

# Prototipado virtual de un controlador por etapas para un brazo robótico en una aplicación AS/RS

Frank Flórez Montes\*

Jhon Alexander Taborda Giraldo\*\*

Fecha de recepción: 13 de septiembre de 2013

Fecha de aprobación: 20 de octubre de 2013

Pp. 163 a 182

## RESUMEN

En este trabajo, se presentan los procesos de diseño y prototipado rápido de un controlado por etapas, para un brazo robótico de un sistema de búsqueda y recuperación automático (AS/RS) de libros para una biblioteca, denominado *Biblio-bot*. El controlador por etapas, se diseñó a través de reglas de decisión secuencial y combinatorial basadas en máquinas de estado y diagramas de flujo del entorno de *Simulink-Stateflow*, el cual permite la generación automática de código de ejecución en tiempo real, además de la validación del diseño en entornos de realidad virtual. El artículo presenta los resultados del proyecto *Biblio-bot* relacionados con el diseño y puesta a punto del proceso de recuperación de libros por parte del brazo robótica, dado que es el proceso más crítico en el funcionamiento del sistema.

## PALABRAS CLAVE

Brazo robótico, prototipado virtual, realidad virtual, control híbrido, máquina de estados.

---

\* Master en Ingeniería - Automatización Industrial, Universidad Nacional de Colombia

\*\* Ingeniero Electrónico, Universidad del Magdalena.

## *Virtual prototype of a controller by stages for a robotic arm in an AS/RS application*

### ABSTRACT

*In this research paper we present the design and rapid prototyping processes of a stage-based controller for a robotic arm in an automatic search and retrieval system (AS / RS) of books in a library, called Biblio-bot. The design of a stage-based controller uses sequential and combinational decision rules based on state machines and flow charts of the Simulink-State flow environment. This toolbox allows automatic code generation at real-time execution and validation of designs in virtual reality environments. We show outcomes of the Biblio-bot project concerning the design and tuning of books' recovery processes using the robotic arm. This process is the most critical task in the Biblio-bot system performance.*

### KEY WORDS

*Robotic Arm, Virtual Prototype, Virtual Reality, Hybrid Control, State Machine, State Flow, Biblio-bot.*

## *Prototype virtuel d'un contrôleur séquentiel pour bras robotisé d'application AS / RS*

### RÉSUMÉ

*Cet article expose les processus de conception et de prototypage d'un contrôleur séquentiel pour bras robotisé. Il s'agit d'un système de recherche et de récupération automatique (AS / RS) de livres pour bibliothèque appelé Biblio-bot. Le contrôleur séquentiel a été conçu sur la base de machines d'état et de diagrammes de flux extraits d'un environnement Simulink-Stateflow permettant à la fois une création automatique et une exécution de code en temps réel mais également une validation conceptuelle en environnement de réalité virtuelle. L'article présente les résultats du projet Biblio-bot associé à la conception et au développement du processus de récupération de livres par bras robotisé. Il s'agit là du processus essentiel dans le fonctionnement du système Biblio-bot.*

### Mots-clés

*Bras Robotisé, Prototypage Virtuel, Réalité Virtuelle, Machine d'Etat, Contrôle Hybride.*

## *Prototipo virtual de um controlador por etapas para um braço robótico em uma aplicação AS/RS*

### RESUMO

*Este trabalho apresenta os processos de desenho e prototipo rápido de um controlador por etapas para um braço robótico de um sistema de busca e recuperação automático (AS/RS) de livros para uma biblioteca, chamado Biblio-bot. O controlador por etapas foi desenhado a través de regras de decisão sequencial e combinacional baseadas em máquinas de estado e diagramas de fluxo do entorno de Simulink-Stateflow, que permite a geração automática de código de execução em tempo real, assim como a validação do desenho em entornos de realidade virtual. Este artigo apresenta os resultados do projeto Biblio-bot relacionados ao desenho e ajuste do processo de recuperação de livros usando o braço robótico, porque este é o processo mais crítico no funcionamento do sistema Biblio-bot.*

### PALAVRAS-CHAVE

*Braço Robótico, Prototipado Virtual, Realidade Virtual, Controle Híbrido, Máquina de Estados.*

# 1. Introducción

---

**P**oder analizar y entender un proceso complejo, por medio de las pequeñas partes que lo conforman es una de las estrategias más utilizadas en ingeniería. Para esto, se debe contar con herramientas especializadas que permitan descomponer cualquier movimiento o proceso en sus factores mínimos y siempre estar explorando nuevas herramientas que faciliten esta tarea, como es el caso del *Toolbox StateFlow* (SF) de MATLAB( Mathworks, 2003). Básicamente funciona como un editor de máquinas de estado controlado por parámetros internos y/o externos en un modelo de *Simulink*.

