

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica



“Creación de una interfaz para la manipulación de una garra robótica”

Tesina para obtener el título de
“Licenciado en Ciencias de la Electrónica”

Tesista: **Roberto Youshimatz Bravo**

Asesor: **M. C. Pablo Sánchez Sánchez**

9 de noviembre de 2016

Objetivos

A continuación se presentan los objetivos de la tesina realizando la debida diferenciación entre el objetivo general y los objetivos particulares que deseamos alcanzar.

- **Objetivo general**

- Desarrollar un simulador por computadora basado en una garra robótica.

- **Objetivos particulares**

- Obtener el modelo dinámico del sistema propuesto.
- Analizar y aplicar la estructura de control PD en una aplicación de control de posición, también conocido como *regulación*.
- Desarrollar una interfaz gráfica de usuario (GUI) basado en una garra de tres dedos.
- Realizar diversas pruebas del simulador interpretando los resultados al realizar la tarea asignada (control de posición).

Justificación

- **Justificación académica**

- Aplicar el conocimiento adquirido en la carrera para poder desarrollar un simulador y evaluar una estructura de control. Cumpliendo con las etapas de diseño (modelado, evaluación e implementación).

- **Justificación social**

- La intención en desarrollar el simulador es contar con una herramienta que permita su uso a estudiantes, quienes después de analizar y obtener el modelo, visualicen el movimiento del sistema.

Índice general

Objetivos	III
Justificación	V
1. Introducción	1
1.1. Definición y clasificación del robot	2
1.2. Morfología del robot	2
1.2.1. Configuración de estructuras.	2
1.2.2. Garras de cierre angular y paralelo.	3
1.2.3. Garras de vacío	4
1.2.4. Garras magnéticas	4
1.3. Conclusión	7
2. Conceptos básicos	9
2.1. Matriz de inercias $M(q)$	11
2.2. Matriz de coriolis y fuerza centrípeta $C(q, \dot{q})$	12
2.3. Par gravitacional $g(q)$	13
2.4. Propiedades del modelo dinámico	13
2.5. Moldeo de energía	14
2.6. Índice de desempeño	16
2.7. Método de Runge-Kutta	17
2.8. Estabilidad en el sentido de Lyapunov	17
2.8.1. Punto de equilibrio	18
2.8.2. Estabilidad	18
2.8.3. Estabilidad asintótica	18
2.8.4. Estabilidad asintótica global	18
2.8.5. Estabilidad exponencial global	19
2.8.6. Inestabilidad	19
3. Modelado del sistema	21
3.1. Modelo dinámico del dedo 1	22
3.2. Modelo dinámico del dedo 2	25
3.3. Modelo dinámico del dedo 3	26

4. Estructura de control	27
4.1. Control del robot	27
4.2. Control PD	27
5. Plataforma de simulación	29
5.1. Diseño gráfico del sistema	30
5.2. Simulación	33
5.3. Simulación del modelo dinámico	34
5.4. Simulación del controlador	35
5.5. Simulador propuesto	35
Conclusiones	43
A. Manual de usuario	45
Bibliografía	51

Índice de figuras

1.1. Robot industrial	2
1.2. Pinzas radiales y angulares	3
1.3. Puntos de contacto	3
1.4. Pinzas con ventosas	4
1.5. Garra magnética	5
1.6. Soldadura por puntos	6
1.7. Proceso de pintura	6
3.1. Posición inicial	21
3.2. Posición inicial dedo 2	25
3.3. Posición inicial dedo 3	26
5.1. Plataforma de simulación	29
5.2. SolidWorks [®]	30
5.3. Base del sistema	30
5.4. Dedo vista superior	31
5.5. MATLAB [®]	31
5.6. GUI	32
5.7. Diagrama de flujo	33
5.8. Diagrama	34
5.9. Posición deseada	34
5.10. Posición deseada	35
5.11. Señal del controlador PD	36
5.12. Ventana de nuevo panel	36
5.13. Archivo .fig	37
5.14. Archivo .m	38
5.15. Sistema y Panel de Control	39
5.16. Garra cerrada	39
5.17. Garra abierta	40
5.18. Señal cuando cierra la garra	40
5.19. Señal cuando abre la garra	41
A.1. Panel de control	45

Capítulo 1

Introducción

Tomar un objeto y manipularlo es una actividad que parece natural, y hasta simple, ya que los seres humanos la realizan sin esfuerzo. Pero la realidad es que todo aquel que quiso darle esta capacidad a una máquina ha descubierto que se trata de una tarea difícil de implementar [1]. Debido al gran crecimiento de la tecnología, las aplicaciones de la robótica han ido evolucionando para beneficio de la sociedad, algunos ejemplos son: realizar tareas de riesgo, aumentar la productividad con menos tiempo y bajo costo, y con la exactitud para realizar las tareas programadas [2].

La robótica es un área multi-disciplinaria, cuyo fin es estudiar el diseño, fabricación y aplicación de los robots, así como las piezas que lo conforman. La palabra robot viene del vocablo checo *robot* que se refiere al trabajo realizado de manera forzada. Un robot es un manipulador automático servo-controlado, reprogramable, capaz de ubicar y orientar piezas, siguiendo trayectorias variables programadas, para la ejecución de diversas tareas [1,2].

Los robots para obtener información del medio utilizan transductores y sensores, después de realizar la etapa de acoplamiento de la señal, realizan la conversión y el procesamiento, el robot debe de influenciar su ambiente para realizar la tarea programada, por lo cual utiliza actuadores, ejemplo de ellos los motores, garras, etcétera. En la presente tesis se pretende desarrollar una plataforma gráfica de simulación basada en una garra robótica de tres dedos. Una garra o actuador final es un dispositivo que se une a la muñeca del brazo del robot manipulador con la finalidad de ayudar en la realización de una tarea específica. La razón por la que existen distintos tipos de elementos terminales es, precisamente por las funciones que realizan. Los diversos tipos se pueden dividir en dos categorías: herramientas y pinzas [7]. Las *herramientas* se usan como actuador final en aplicaciones en donde se le exige al robot realizar alguna operación sobre una pieza de trabajo mientras que las pinzas se utilizan para tomar un objeto, normalmente la pieza de trabajo, sujetarlo y manipularlo durante el ciclo de trabajo del robot [3].

En ésta tesis se analizará una garra con tres dedos, se bosquejará usando un software de diseño asistido por computadora que permita hacer una representación tridimensional. Para realizar la simulación de la garra es necesario describir su comportamiento a través del modelo dinámico. El *modelo dinámico* describe el comportamiento del sistema a un estímulo específico, este debe ser resuelto de manera iterativa mediante el uso de un proceso numérico. Obtener del modelo dinámico de un robot es uno de los aspectos más complejos e importantes de la robótica, ya que es imprescindible para: la simulación del movimiento del robot; el diseño y evaluación de la estructura mecánica, así como del control dinámico y el dimensionamiento de los actuadores [4].

1.1. Definición y clasificación del robot

Un robot es un manipulador reprogramable y multi-funcional diseñado para mover materiales, partes, herramientas o dispositivos específicos mediante movimientos programados para realizar diferentes tareas [4]. En su aplicación industrial los robots manipuladores son comúnmente usados en tareas repetitivas y de alta precisión, así como en ciertas actividades que pueden ser de peligro para una persona. Las principales ventajas a las que se hace referencia el uso de los robots manipuladores en la industria son: la reducción de costos, el incremento de la precisión, la calidad y la productividad, y una mayor flexibilidad comparada con la de las máquinas especializadas [3].

1.2. Morfología del robot

El uso de robots en aplicaciones industriales requiere el aumento de las capacidades por medio de dispositivos adicionales tales como *los periféricos* que se une a la muñeca del robot y *los sensores* que le permiten al robot interactuar con su entorno [3].

