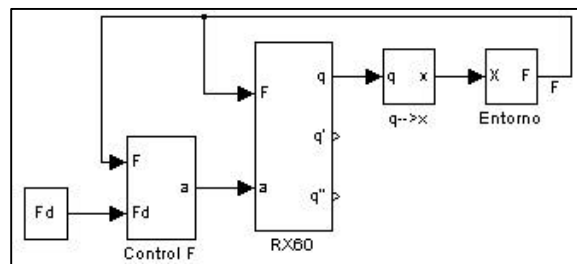


Figura 5: Esquema del control de Admitancia.

Los parámetros más característicos de este tipo de control son las matrices de inercia, amortiguación y

rigidez que presentan el entorno (se introducen en el bloque de entorno: H_e , B_e y K_e) y el robot manipulador (se introducen en el bloque Control Z y Control F: A_z , B_z y K_z). Para obtener más información sobre este tipo de control consúltese [3] o [4].

2.3 EL ENTORNO

Las *características del entorno*, tanto las geométricas como las dinámicas, son importantes cuando se realiza control de fuerza. Por un lado deben ser tenidas en cuenta las dimensiones del medio y su situación respecto al robot manipulador y por otro sus características físicas, y en concreto su impedancia mecánica (inercia, amortiguamiento y elasticidad), determinan su comportamiento ante las fuerzas ejercidas por el manipulador y con ello el comportamiento del manipulador que las ejerce. Según su **impedancia mecánica** se puede establecer una clasificación de entornos dependiendo de la característica predominante en el mismo:

- Entornos Inerciales.
- Entornos Elásticos (Rígidos o Capacitivos).
- Entornos Viscosos (Amortiguados o Resistivos).

En la figura 6 se puede ver un ejemplo de modelo de entorno elástico con rozamiento.

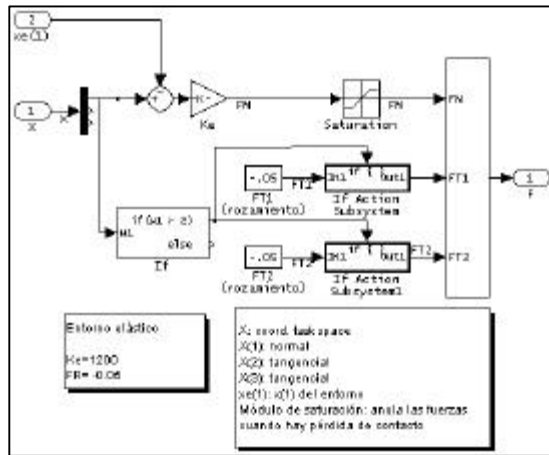


Figura 6: Ejemplo de entorno elástico.

En la toolbox que se presenta se incluyen entornos con distintas impedancias mecánicas como el que se muestra en la figura 7, donde aparece un entorno con impedancia mecánica genérica configurable así como unas tablas donde se puede ver su ecuación dinámica junto a una explicación de todos los elementos que la forman. Además se adjunta una tabla para identificar el tipo de control (de impedancia o admitancia) que resulta adecuado aplicar, dependiendo del tipo de entorno, para obtener una solución estable.

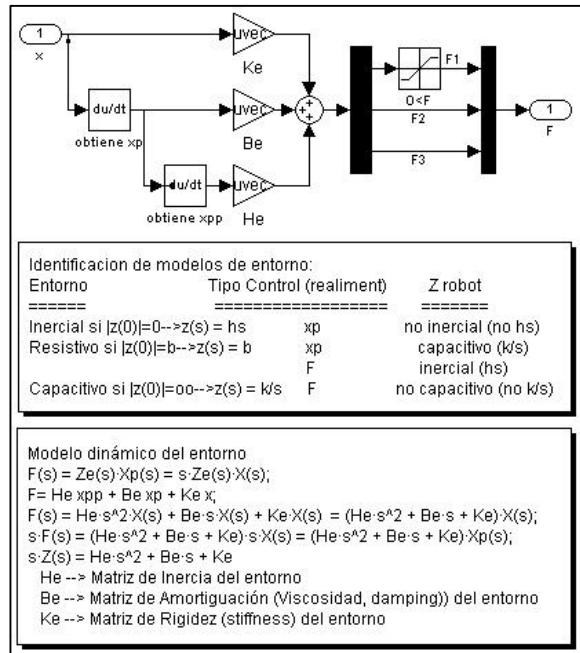


Figura 7: Entorno genérico y tablas explicativas.

En todos los modelos de entorno incluidos aparece como entrada la posición del robot manipulador en el espacio cartesiano de tareas y se obtiene como salida la fuerza ejercida por el manipulador.

En cuanto a las **características geométricas** se ha utilizado un entorno consistente en una superficie plana inclinada un ángulo α , definible por el usuario, con respecto a la horizontal en la que se encuentra el robot manipulador. Dichas características se pueden observar en la figura 8.

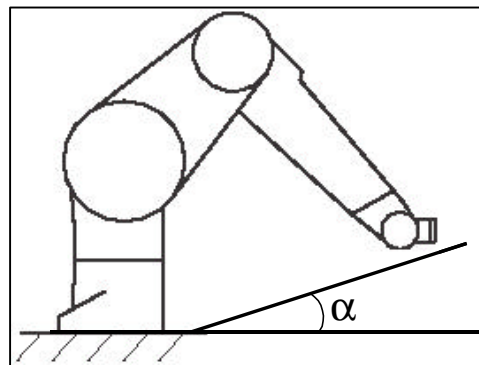


Figura 8: Geometría del entorno.