Para esta investigación, se utilizó esta herramienta junto con el *Toolbox* de Realidad Virtual (VR) (Mathworks, 2004), para caracterizar, modelar y controlar el movimiento de un brazo robótico de un sistema de búsqueda y recuperación automático (AS/RS), aplicado en una biblioteca para optimizar parámetros como la utilización del espacio y el tiempo de búsqueda. Con el control por etapas, se busca garantizar la facilidad de analizar el sistema completo o individualmente y la posibilidad de ajustar o ampliar las etapas según las necesidades del sistema, especialmente en una biblioteca donde fácilmente se pueden adicionar nuevos elementos como libros y estanterías. El objetivo principal es la modelación del movimiento del brazo robótico además de evaluar la efectividad del control por etapas para manipular todas sus articulaciones, dejando de lado los parámetros técnicos del sistema electrónico y los demás parámetros de los sistemas AS/RS.

## 2. Metodología



**E**l diseño modular, es un concepto ampliamente utilizado en la construcción de equipos electrónicos debido a las posibilidades de análisis que brinda en comparación con los sistemas integrados, ya que es posible rastrear fallas estudiando uno a uno los módulos del sistema. Si alguno presenta un problema basta con cambiar esa única parte y no todo el equipo completo. Este método, puede ser utilizado en el diseño de movimientos en sistemas electromecánicos, en donde la precisión depende del correcto movimiento de todas sus partes, tanto las partes electrónicas como la estructura mecánica general, que a menudo tiene problemas físicos por acción de inercias y aceleraciones tangenciales.

El presente estudio, se concentra en el diseño por etapas del movimiento de un brazo robótico mientras recupera un libro, del cual solo se conoce su posición, el actuador debe desplazarse hasta ella, coordinar el movimiento de sus articulaciones para extraer de forma segura el libro deseado y finalmente, acercarlo a la zona de descarga.

Este proceso fue descompuesto en las etapas generales (Figura 1), siendo la etapa dos, la más importante en el proceso general por lo que se enfatiza en su explicación y desarrollo.

**Figura 1. Etapas generales**

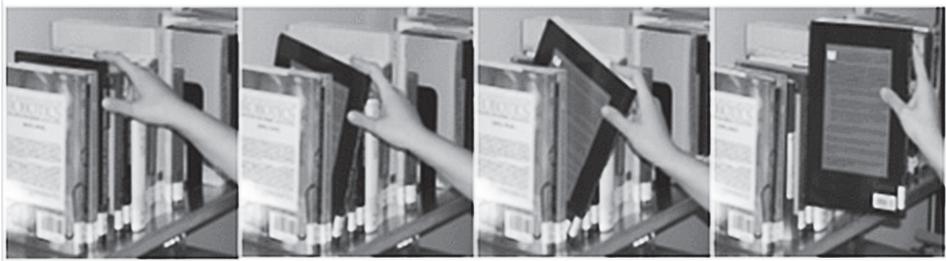


Fuente. Elaboración propia del autor.

## 2.1 Proceso de recuperación

El éxito del proceso se mide con la adquisición del libro de forma fácil y segura, por lo que se buscó imitar el método (Figura 2), donde el dedo índice se ubica en la parte superior del libro y se tira hacia la parte de atrás; luego, con los dedos medio y pulgar, se toma de los lados y se realiza la extracción del texto correspondiente (Prats, 2009) (Minzhou, 2009).

**Figura 2. Recuperación manual**

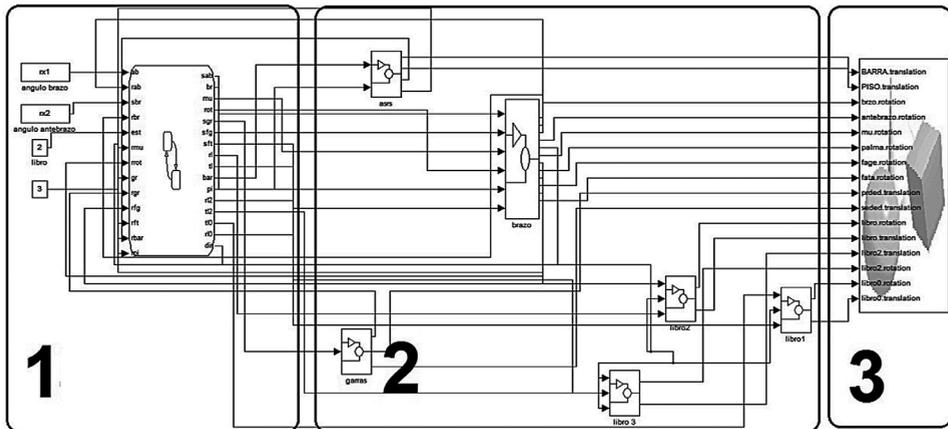


Fuente. Prats (2009).