1.2.1. Configuración de estructuras.

En robótica, el término *efector final* se usa para describir herramientas que están unidas a la muñeca del brazo del robot que le permite al robot realizar una aplicación particular. Este herramental especial debe diseñarse específicamente para la aplicación. Las garras se clasifican según el sistema de ejecución empleado [3].

Las garras estándar tienen dos tipos de movimientos de cierre, paralelo o angular, y el movimiento puede ser operado mediante hidráulica, neumática, eléctrica o por resortes.



Figura 1.1: Robot industrial

1.2.2. Garras de cierre angular y paralelo.

En parada, la garra puede estar abierta o cerrada, el controlador enviará la señal programada para su funcionamiento. El estándar es abierto y cerrado por aire a presión, también se pueden accionar por un resorte. Cada elección tiene sus ventajas y desventajas. La más sencilla es la de cerrado por energía, y abierto con resorte, ya que solo se envía una señal de cierre.

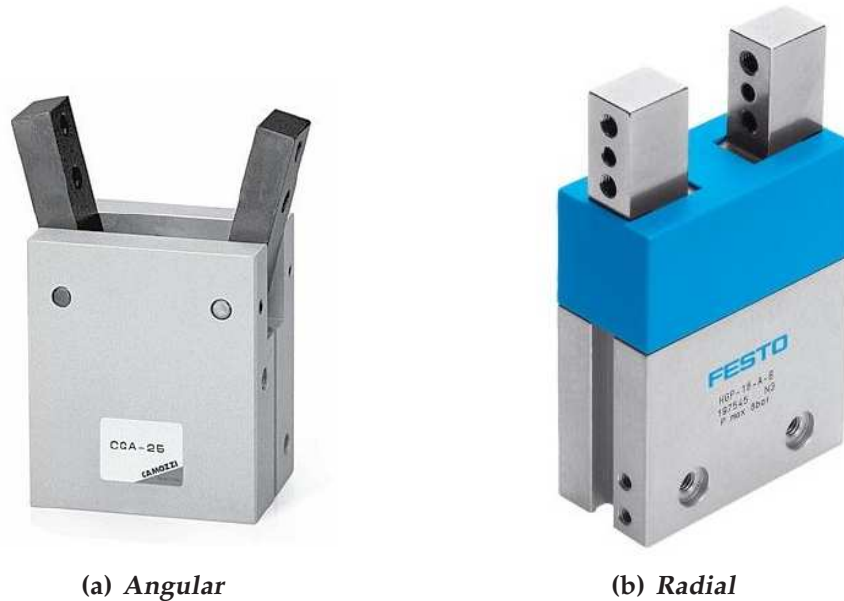


Figura 1.2: Pinzas radiales y angulares

También podemos elegir entre el número de puntos de agarre, variará según la forma del elemento a manipular. Cuantos más puntos más exacto será la posición de entrega de la pieza, sobre todo en los casos de formas irregulares.



Figura 1.3: Puntos de contacto

Las garras activadas eclécticamente tienen un servomotor para el cierre y la apertura. Este servomotor tiene la ventaja de poder modular la presión ejercida en la recogida de la pieza por lo que nos proporciona un abanico de posibilidades ya que podemos recoger tanto piezas de morfología dura así como piezas muy endebles y frágiles sin que sufran daño alguno.

1.2.3. Garras de vacío

La energía utilizada en este sistema de agarre es producido por una bomba de vacío, usando para la recogida unas ventosas de succión acopladas a la garra. La potencia de agarre depende del vacío producido y del área de aplicación.

Según el área disponible para la succión necesitaremos de una ventosa, si el espacio y el peso del elemento son reducidos, o de varias en caso contrario como por ejemplo para levantar planchas de acero. Un sensor de proximidad envía una señal al controlador indicando que ha llegado al final de su recorrido, y así detener el robot, acto seguido envía otra señal para proceder a la succión de la pieza.



Figura 1.4: Pinzas con ventosas

En los casos donde es necesario recoger muchos elementos a la vez podemos utilizar una batería de ventosas. Esto puede suceder cuando tenemos mucha producción y requerimos gran velocidad en el paletizado. Esta batería de ventosas nos permite incluso realizar capas completas de mosaico de una sola vez.

1.2.4. Garras magnéticas

Cuando los elementos contienen metales susceptibles de ser atraídos por imanes, pueden ser recogidos por un electroimán aplicado en el terminal.

El *accionamiento neumático* es el más utilizado ya que ofrece mayores ventajas en simplicidad, precio y fiabilidad, aunque presenta dificultades de control de posiciones intermedias. En ocasiones se usan *accionamientos de tipo eléctrico*.

En la pinza se colocan sensores para detectar el estado de la misma (abierto o cerrado). Se pueden incorporar a la pinza otro tipo de sensores para controlar el estado de la pieza, sistemas de visión que incorporen datos geométricos de los objetos, detectores de proximidad, sensores fuerza par, etcétera.



Figura 1.5: Garra magnética

Existen ciertos elementos comerciales que sirven de base para la pinza, siendo posible a partir de ellos diseñar actuadores válidos para cada aplicación concreta. Sin embargo, en otras ocasiones el actuador debe ser desarrollado en su totalidad íntegramente, constituyendo un porcentaje importante dentro del coste total de la aplicación.

En la elección o diseño de una pinza se deben tener en cuenta diversos factores. El tipo de objeto, el tipo de manipulación, el peso, la forma, el tamaño del objeto y la fuerza que es necesaria para ejercer la sujeción. Entre los parámetros de la pinza cabe destacar su peso (que afecta a las inercias del robot), el equipo de accionamiento y la capacidad de control.

Una herramienta se utilizaría como efector final en aplicaciones en donde se exija al robot realizar alguna operación en la pieza de trabajo. Estas aplicaciones incluyen la soldadura por puntos la cual es un proceso en el que dos piezas de metal se soldán en puntos localizados al hacer pasar una gran corriente eléctrica a través de las piezas donde se efectúa la soldadura.



Figura 1.6: Soldadura por puntos

Uno de los procesos importantes dentro de la industria es el recubrimiento de piezas con un cierto material el cual puede ser pintura, esmalte o incluso partículas metálicas con el objeto de proteger la pieza o bien con fines decorativos. La mayoría de estos procesos se realiza mediante el uso de herramientas neumáticas tales como aerógrafos, cuya función es pulverizar la pintura para realizar el recubrimiento.



Figura 1.7: Proceso de pintura

1.3. Conclusión

En base a esto podemos darnos cuenta de la variedad de tipos de garras que existen, así como sus distintos tipos de aplicaciones a las que pueden estar sujetas, todo en base a una programación que les indica el trabajo que deben hacer.

Capítulo 2

Conceptos básicos

Los robots manipuladores son sistemas mecánicos articulados con eslabones conectados entre sí a través de uniones conocidas como articulaciones. Las articulaciones son de tres tipos: rotacionales, angulares y traslacionales. Para un robot de n grados de libertad, el vector de posición articular q tendrá n elementos:

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad (2.1)$$

Por otro lado, también resulta de gran interés, sobre todo desde un punto de vista práctico, la determinación de la posición y orientación del dispositivo terminal del robot, puesto que este dispositivo terminal realiza la tarea encomendada al robot, tal como abrir o cerrar una garra, aplicar una soldadura, etcétera, dicha posición y orientación se expresa en términos del sistema de referencia cartesiano $\{x, y, z\}$ ubicado en la base del robot, así como eventualmente en términos de los llamados ángulos de Euler.

Dichas coordenadas (y ángulos) son agrupados en el vector x de posiciones cartesianas, donde $m \leq n$.