2.4 EL SISTEMA DE REFERENCIA DE LA TAREA

El *sistema de referencia de la tarea* a realizar debe elegirse de forma que el desarrollo del control y la especificación de objetivos sea lo más fácil posible: Normalmente se suele hacer coincidir las direcciones en las que se quiere que el manipulador ejerza una

determinada fuerza con alguno de los ejes del sistema de referencia. Lo cual suele depender directamente del entorno elegido sobre el que se ejercen fuerzas. En la Toolbox se ha tomado como sistema de referencia el mostrado en la figura 9. Como se puede observar, tiene una componente paralela a la superficie (v), otra permanece ortogonal a la misma (u) mientras que la tercera componente (w) se sitúa de manera que formen un sistema cartesiano ($0,u,v,w$).

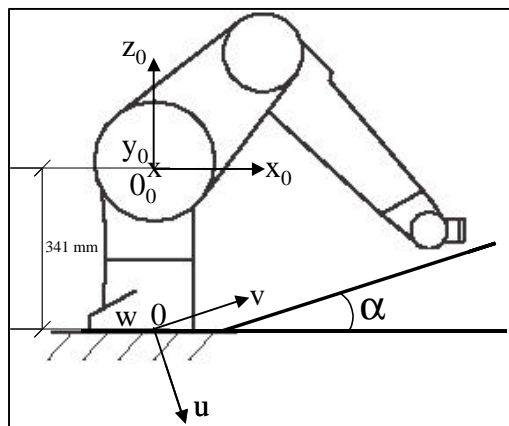


Figura 9: Sistema de referencia de la tarea.

Otra de las cuestiones a tener en cuenta es como afecta el espacio de la tarea a todas las funciones utilizadas. Lo más cómodo para el usuario es dar las referencias (normalmente de posición y/o fuerza) según el espacio de la tarea, así se debe hacer cuando se use la Toolbox. Se emplean bloques S-function que internamente hacen llamadas a funciones en formato matlab (.m), estas funciones son las que se encargan de tener en cuenta las conversiones entre sistemas de referencia y están basadas en los Jacobianos.

3. ESTRUCTURA DE LA TOOLBOX

La presentación de la Toolbox se ha creado estructurándola en 4 bloques principales que se llaman:

- Métodos de Control de Fuerza. Contiene los bloques de control descritos anteriormente, estructurados a su vez de la forma siguiente:
 - Control de Rigidez
 - Control Híbrido
 - Control de Impedancia
- Entorno. Contiene modelos de entorno en 1, 2 y 3 dimensiones (según las dimensiones que se den al espacio de tareas). Hay 4 entornos en cada uno de ellos: genérico, inercial, capacitivo (elástico) y resistivo (viscoso).
- Utilidades. Contiene bloques con S-functions que llaman a funciones Matlab en formato .m

que son necesarias para calcular la cinemática directa, el Jacobiano, etc.

- Ejemplos. Se han realizado una serie de ejemplos que están incluidos en la toolbox que permiten ayudar al uso de la misma.

La figura 10 muestra la estructura de la Toolbox y como está organizados los modelos, bloques y funciones.

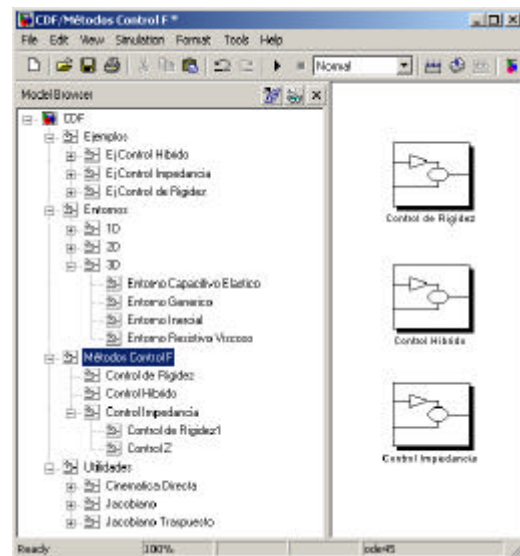


Figura 10: Estructura de la Toolbox de Control de Fuerza.

4. CONCLUSIONES

El conjunto de herramientas presentado en este artículo constituye una primera versión de lo que debe ser una toolbox completa de control de fuerza en entorno Matlab/Simulink donde se puedan encontrar y utilizar entornos, robots y estrategias de control de fuerza más variadas que puedan servir para simular aplicaciones antes de ser realizadas en situaciones reales. Se han simulado una serie de situaciones que más adelante serán probadas en entornos reales con el robot manipulador Stäubli RX60, del que dispone el Laboratorio de Ingeniería de Sistemas y Automática.

Agradecimientos

Este trabajo ha sido financiado por el Proyecto CONPOS (Control de Precisión en Sistemas de Posicionamiento) con código CICYT DPI2001-2424-C02-02.

Referencias

- [1] Corke., P.I., (mar 1996) "A Robotics Toolbox for MATLAB", *IEEE Robotics and Automation Magazine*, nº 1-vol.3 pp. 24-32.

- [2] Craig, J. J., (1989) "Introduction to Robotics, Mechanics and Control", Addison-Wesley, Nueva York.
- [3] Khalil, W y Dombre, E., (2002) "Modeling, Identification & Control of Robots", Hermes Penton, Londres.
- [4] Lewis, F.L., Abdallah, C.T. y Dawson, D.M. (1993) "Control of Robot Manipulators", McMillan, Nueva York.
- [5] Maza, J.I. y Ollero, A., (2001) "HEMERO. Una herramienta MATLAB-Simulink para la Robótica", *CD-ROM, Marcombo*.