Con ese objetivo, se programó el modelo en *Simulink* (Figura 3), calculando las posiciones para el brazo con el algoritmo de Denavit-Hartenverg (Craig, 2006), (García, 2001).

La máquina de estados se construyó con el *Toolbox Stateflow* como controlador central (recuadro 1 de la Figura 3), enviando datos de posición hacia el mundo virtual (recuadro 3) a través de una serie de subsistemas de acople conformados por integradores (Ogata, 2003) y constantes de condiciones iniciales (recuadro 2) que cumplen la función de traducir niveles de señal, a coordenadas espaciales para el mundo virtual.

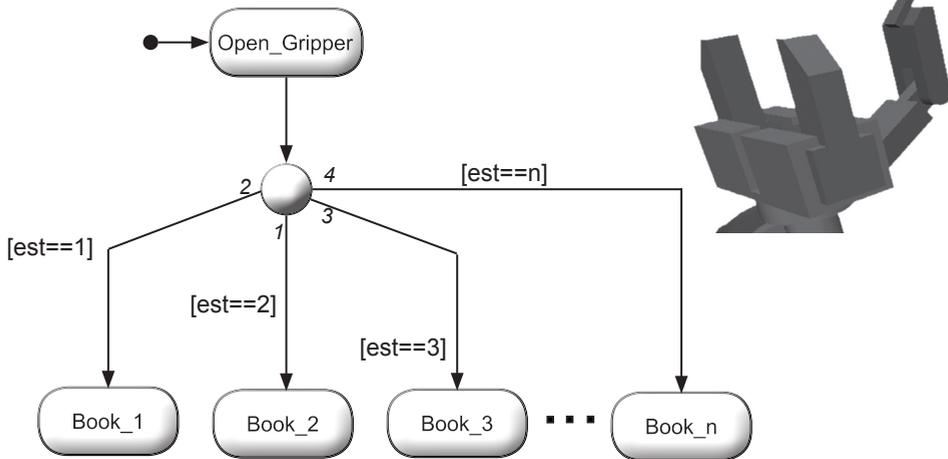
**Figura 3. Modelo en Simulink**



Fuente. Flórez y Taborda (2012).

Dentro de la máquina de estados, existen una serie de bloques que definen las etapas del movimiento, siendo cada una independiente de la otra y habilitadas por posiciones espaciales alcanzadas por las articulaciones en el mundo virtual. Una de las acciones necesarias para el funcionamiento de todo el sistema, es la apertura de la pinza, esta acción se realiza en la primera etapa denominada *Open\_Gripper*.

Figura 4. Apertura de la pinza



Fuente . Elaboración propia del autor.

Una vez se ha completado la posición inicial de la pinza, se procede a ubicar el brazo frente a uno de los libros disponibles, el usuario debe ingresar el numero del libro que quiere y así el sistema define la ruta hacia *book\_1*, *book\_2* y *book\_3* o cualquier otro libro, agregado posteriormente como *book\_n*. Dentro de todos los bloques, se maneja la misma estructura para ubicar las articulaciones según las ecuaciones del algoritmo de Denavit-Hartenverg, teniendo en cuenta que las medidas del cuerpo, brazo, antebrazo y muñeca son  $l_0=15\text{cm}$ ,  $l_1=15\text{cm}$ , y  $l_2=12\text{cm}$ ,  $l_3=5\text{cm}$  respectivamente; de esta forma se procede. definir las matrices de transformación homogénea asumiendo las igualdades  $C_i=\cos\theta_i$  y  $S_i=\text{sen}\theta_i$ , donde  $i=[1,4]$  especifica el segmento con el que se está trabajando (cuerpo, brazo, antebrazo o muñeca).

$${}^0A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^1A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_1C_2 \\ S_2 & C_2 & 0 & l_1S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^2A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & l_2C_3 \\ S_3 & C_3 & 0 & l_2S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^3A_4 = \begin{bmatrix} C_4 & -S_4 & 0 & l_3C_4 \\ S_4 & C_4 & 0 & l_3S_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Desarrollando la multiplicación de matrices  $T=^0A_1 * ^1A_2 * ^2A_3 * ^3A_4$  como implica el método, se llega a las ecuaciones (5), (6) y (7) que definen la posición final del actuador y los ángulos de cada una de sus articulaciones.