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (2.2)$$

Cuando el extremo final del robot puede tomar cualquier posición y orientación en el espacio euclidiano de dimensión 3, se tendrá $m = 6$. Por otra parte, si el movimiento del robot se realiza en un plano (dimensión 2) y sólo interesa la posición de su extremo final, entonces $m = 2$. Si además se está interesado en su orientación en el plano, entonces $m = 3$.

El modelo dinámico de un robot manipulador es aquel que describe el comportamiento de un sistema a un estímulo específico, ya sea este un estímulo interno o externo, cuya representación matemática es:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (2.3)$$

donde: $q, \dot{q}, \ddot{q} \in \mathbb{R}^{n \times 1}$ son el vector de posición, velocidad y aceleración articular del robot respectivamente, $M(q) \in \mathbb{R}^{n \times n}$ es la matriz de inercias, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ es la matriz de coriolis y fuerza centrípeta, $g(q) \in \mathbb{R}^{n \times 1}$ es el par gravitacional y $\tau \in \mathbb{R}^{n \times 1}$ es el vector de pares aplicados.

Para la obtención del modelo dinámico se usa la ecuación de movimiento de Euler-Lagrange. El método de Euler-Lagrange requiere de: **1)** la obtención de la cinemática directa, **2)** el cálculo de la velocidad, y **3)** la rapidez, así como **4)** del cálculo de las energías cinética y potencial, lo que permite obtener **5)** el Lagrangiano, el cual **6)** se evalúa en en la ecuación de movimiento de Euler-Lagrange.

A continuación se describen los pasos a seguir para la obtención del modelo dinámico:

Paso 1. Obtención de la cinemática directa. Describe la relación entre la posición articular q y la posición espacial (x, y, z) del efector final del robot, la cual se describe como:

$$x = f(q) \quad (2.4)$$

Paso 2. Obtención de la velocidad. Es la distancia que recorre el robot en función del tiempo, partiendo de su posición de origen, y está determinada mediante la ecuación:

$$v = \frac{dq}{dt} \quad (2.5)$$

donde v es la velocidad del robot; q es la matriz de posición del robot y t es el tiempo de desplazamiento del robot.

Paso 3. Cálculo de la rapidez. Es la razón de cambio de la velocidad del cuerpo y se calcula aplicando la norma del vector al cuadrado:

$$\|\dot{v}\|^2 = \dot{v}^T \dot{v} \quad (2.6)$$

Paso 4. Cálculo de las energías. Se calcula la energía cinética y la potencial del sistema

- **Energía cinética.** Está definida como el trabajo necesario para acelerar un cuerpo de masa m dada desde el reposo hasta su posición deseada. Una vez conseguida ésta energía durante la aceleración, el cuerpo mantiene su energía cinética salvo que cambie su rapidez o su masa. Para que el cuerpo regrese a su estado de reposo se requiere un trabajo negativo de la misma magnitud de la energía cinética. La energía cinética nunca puede ser

negativa, y es cero sólo si la partícula está en reposo. La suma de la energía de traslación más la energía de rotación tiene una representación matemática dada como:

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2}mv^2 + \frac{1}{2}I\dot{q}^2 \quad (2.7)$$

donde m es la masa de la partícula, v es la velocidad lineal de la partícula, I es el momento de inercia de la partícula y \dot{q} es la velocidad articular.

- **Energía potencial.** Es la energía que una partícula tiene debido a su posición en el espacio, no a su movimiento cuya ecuación es:

$$\mathcal{U}(q) = m g h \quad (2.8)$$

donde m es la masa de la partícula, g es la fuerza gravitatoria, h es la altura.

Paso 5. Obtención del Lagrangiano. El Lagrangiano es una función escalar a partir de la cual se puede obtener la evolución temporal, las leyes de conservación y otras propiedades importantes de un sistema dinámico, el cual obtenemos a partir de la diferencia de las energías cinética y potencial:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q) \quad (2.9)$$

Paso 6. Evaluación del Lagrangiano. La ecuación de movimiento de Euler-Lagrange se utiliza para describir cualquier sistema mecánico por medio de coordenadas generalizadas de posición y se representa mediante la siguiente fórmula:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau \quad (2.10)$$

A continuación se analizan las características más importantes del modelo dinámico del robot manipulador, las cuales se toman en cuenta para el diseño de controladores.

2.1. Matriz de inercias $M(q)$

La *matriz de inercias* $M(q) \in \mathbb{R}^{n \times n}$ es una matriz simétrica, definida positiva. Existe una constante real positiva α tal que:

$$M(q) \geq \alpha I \quad \forall q \in \mathbb{R}^n \quad (2.11)$$

donde $I \in \mathbb{R}^{n \times n}$ denota la matriz identidad. la matriz $M(q)^{-1} \in \mathbb{R}^{n \times n}$ existe y es definida positiva.

Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante $\beta > 0$ tal que:

$$\lambda[M(q)] \leq \beta \quad \forall q \in \mathbb{R}^n \quad (2.12)$$

una manera sencilla de calcular β es la dada por:

$$\beta \geq n[\text{máx } |M_{ij}(q)|] \quad (2.13)$$

donde $M_{ij}(q) \in \mathbb{R}^{n \times n}$ denota al ij -ésimo elemento de la matriz $M(q)$.

Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante $K_M > 0$ tal que:

$$\|M(x)z - M(y)z\| \leq K_M \|x - y\| \|z\| \quad (2.14)$$

para todo vector $x, y, z \in \mathbb{R}^n$. Una manera sencilla de determinar K_M es:

$$K_M \geq n^2 \left[\text{máx} \left| \frac{\partial M_{ij}(q)}{\partial q_k} \right| \right] \quad (2.15)$$

Para el caso de robots provistos únicamente de articulaciones rotacionales, existe una constante $K'_M > 0$ tal que:

$$\|M(x)y\| \leq K'_M > 0 \|y\| \quad (2.16)$$

para todo $x, y \in \mathbb{R}^n$

2.2. Matriz de coriolis y fuerza centrípeta $C(q, \dot{q})$

La primera observación que se puede hacer sobre el término $C(q, \dot{q})$ es que es un término cuadrático con respecto a la velocidad generalizada \dot{q} . La matriz de coriolis y fuerza centrípeta está acotada de la siguiente forma:

$$\|C(q, \dot{q})\| \leq v_b(q) \|\dot{q}\|^2 \quad (2.17)$$

donde $v_b(q) \in \mathbb{R}$ es un escalar constante para una articulación de tipo rotacional, generalmente es una función escalar de q para una articulación prismática; aunado a esto existen numerosas factorizaciones de $C(q, \dot{q})$ una de las cuales está dada por:

$$C(q, \dot{q}) = C(q, \dot{q})\dot{q}. \quad (2.18)$$

Esta factorización es particularmente importante en controladores que aprovechan la pasividad del robot, donde $C(q, \dot{q})$ es seleccionada para satisfacer la siguiente propiedad llamada matriz antisimétrica:

$$\frac{1}{2} \dot{q}^T [\dot{M}(q) + 2C(q, \dot{q})] \dot{q} = 0. \quad (2.19)$$

La matriz de coriolis y fuerza centrípeta se define como:

$$C(q, \dot{q}) = \dot{M}(q) \dot{q} - \frac{1}{2} \frac{\partial}{\partial q} [\dot{q}^T M(q) \dot{q}] \quad (2.20)$$

2.3. Par gravitacional $g(q)$

Para un robot con articulaciones rotacionales el par gravitacional está acotado de la siguiente forma:

$$\|g(q)\| \leq g_b \quad (2.21)$$

donde g_b es un escalar constante. El par gravitacional se obtiene mediante el gradiente de la energía potencial del robot.

$$g(q) = \frac{\partial \mathcal{U}(q)}{\partial q} \quad (2.22)$$

2.4. Propiedades del modelo dinámico

El modelo dinámico, tiene las siguientes propiedades que nos sirven para el diseño de algoritmos de control:

- P1.** Para el modelo del robot rígido, hay una entrada de control independiente para cada coordenada generalizada y no es válido para robots que poseen flexibilidad en las articulaciones.
- P2.** La ecuación dinámica define un mapa pasivo desde el par de entrada τ a la velocidad generalizada \dot{q} . Esta propiedad puede ser observada utilizando la ecuación de movimiento de Hamilton. El hamiltoniano $H(q, \rho)$ está definido por:

$$\mathcal{H}(q, \rho) = \mathcal{K}(q, \dot{q}) + \mathcal{U}(q) \quad (2.23)$$

lo que podemos representar de la siguiente forma:

$$\frac{\partial \mathcal{H}(q, \rho)}{\partial t} = \frac{\partial \mathcal{K}(q, \dot{q}) + \mathcal{U}(q)}{\partial t} \quad (2.24)$$

para lo cual:

$$\int_0^\infty \dot{q}^T(u) \tau(u) dy = \mathcal{H}(t) - \mathcal{H}(0) \leq -\mathcal{H}(0) \quad (2.25)$$

lo que define un mapa pasivo desde el par de entrada τ a la velocidad generalizada \dot{q} .

P3. La ecuación de movimiento es lineal por lo que es conveniente definir el conjunto de parámetros. En otras palabras, podemos escribir la ecuación de la siguiente forma:

$$\Psi(q, \dot{q}, \ddot{q})\theta = \tau \quad (2.26)$$

en donde $\Psi(q, \dot{q}, \ddot{q})$ es la matriz de funciones conocidas, θ es un vector que contiene los parámetros físicos del robot como la masa de las articulaciones, los momentos de inercia u otros factores.

2.5. Moldeo de energía

El *moldeo de energía* es un método para el diseño de controladores, el cual no considera la fricción y las perturbaciones externas.

$$M(q)\dot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (2.27)$$

con un error de posición:

$$\tilde{q} = q_d - q \quad (2.28)$$

donde q_d es la posición deseada y q es la posición real de los eslabones del robot.

ME1. El primer paso consiste en encontrar el modelo en lazo cerrado, para lo cual expresamos las variables de estado del sistema considerando el error de posición como un estado.

$$x_1 = q_d - q \quad (2.29)$$

$$\dot{x}_1 = -\dot{q} = x_2 \quad (2.30)$$

$$\dot{x}_2 = \ddot{q} = M(q)^{-1}[\tau - C(q, \dot{q})\dot{q} - g(q)] \quad (2.31)$$

lo que nos describe un sistema matricial de la forma:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1}[\tau - C(q, \dot{q})\dot{q} - g(q)] \end{bmatrix} \quad (2.32)$$

ME2. El segundo paso es considerar una familia de controladores de la forma:

$$\tau_1 = \nabla \mathcal{U}(K_p, \tilde{q}) - f_v(K_v, \dot{q}) + g(q) \quad (2.33)$$

donde $\mathcal{U}(K_p, \tilde{q})$ es la energía potencial artificial que depende de K_p y esto definida por:

$$\mathcal{U}(K_p, \tilde{q}) = \frac{f(\tilde{q})^T K_p f(\tilde{q})}{2} \quad (2.34)$$

El término $f_v(K_v, \dot{q})$ es una función derivativa que depende de k_v (función de amortiguamiento) cuya expresión matemática es:

$$f_v(K_v, \dot{q}) = q^T f_v(K_v, \dot{q}) > 0 \quad (2.35)$$

Los elementos de la ecuación anterior tienen las siguientes propiedades:

- $\mathcal{U}(K_p, \tilde{q}) > 0$ es diferente a $\tilde{q}^T q^T f_v(K_v, \dot{q}) > 0$ tiene propiedad disipativa.
- $\nabla \mathcal{U}(K_p, \tilde{q})|_{\tilde{q}=0} = 0$

ME3. El tercer paso es considerar una función de Lyapunov de la forma:

$$V(q, \tilde{q}) = \frac{\dot{q}^T M(q) \dot{q}}{2} + \mathcal{U}(K_p, \tilde{q}) \quad (2.36)$$

La función $\mathcal{U}(K_p, \tilde{q})$ debe cumplir las condiciones de la función de Lyapunov:

$$V(0, 0) = 0 \quad \dot{q} \wedge \tilde{q} = 0 \quad (2.37)$$

$$V(\dot{q}, \tilde{q}) > 0 \quad \forall \dot{q}, \tilde{q} \neq 0 \quad (2.38)$$

ME4. El cuarto paso es derivar la función de Lyapunov:

$$\dot{V}(\dot{q}, \tilde{q}) = \dot{q}^T M(\dot{q}) \ddot{q} + \frac{\dot{q}^T \dot{M}(\dot{q}) \dot{q}}{2} - \frac{\partial \mathcal{U}(K_p, \tilde{q})^T}{\partial \tilde{q}} \dot{q} \quad (2.39)$$

ME5. El quinto paso es obtener un punto de equilibrio estable, cumpliendo con la condición:

$$V(\dot{q}, \tilde{q}) \leq 0 \quad (2.40)$$

Hasta este punto solamente se ha comprobado que el sistema es estable.

ME6. Finalmente comprobamos estabilidad asintótica en forma global con el empleo del teorema de LaSalle.

$$V(\dot{q}, \tilde{q}) < 0 \quad (2.41)$$

existe el conjunto Ω tal que:

$$\Omega = [q \in \mathbb{R}^n : \dot{v}(q) = 0] \quad (2.42)$$

Si $q(0) = 0$ es la única condición inicial en Ω para el cual $q(t) \in \Omega$ para $t \geq 0$, entonces el origen $x = 0 \in \mathbb{R}^n$ es un equilibrio asintóticamente estable en forma global.

2.6. Índice de desempeño

El índice de desempeño se utiliza para medir la norma \mathcal{L}^2 del error de posición. Un valor pequeño de \mathcal{L}^2 representa un error pequeño de posición y por lo tanto indica un mejor desempeño. Una función vectorial $\mathbb{R}^n \rightarrow \mathbb{R}^n \in \mathcal{L}^2$, si al evaluar

$$\mathcal{L}^2 = \sqrt{\int_0^{\infty} \|f(t)\|^2 dt} < \infty \quad (2.43)$$

donde $\|f(t)\|$ es la norma euclidiana de la función sobre el intervalo; es un número escalar. Esta propiedad en funciones vectoriales es una medida para determinar la convergencia conforme el tiempo evolucione.

Debido a que el tiempo de simulación es finito debemos aplicar el concepto de valor efectivo para calcular la desviación de la función entre los intervalos de simulación, por lo cual definimos la norma \mathcal{L}^2 como:

$$\mathcal{L}^2 = \frac{1}{T} \sqrt{\int_0^T \|\tilde{q}\|^2 dt} < \infty \quad (2.44)$$

cuya representación discreta de la norma es:

$$\int \|\tilde{q}\|^2 dt \rightarrow I_k = I_{k-1} + h \|\tilde{q}\|^2 \mathcal{L}^2 = \sqrt{\frac{1}{T} I_k} \quad (2.45)$$

donde h es el período de muestreo y T es el intervalo de evaluación.