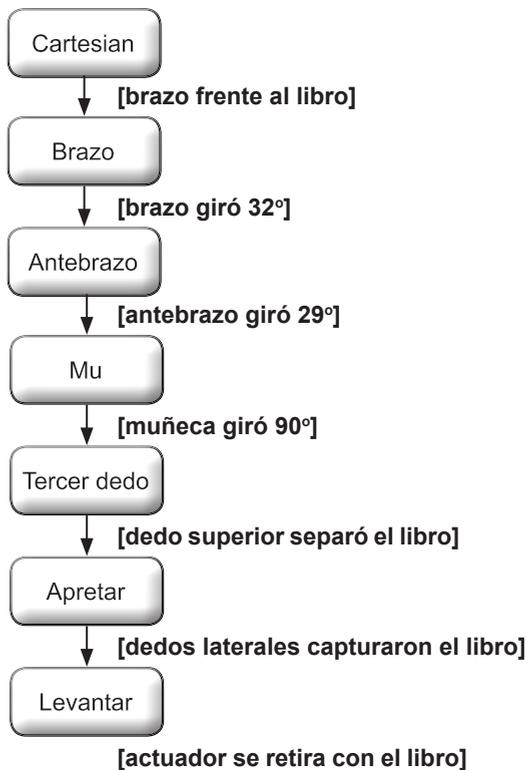
$$p_x = C_1 \{C_2 L_2 + C_4 L_4 [C_2 (C_3 - S_3) - S_2 (C_3 + S_3)] + L_3 (C_2 C_3 - S_2 S_3)\} \quad (5)$$

$$p_y = S_1 \{C_2 L_3 (C_2 + L_2 / L_3 - (S_2 S_3) / S_1) + C_4 L_4 [C_2 (C_3 - S_3) - S_2 (C_3 + S_3)]\} \quad (6)$$

$$p_z = L_1 + L_2 S_2 + C_4 L_4 [S_3 (C_2 - S_2) + C_3 (S_2 + C_2)] + L_3 (C_2 S_3 + C_3 S_2) \quad (7)$$

Una vez obtenidos los datos de posición, se programa la estructura básica de todos los bloques (figura 5), por lo que si se requieren más libros solo se deben duplicar los bloques existentes y modificar sus posiciones finales según las ecuaciones descritas anteriormente.

Figura 5. Estructura básica



Fuente. Elaboración propia del autor.

Cuando se conoce la posición, se procede a definir cómo debe moverse el actuador intentando disminuir los problemas de precisión obtenidos por la inercia del sistema mecánico, así que se decidió que todos los bloques de este subsistema utilizaran un método de velocidad variable en el cual se buscara que la velocidad del objeto tienda a formar una campana de *Gauss*, acelerando la articulación hasta llegar a la mitad del recorrido de la posición necesitada y de ahí en adelante, su velocidad tienda a cero a medida que se acerca a su posición final.

## 2.2 Funcionamiento de la estructura básica

El primer bloque del subsistema es comúnmente llamado Cartesian, en donde se posiciona el brazo frente al libro (Ranferi, 2007). Una vez ahí, comienza el posicionamiento de las articulaciones con los bloques “brazo”, “antebrazo” y “mu” donde se controlan el antebrazo, brazo y muñeca respectivamente. A continuación se explica el mecanismo de cada bloque.

### 2.2.1 Brazo

Este subsistema está conformado por cinco bloques (Figura 6), los cuales cumplen respectivamente las siguientes funciones:

**bl\_1:** convierte el ángulo final que deben tener las articulaciones según el algoritmo de Denavit-Hartenverg a valores de señal; dichos niveles, son los que determinan la transición de un bloque a otro y que en el mundo virtual se perciben como un cambio de velocidad de una articulación.