2.7. Método de Runge-Kutta

Uno de los métodos más utilizados para resolver numéricamente problemas de ecuaciones diferenciales ordinarias con condiciones iniciales es el método de Runge-Kutta de cuarto orden, el cual proporciona un pequeño margen de error con respecto a la solución real del problema y es fácilmente programable en un software para realizar las iteraciones necesarias. El método de Runge-Kutta se utiliza para resolver ecuaciones diferenciales de la forma:

$$\frac{dy(t)}{dt} = f(t, y) \quad y(t_0) = y_0 \quad (2.46)$$

Y es útil para casos en los que la solución no puede hallarse por los métodos convencionales (como separación de variables). En el método de Runge-Kutta cuatro elige un tamaño de paso h y un número máximo de iteraciones n tal que:

$$y_0 = y(t_0) \quad (2.47)$$

$$k_1 = hf(t_i, y_i) \quad (2.48)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \quad (2.49)$$

$$k_3 = hf\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \quad (2.50)$$

$$k_4 = hf(t_i + h, y_i + k_3) \quad (2.51)$$

cuya iteración es la siguiente:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \quad (2.52)$$

2.8. Estabilidad en el sentido de Lyapunov

La teoría de estabilidad de Lyapunov tiene como principal objetivo estudiar el comportamiento de sistemas dinámicos descritos por ecuaciones diferenciales de la forma:

$$\dot{x}(t) = f(t, x(t)), \quad x(0) \in \mathbb{R}^n \quad \forall t \geq 0 \quad (2.53)$$

donde el vector $x(t) \in \mathbb{R}^n$ se refiere al estado del sistema dinámico representado por (1.7) y $x(0) \in \mathbb{R}^n$ se denomina la condición inicial o estado inicial. La función $f : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ es una función continua en t y $x(t)$, y se supone que:

- La ecuación (1.7) tiene una solución única en el intervalo $[0, \infty)$, correspondiente a cada condición inicial de $x(0)$.
- Si $x(t)$ es la solución de (1.7) correspondiente a la condición inicial $x(0)$, entonces $x(t)$ depende de una manera continua del estado inicial $x(0)$.

2.8.1. Punto de equilibrio

Un vector constante $x_e \in \mathbb{R}^n$ es un punto de equilibrio o estado de equilibrio si:

$$f(t, x_e) = 0 : \quad \forall t \geq 0 \quad (2.54)$$

Como consecuencia se tiene que si la condición inicial $x(0) \in \mathbb{R}^n$ es justamente un punto de equilibrio ($x(0) = x_e \in \mathbb{R}^n$), entonces se satisface: $x(t) = x_e \quad t \geq 0$

2.8.2. Estabilidad

El origen $x = 0 \in \mathbb{R}^n$ es un punto de equilibrio estable (en el sentido de Lyapunov) si para cada número $\varepsilon \geq 0$, se puede encontrar un número $\delta \geq 0$, tal que:

$$\|x(0)\| \delta \implies \|x(t)\| < \varepsilon \quad \forall t \geq 0 \quad (2.55)$$

Debe tenerse claro que la definición de estabilidad requiere la existencia de un $\delta > 0$ para cada $\varepsilon > 0$ y no para algún $\varepsilon > 0$.

Lo anteriormente dicho significaría que el origen es un punto de equilibrio estable si para toda condición inicial acotada, las soluciones también están acotadas. La estabilidad del origen debe ser interpretada como dada una pequeña discrepancia de la condición inicial $x(0)$ con respecto al equilibrio (origen), la solución $x(t)$ correspondiente permanecerá acotada.

2.8.3. Estabilidad asintótica

El origen $x = 0 \in \mathbb{R}^n$ es un punto de equilibrio asintóticamente estable si:

- 1 El origen es estable.
- 2 El origen es atractivo, es decir, existe un número $\delta' > 0$ tal que:

$$\|x(0)\| \delta' \implies \|x(t)\| \rightarrow 0 \quad \text{cuando } t \rightarrow \infty \quad (2.56)$$

2.8.4. Estabilidad asintótica global

El origen $x = 0 \in \mathbb{R}^n$ es un punto de equilibrio asintóticamente estable en forma global de la ecuación (1.7) si:

- 1 El origen es estable.
- 2 El origen es atractivo globalmente, es decir,

$$\|x(t)\| \rightarrow 0 \quad \text{cuando } t \rightarrow \infty, \quad x(0) \in \mathbb{R}^n \quad (2.57)$$

De la definición anterior se deduce que un punto de equilibrio asintóticamente estable globalmente implica que dicho equilibrio es también asintóticamente estable, pero lo contrario es falso.

2.8.5. Estabilidad exponencial global

El origen $x = 0 \in \mathbb{R}^n$ es un punto de equilibrio exponencialmente estable en forma global de (1.7) si existen constantes positivas α y β tales que:

$$\|x(t)\| < \alpha \|x(0)\| e^{-\beta t}, \quad \forall t \geq 0, \quad \forall x(0) \in \mathbb{R}^n \quad (2.58)$$

De acuerdo con lo anterior un punto de equilibrio exponencialmente estable en forma global es también un punto de equilibrio asintóticamente estable en forma global. Sin embargo, lo contrario no es necesariamente verdadero.

2.8.6. Inestabilidad

El origen $x = 0 \in \mathbb{R}^n$ es un punto de equilibrio inestable de la ecuación (1.7) si éste no es estable. El concepto de equilibrio (origen) inestable es equivalente a enunciar que existe al menos un $\varepsilon > 0$ para el cual no es posible encontrar un $\delta > 0$ tal que:

$$\|x(t)\| < \delta \implies \|x(t)\| < \varepsilon \quad \forall t \geq 0, \quad (2.59)$$

en otras palabras, si existe al menos un $\varepsilon > 0$, el cual se desea que acote la norma de la solución $\|x(t)\| < \varepsilon \quad \forall t \geq 0$, entonces el origen es inestable. Es importante señalar que en ningún momento se afirma que la solución $x(t)$ deba crecer indefinidamente.

Capítulo 3

Modelado del sistema

Se usa el método científico en su topología conocido como *método sistémico*, el cual consiste en modelar el objeto mediante la determinación de sus componentes, así como su relación entre ellos en función de las etapas de diseño.

A continuación se muestra el modelo matemático del sistema propuesto, partiendo de la posición original (posición de casa), con el análisis correspondiente como se muestra en la figura 3.1.

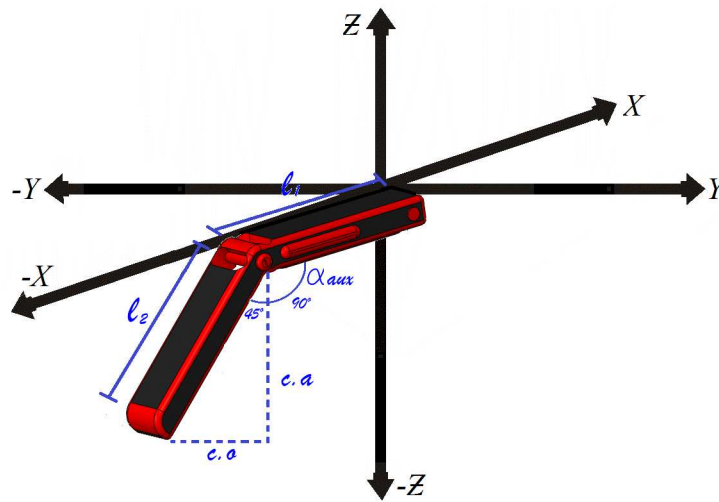


Figura 3.1: Posición inicial

Para sacar la referencia de nuestro sistema, se consideró que el dedo está formado por dos longitudes, donde l_1 va desde el punto de origen del sistema de coordenadas hasta el punto donde el dedo se dobla formando el ángulo α , sobre el eje x negativo, y l_2 que va del punto donde se dobla el dedo formando el ángulo α hasta la punta del mismo sobre el eje x negativo.

Para comenzar haciendo el álgebra correspondiente tomamos en cuenta que nuestro ángulo α lo dividimos en dos ángulos, uno de 45° y otro de 90° . Para lo siguiente tomamos como referencia el ángulo de 45° , puesto que deseamos calcular la posición en coordenadas cartesianas del extremo final de nuestro dedo, mientras se mantiene en su posición inicial (posición de casa).