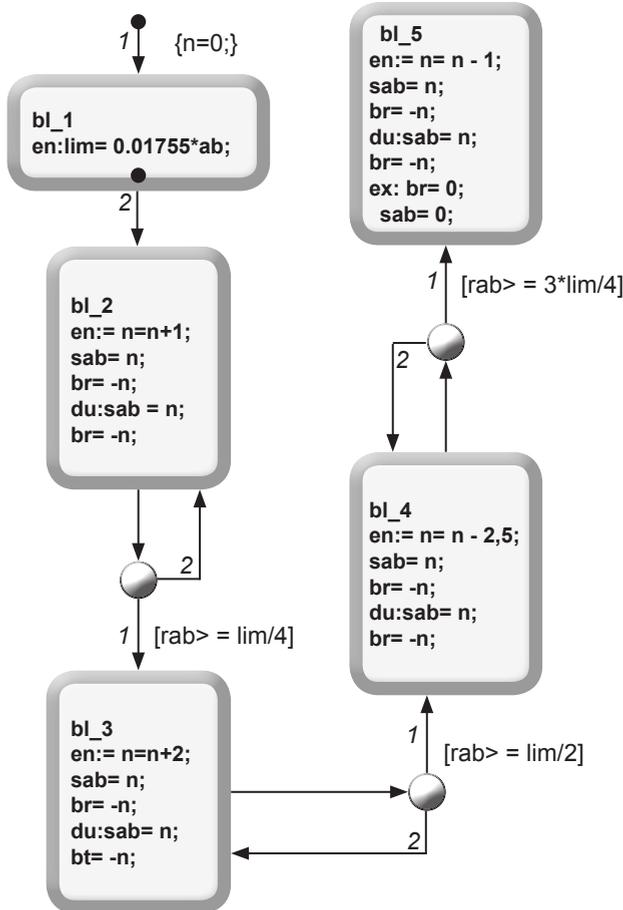
**bl\_2:** inicia el movimiento de la articulación, tratando de mantener la velocidad constante hasta que se ha recorrido un cuarto de su trayectoria completa.

**bl\_3:** incrementa la velocidad de la articulación hasta el doble de la utilizada en el bloque bl\_2 y la mantiene hasta que se haya recorrido la mitad de la trayectoria.

**bl\_4:** disminuye la velocidad de la articulación mientras recorre el camino desde la mitad de la trayectoria hasta  $\frac{3}{4}$  del camino.

**bl\_5:** este bloque, controla la velocidad de la articulación en el último  $\frac{1}{4}$  de su recorrido, disminuye su velocidad lentamente permitiendo que se acerque a su posición final con una inercia y cantidad de movimiento mínimo.

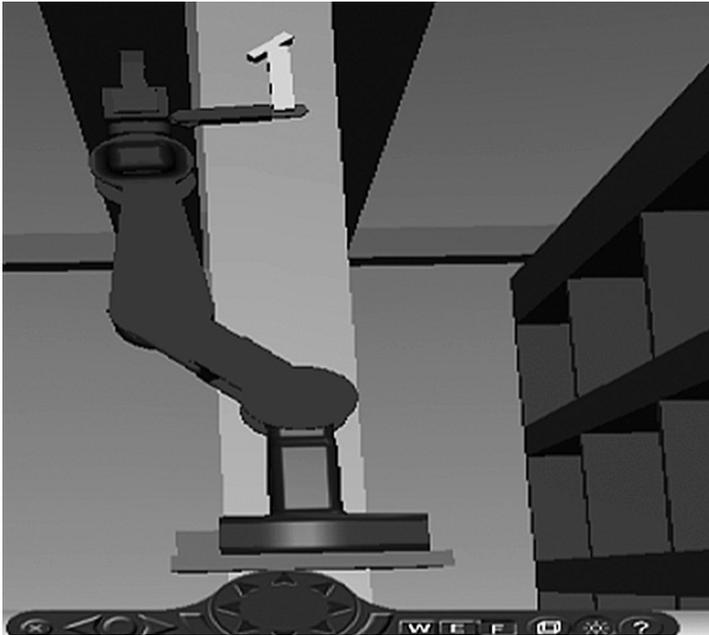
Figura 6. Máquina de estados para el brazo



Fuente. Elaboración propia del autor.

Este subsistema está diseñado para controlar la articulación del brazo, pero por ser esta la base para el antebrazo, logra moverlo a un ángulo diferente del requerido, así que durante todos los bloques de movimiento en este subsistema, se envía la orden de girar en sentido, contrario a la articulación antebrazo; medida que el brazo gira en un sentido el antebrazo gira en el contrario y al final del movimiento el primero queda en el ángulo apropiado mientras que el segundo aún se encuentra con su orientación vertical inicial.

**Figura 7. Posicionamiento del brazo**

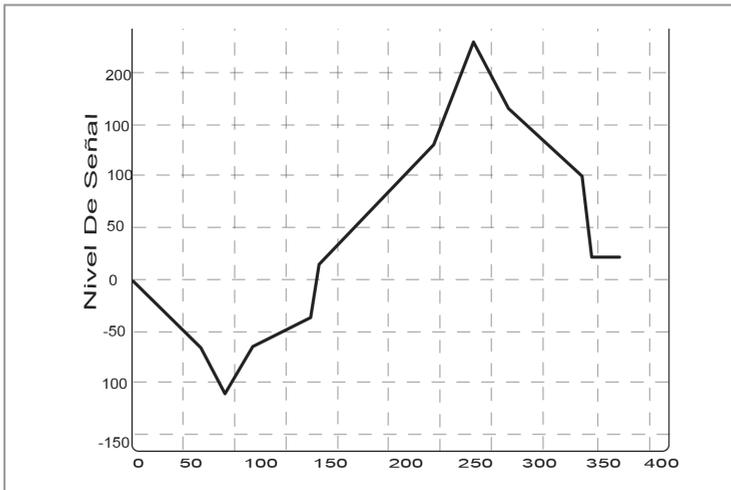


Fuente. Elaboración propia del autor.