3.1. Modelo dinámico del dedo 1

A continuación se presentan las ecuaciones algebraicas correspondientes para determinar la matriz de posición de casa de nuestro sistema con los valores correspondientes:

$$\alpha_{aux} = 45^\circ \quad (3.1)$$

$$c.o = h \operatorname{sen}(\alpha_{aux}) \quad (3.2)$$

$$c.a = h \operatorname{cos}(\alpha_{aux}) \quad (3.3)$$

Como el dedo rota al rededor del eje y del sistema de coordenadas, tenemos que y_c es igual a cero.

Paso 1. Definición de posición de casa. Se define la posición de casa del sistema

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} -\left(\ell_1 + \frac{\ell_2}{\sqrt{2}}\right) \\ 0 \\ -\left(\frac{\ell_2}{\sqrt{2}}\right) \end{bmatrix} \quad (3.4)$$

Paso 2. Definición de la cinemática directa. En base a la matriz de rotación $\operatorname{Rot}(y, \beta)$ sobre el eje y se usa con la *posición de casa* para obtener la cinemática directa:

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \operatorname{Rot}(y, \beta) \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} -\left(\ell_1 + \frac{\ell_2}{\sqrt{2}}\right) \operatorname{cos}(q_1) - \left(\frac{\ell_2}{\sqrt{2}}\right) \operatorname{sen}(q_1) \\ 0 \\ \left(\ell_1 + \frac{\ell_2}{\sqrt{2}}\right) \operatorname{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}}\right) \operatorname{cos}(q_1) \end{bmatrix} \quad (3.5)$$

Paso 3. Cálculo de la velocidad. Derivando la cinemática directa se obtiene la velocidad del sistema:

$$v = \frac{d}{dt} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \left(\ell_1 + \frac{\ell_2}{\sqrt{2}}\right) \operatorname{sen}(q_1) \dot{q}_1 - \left(\frac{\ell_2}{\sqrt{2}}\right) \operatorname{cos}(q_1) \dot{q}_1 \\ 0 \\ \left(\ell_1 + \frac{\ell_2}{\sqrt{2}}\right) \operatorname{cos}(q_1) \dot{q}_1 + \left(\frac{\ell_2}{\sqrt{2}}\right) \operatorname{sen}(q_1) \dot{q}_1 \end{bmatrix} \quad (3.6)$$

Paso 4. Obtención de la rapidez. Evaluando la norma del vector al cuadrado $v^2 = \|v\|^2 = v^T v$ se obtiene la rapidez:

$$\begin{aligned}
v^T v &= \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) \dot{q}_1 - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \dot{q}_1 \right]^2 + \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \dot{q}_1 + \left(\frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) \dot{q}_1 \right]^2 \\
&= \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 \text{sen}^2(q_1) \dot{q}_1^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \text{cos}^2(q_1) \dot{q}_1^2 - 2 \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) \dot{q}_1 \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \dot{q}_1 \\
&\quad \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 \text{cos}^2(q_1) \dot{q}_1^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \text{sen}^2(q_1) \dot{q}_1^2 + 2 \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \dot{q}_1 \left(\frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) \dot{q}_1 \\
&= \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 \dot{q}_1^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \dot{q}_1^2
\end{aligned} \tag{3.7}$$

Paso 5. Obtención de las energías. Para obtener la energía cinética se usa la rapidez y la energía potencial usa la altura relativa del movimiento del objeto.

- **Energía cinética.** Para calcular la energía cinética se usa su definición:

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2} m v^2 + \frac{1}{2} I \dot{q}^2 \tag{3.8}$$

Después de manipular (3.8) se tiene que la energía cinética está definida como:

$$\mathcal{K}_1(q, \dot{q}) = \frac{1}{2} \left[m_1 \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + m_1 \left(\frac{\ell_2}{\sqrt{2}} \right)^2 + I_1 \right] \dot{q}_1^2 \tag{3.9}$$

- **Energía potencial.** La energía potencial está definida como:

$$\mathcal{U}(q) = mgh \tag{3.10}$$

donde m representa la masa del dedo, g es la fuerza de gravedad que actúa sobre el dedo, y h es la altura que se mide desde el punto de origen hasta el extremo del dedo y está definida como:

$$h_1 = \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \tag{3.11}$$

Finalmente se tiene que la energía potencial es:

$$\mathcal{U}_1(q) = m_1 g \left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - m_1 g \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \tag{3.12}$$

Paso 6. Obtención del Lagrangiano. Con la energía cinética y la energía potencial se calcular el Lagrangiano, el cual es la diferencia de la energía cinética y la potencial:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q) \quad (3.13)$$

Así se tiene:

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2}m_1 \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \right] \dot{q}_1^2 - m_1 g \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \right] \quad (3.14)$$

Paso 7. Evaluación del Lagrangiano. Se evalúa el Lagrangiano en la ecuación de movimiento de Euler-Lagrange, la cual está definida como:

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau \quad (3.15)$$

- Primero se evalúa $\frac{\partial \mathcal{L}(q, \dot{q})}{\partial q}$

$$\begin{aligned} \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} &= \frac{1}{2}m_1 \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \right] \dot{q}_1^2 - m_1 g \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \right] \\ &= m_1 \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \right] \dot{q}_1 \end{aligned} \quad (3.16)$$

- Se deriva (3.16) para obtener $\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right]$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] = m_1 \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \right] \ddot{q}_1 \quad (3.17)$$

- Finalmente se evalúa $\frac{\partial \mathcal{L}(q, \dot{q})}{\partial q}$:

$$\begin{aligned} \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} &= \frac{1}{2}m_1 \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \right] \dot{q}_1^2 - m_1 g \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \right] \\ &= -m_1 g \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \right] \end{aligned} \quad (3.18)$$

Paso 8. Obtención del modelo dinámico. Se organizan las ecuaciones obtenidas y se obtiene τ :

$$\tau_1 = m_1 \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right)^2 + \left(\frac{\ell_2}{\sqrt{2}} \right)^2 \right] \ddot{q}_1 + m_1 g \left[\left(\ell_1 + \frac{\ell_2}{\sqrt{2}} \right) \text{sen}(q_1) - \left(\frac{\ell_2}{\sqrt{2}} \right) \text{cos}(q_1) \right] \quad (3.19)$$

3.2. Modelo dinámico del dedo 2

Para el segundo dedo la configuración es la misma que la del primero, tal y como se observa en la Figura 3.1, con la única diferencia en la distancia ℓ_{r1} referida al origen.

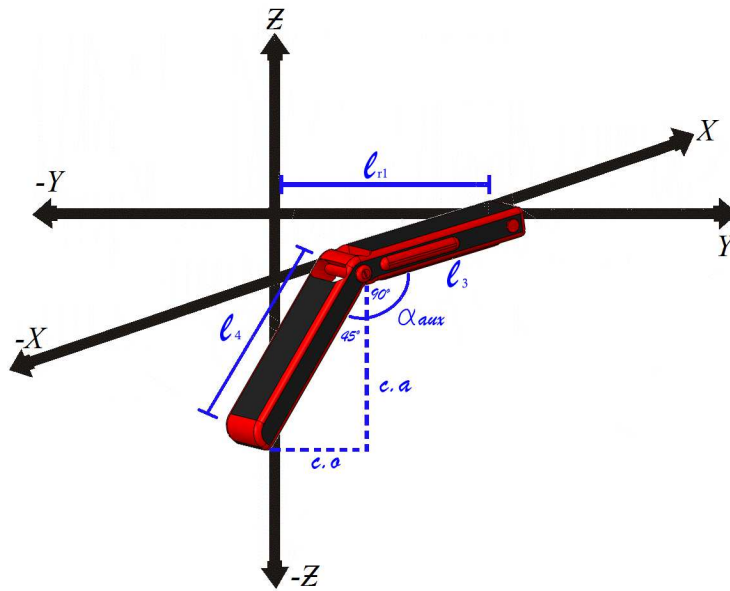


Figura 3.2: Posición inicial dedo 2

Realizando el mismo procedimiento se obtiene:

$$\tau_2 = m_2 \left[\left(\ell_3 + \frac{\ell_4}{\sqrt{2}} \right)^2 + \left(\frac{\ell_4}{\sqrt{2}} \right)^2 + \ell_{r1} \right] \ddot{q}_2 + m_2 g \left[\left(\ell_3 + \frac{\ell_4}{\sqrt{2}} \right) \text{sen}(q_2) - \left(\frac{\ell_4}{\sqrt{2}} \right) \text{cos}(q_2) \right] \quad (3.20)$$

3.3. Modelo dinámico del dedo 3

Lo mismo pasa para el tercer elemento. Para el análisis del tercer dedo se usa la Figura 3.3 con respecto al eje z.