### **2.2.2 Antebrazo**

Este subsistema utiliza el mismo mecanismo que el bloque brazo, con la diferencia que en el bloque ini se debe calcular el nivel de señal en que terminó, gracias al retroceso en el subsistema anterior y reducir el recorrido de la señal en función de esto.

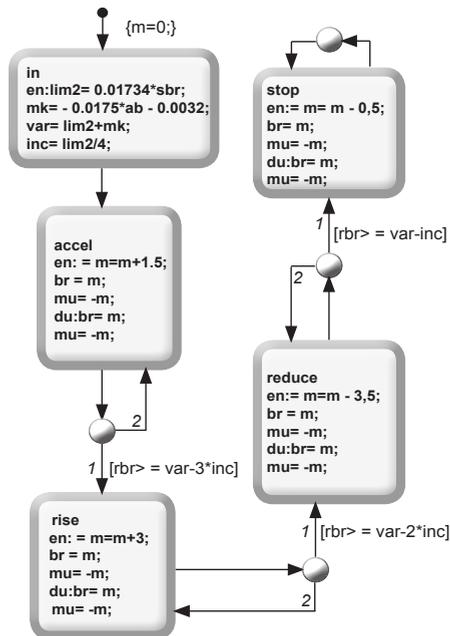
Figura 8. Control de movimiento del antebrazo



Fuente. Elaboración propia del autor.

Para compensar el componente negativo, el sistema del brazo alcanza niveles más altos de señal en el mismo tiempo (Figura 8).

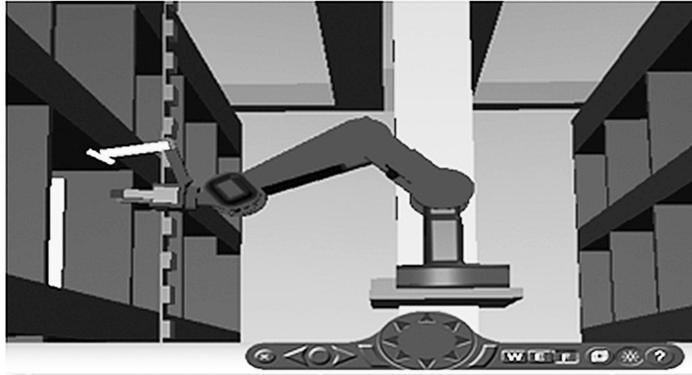
Figura 9. Máquina de estado para el posicionamiento del antebrazo



Fuente. Elaboración propia del autor.

Ahora, el antebrazo complementa el movimiento del brazo para posicionar la pinza.

**Figura 10. Posicionamiento del antebrazo**



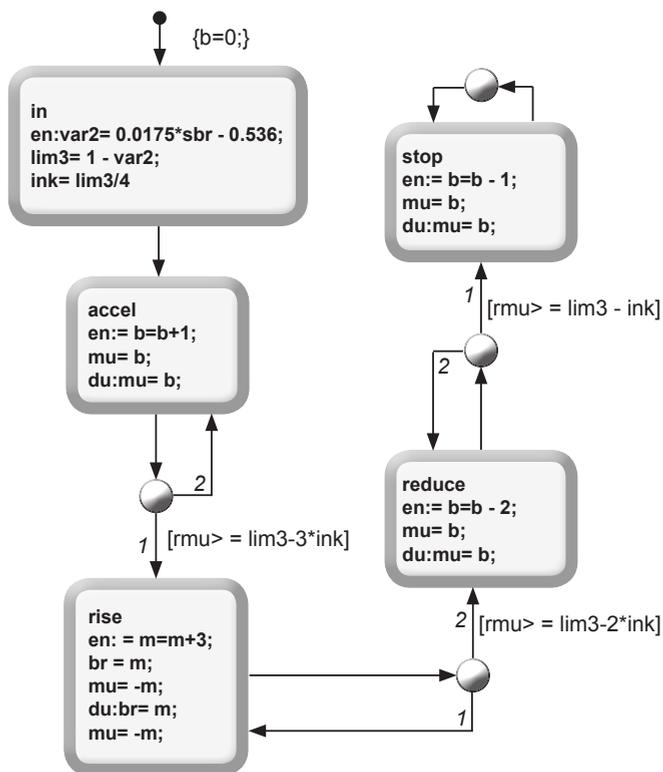
Fuente. Elaboración propia del autor.