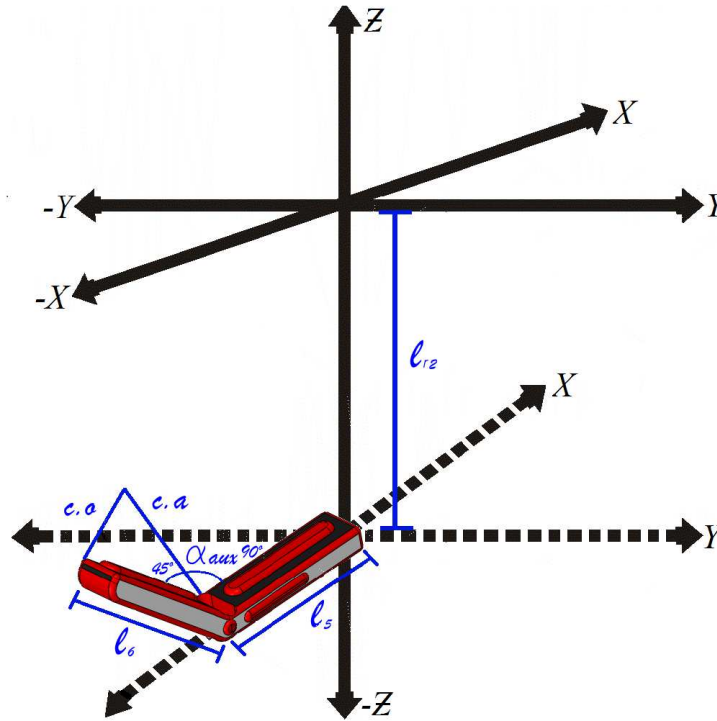


Figura 3.3: Posición inicial dedo 3

El modelo del tercer dedo es el siguiente:

$$\tau_3 = m_3 \left[\left(\ell_5 + \frac{\ell_6}{\sqrt{2}} \right)^2 + \left(\frac{\ell_6}{\sqrt{2}} \right)^2 + \ell_{r2} \right] \ddot{q}_3 + m_3 g \left[\left(\ell_5 + \frac{\ell_6}{\sqrt{2}} \right) \text{sen}(q_3) - \left(\frac{\ell_6}{\sqrt{2}} \right) \text{cos}(q_3) \right] \quad (3.21)$$

Capítulo 4

Estructura de control

Los sistemas físicos no lineales de gran complejidad se caracterizan por que las técnicas de control convencionales son inoperables cuando se requiere que ellos estén en un punto de operación o posición de equilibrio. Para garantizar que un sistema realice la tarea para el cual fue diseñado es necesario conocer las expresiones matemáticas que los describen y la estructura de control que proporciona el torque o par aplicado, en este capítulo se detalla la metodología usada para describir esquemas de control.

4.1. Control del robot

El calculo del vector τ involucra generalmente a una función vectorial no lineal de q, \dot{q}, \ddot{q} , esta función se denomina *ley de control* o simplemente *controlador*. El tipo de control que se desea usar en el simulador se conoce como control de posición; el cual consiste en llevar el efector final del robot desde una posición inicial q_0 a una posición deseada q_d , es decir, encontrar un par aplicado τ tal que la posición real tienda a la posición deseada cuando el tiempo evoluciona hacia el infinito.

4.2. Control PD

El control PD está definido como:

$$\tau = f(K_p, \tilde{q}) - f(K_v, \dot{q}) + g(q) \quad (4.1)$$

donde $f(K_p, \tilde{q})$ es una función que depende de $K_p \in \mathbb{R}^{n \times n}$ la cual es una matriz de ganancias, la cual es diagonal y se conoce como *ganancia proporcional* y $\tilde{q} \in \mathbb{R}^{n \times 1}$ que es el error de posición; $f(K_v, \dot{q})$ es una función que depende de $K_v \in \mathbb{R}^{n \times n}$ que es la matriz de ganancia derivativa, esta matriz es diagonal, y $\dot{q} \in \mathbb{R}^{n \times 1}$ que es la velocidad articular.

La representación más simple del control PD es:

$$\tau = f(K_p, \tilde{q}) - f(K_v, \dot{q}) + g(q). \quad (4.2)$$

Considerando el modelo dinámico de un robot de n grados de libertad, ecuación (2.3). Por comodidad se escribe nuevamente el modelo:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (4.3)$$

Aplicando el control PD, (4.2) se tiene:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(\tau, \dot{q}) = K_p\tilde{q} - K_v\dot{q} + g(q) + f(\tau, \dot{q}) \quad (4.4)$$

Eliminando términos tenemos:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} = K_p\tilde{q} - K_v\dot{q} \quad (4.5)$$

Se obtienen las variables de estado.

$$x_1 = q_d - q \quad (4.6)$$

$$\dot{x}_1 = -\dot{q} = x_2 \quad (4.7)$$

$$\dot{x}_2 = \ddot{q} = M^{-1}(q) [K_p\tilde{q} - K_v\dot{q} - C(q, \dot{q})\dot{q}] \quad (4.8)$$

Se obtiene la ecuación de lazo cerrado:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} -x_2 \\ M^{-1}(K_px_1 - K_vx_2 - C(q, \dot{q})x_2) \end{bmatrix} \quad (4.9)$$

Evaluando la ecuación (4.9) homogénea con $\dot{x} = 0$ se tiene:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -x_2 \\ M^{-1}(K_px_1 - K_vx_2 - C(q, \dot{q})x_2) \end{bmatrix} \quad (4.10)$$

Considerando el sistema homogéneo, es decir $\dot{x} = 0$, el punto de equilibrio es $x = (x_1, x_2) = (0, 0)$ se encuentra en el origen, es único y existe.

Capítulo 5

Plataforma de simulación

Una plataforma gráfica de simulación o simulador permite reproducir el comportamiento de un sistema mediante las ecuaciones matemáticas que los describen y brindar un ambiente gráfico de interpretación de datos dependientes de los estímulos internos y externos, para que el usuario final entienda dicho comportamiento. Por esta razón los simuladores son herramientas utilizadas en el diseño y creación de prototipos [10,11].



Figura 5.1: Plataforma de simulación

La evolución de los simuladores se ha dado en función a los avances de las computadoras y de los lenguajes de programación; los simuladores actualmente son una herramienta poderosa, exacta y amigable en la interacción con el usuario. Es importante recalcar que un simulador es una herramienta que requiere del modelo dinámico del sistema y de una estructura de control que nos permita realizar correctamente la tarea asignada [11].

5.1. Diseño gráfico del sistema

El primer paso dentro del diseño del simulador es bosquejar el prototipo a través de vistas, las cuales sirven para tener una referencia del sistema en el espacio (x, y, z) . Entre más detallado esté el dibujo obtendremos una mejor representación gráfica del sistema.