### 2.2.3 Muñeca

En este subsistema, se controla el giro de la pinza por lo que el recorrido que debe hacer siempre es de 90 a cero grados; en lo que resta el sistema, es igual al usado para mover el antebrazo.

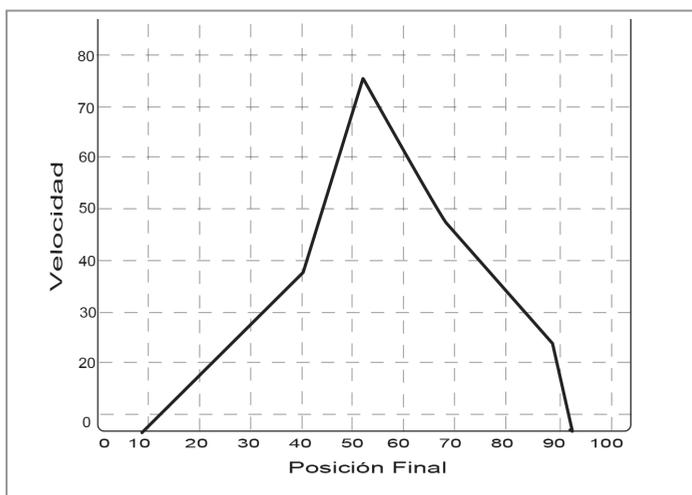
Como ya se ha explicado todas las articulaciones, inicialmente están orientadas verticalmente, por lo que una vez se conoce su posición final comienza un proceso de movimiento en el cual la velocidad se incrementa hasta llegar a la mitad del recorrido y después, tiende a reducirse a medida que se aproxima a la posición final. La forma de triángulo se produce por la utilización de pocas etapas en la máquina de estado, para conseguir un arco más aproximado se necesitarían 'n' bloques de etapas hasta que la figura sea más redondeada.

Figura 11. Máquina de estado de la muñeca



Fuente. Elaboración propia del autor.

Figura 12. Señal de movimiento

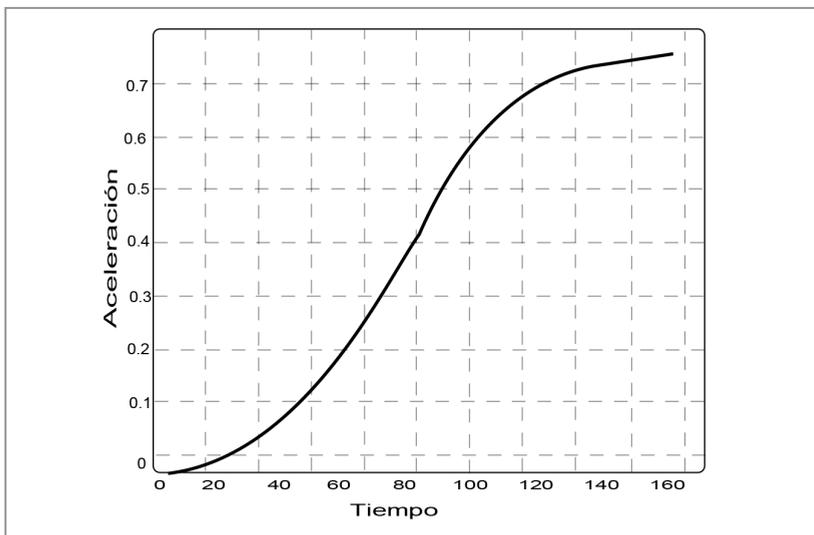


Fuente. Elaboración propia del autor.

## 2.3 Acople *Stateflow* y realidad virtual

Para suavizar la señal de movimiento y que la articulación se mueva de forma fluida, la señal de entrada al bloque VR debe ser continua y de lento crecimiento para evitar que el brazo haga movimientos demasiado bruscos. Esto se consiguió utilizando controladores integrales con constantes de integración sumamente pequeñas, lo cual produce salidas de crecimiento moderado y convergen a la posición final(Figura 13). Una vez alcanzada dicha posición, se debe mantener este valor en la entrada, por lo que se descartó la utilización de controladores proporcionales, dado que estos al finalizar el movimiento pondrían a cero la salida, teniendo como consecuencia un reinicio de la simulación.

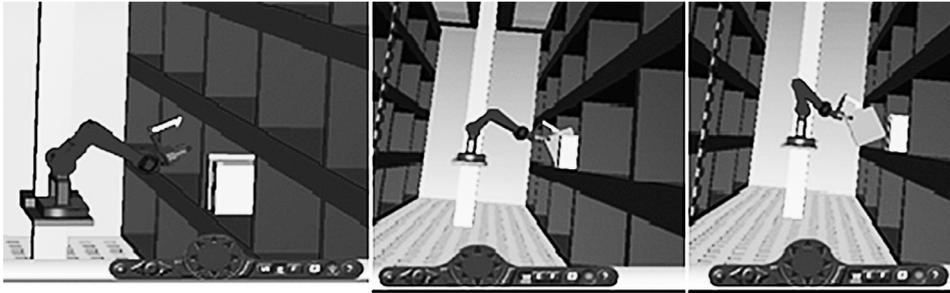
**Figura 13. Señal de control final**



**Fuente. Elaboración propia del autor.**

Con la combinación del *Toolbox de Stateflow* y los bloques de acople se logra representar exitosamente el proceso descrito en el inicio de este artículo (Figura 14), donde se posiciona el brazo y las articulaciones de este para optimizar el trabajo del actuador; posteriormente se acopla el tercer dedo y hace girar el libro para que las pinzas lo aseguren y finalmente, se retira el libro de la estantería y se lleva al final del pasillo para su entrega.

Figura 14. Secuencia de desempeño del sistema *Biblio-bot*



Fuente. Elaboración propia del autor.

### 3. Conclusiones

La validez de estos modelos fue determinada mediante experimentos de simulación en un computador. Para ello fue necesario aprender el uso de la herramienta *Matlab/Simulink* en su modo edición y programación en bloques.

Para la simulación del sistema, se eligieron nuevas herramientas que tienen un gran potencial en el desarrollo de aplicaciones mecatrónicas. Estas fueron el *Toolbox* de realidad virtual y el *Toolbox* de máquinas de estado *Stateflow*, que demostraron ser de gran utilidad en la representación de entornos y situaciones de trabajo con modelos matemáticos, pero demandan un razonable costo computacional y este incrementa a medida que se aproxima el modelo a la realidad.

Mediante el *Toolbox* de realidad virtual, fue posible verificar que el método elegido para la extracción del libro de la estantería es perfectamente viable a la hora de la implementación, debido a que se pudo representar apropiadamente el libro y el actuador con muchas de las reglas del mundo real y su desempeño fue aceptable. Además, esta simulación representa el primer avance en otras etapas como la de ingeniería del producto, la de simulación y validación.

El diseñador de máquinas de estado *Stateflow* nos permitió controlar todo el movimiento del actuador, lo que significó una simplificación considerable en el algoritmo de control, ya que a medida que se desarrollaba la simulación y se anexaban más libros, solo se necesitaba replicar el proceso hecho en uno para la recuperación de otro, ahorrando así el consumo de memoria del computador.

El diseño de un proyecto de gran tamaño como el que se busco representar en esta investigación, es un proceso complejo que implica el mejoramiento continuo de cada una de sus etapas. Sin embargo, es necesario resaltar la importancia de cada pequeña etapa del sistema y la eficiencia con la que se pudo controlar usando *Stateflow*. Lo anterior, conlleva a un control preciso del sistema completo, también hay que tener en cuenta que esta es una primera iteración del diseño y aun esta susceptible a mejoras, pero el análisis por etapas adelantado es muy útil para futuras etapas del proyecto como el diseño específico y la implementación del prototipo.

## 4. Referencias bibliográficas

---

Craig, J. (2006). *Robótica*. Prentice-Hall-. 3ra ed.

Flórez F. y Taborda J. (2012). *Diseño de un sistema automático de préstamo de libros basado en una mano robótica*. Trabajo de tesis no publicado. Santa Marta: Universidad Del Magdalena.

García, A. y Somolinos, J. (2001). *Modelado dinámico y control de un robot flexible de tres grados de libertad*. Universidad de Castilla La Mancha.

Minzhou, L. and Giuseppe, C. (2009). *Analysis and design for changing finger posture in a robotic hand*.

M. Prats. (2009). *Robot physical interaction through the combination of vision, tactile and force feedback*.

Ogata, k. (2003). *Ingeniería de Control Moderna*. Pearson Educación.

Ranferi, D. (2007). *Diseño y construcción de un robot cartesiano de dos grados de libertad para el proceso de spray pyrolysis y síntesis de películas delgadas de cuxo*.

The Mathworks. (2003). *Stateflow and Stateflow Coder*. User's Guide version 5.

The Mathworks. (2004). *Virtual Reality Toolbox User's Guide version 3*.