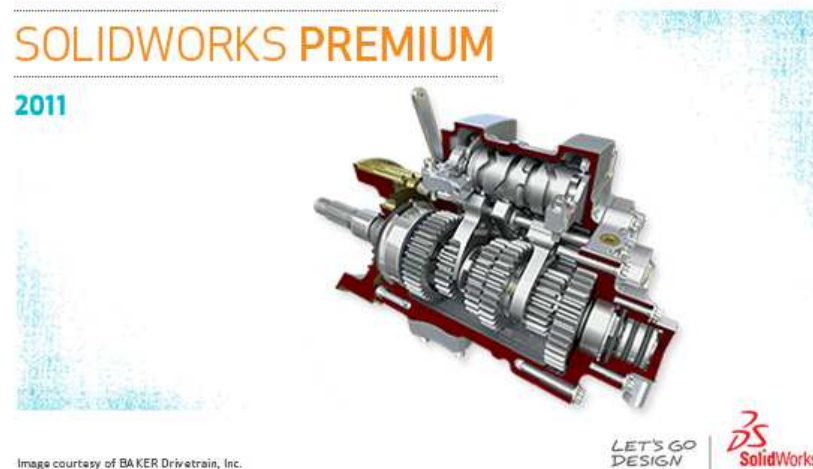


Figura 5.2: SolidWorks®

SolidWorks® es un programa de diseño asistido por computadora para modelado mecánico de piezas. El programa permite modelar piezas y conjuntos y extraer de ellos tanto planos como otro tipo de información necesaria para la producción. Es un programa que funciona con base en las nuevas técnicas de modelado con sistemas CAD. El proceso consiste en traspasar la idea mental del diseñador al sistema CAD, construyendo virtualmente la pieza o conjunto. Posteriormente todas las extracciones (planos y ficheros de intercambio) se realizan de manera bastante automatizada. A continuación se muestran las piezas del sistema en dos tipos diferentes de vistas:

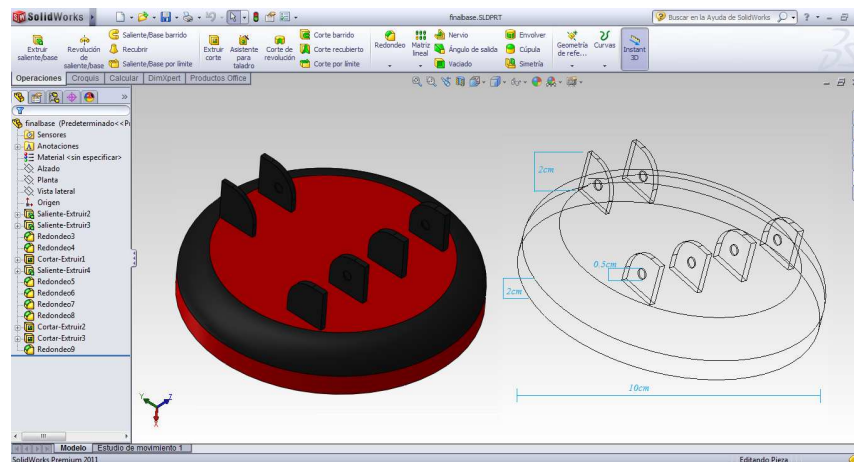


Figura 5.3: Base del sistema

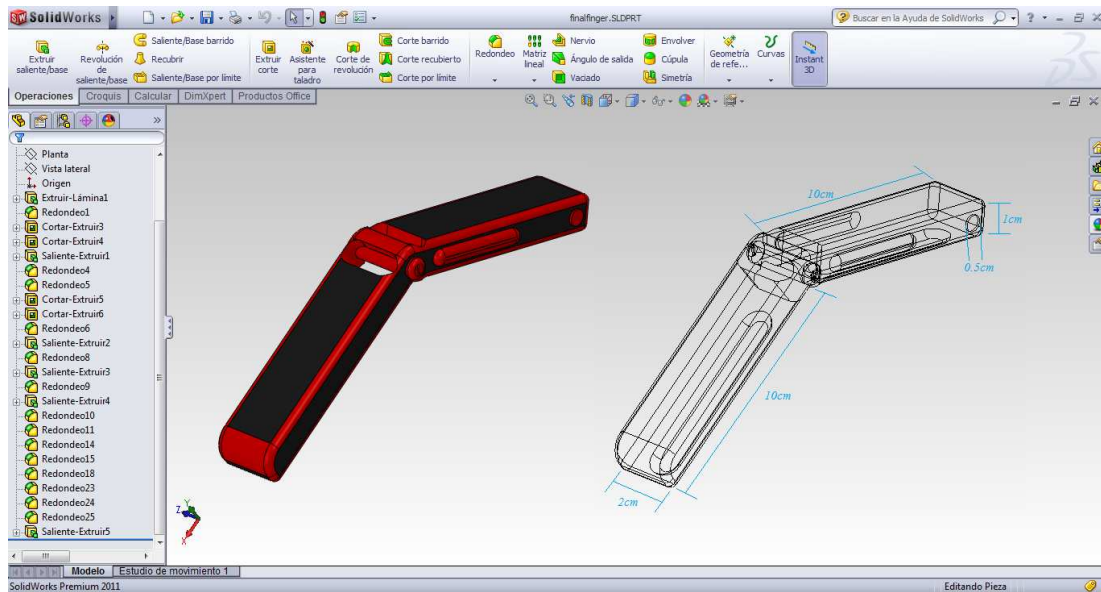
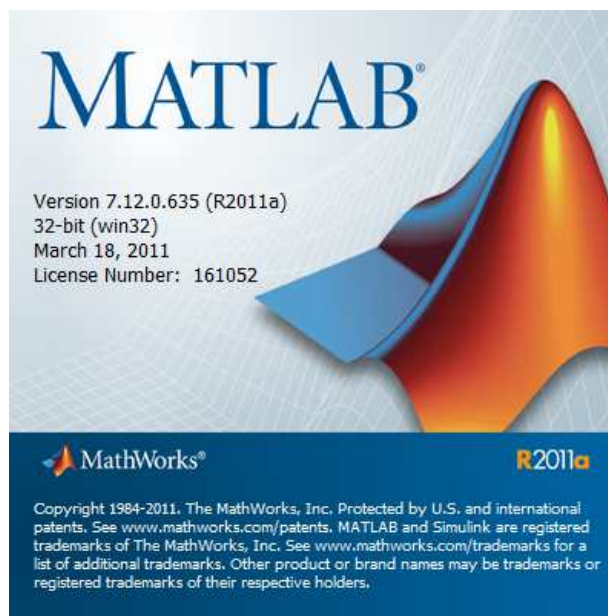


Figura 5.4: Dedo vista superior

MATLAB[®] (abreviatura de **MATrix LABoratory**), es una herramienta de software matemático que ofrece un **entorno de desarrollo integrado (IDE)** con un lenguaje de programación propio conocido como **lenguaje .m** [10].

Figura 5.5: MATLAB[®]

Entre sus aportaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de una **Interfaces Gráfica de Usuario (GUI)** y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

El paquete **MATLAB**[®] dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, *Simulink* (plataforma de simulación multi-dominio) y **GUIDE** (editor de interfaces gráfica de usuario o Graphic User Interface GUI). Además, se pueden ampliar las capacidades de **MATLAB**[®] con las cajas de *herramientas* (toolboxes); y las de **Simulink**[®] con los *paquetes de bloques* (blocksets) [10].

SolidWorks se enlaza con **MATLAB**[®], y para ser más precisos con **Simulink**[®], a partir del cual se genera un diagrama a bloques de nuestro sistema. A dicho diagrama se le aplicará el controlador PD para saber cuál es su comportamiento en cuanto a su tiempo de respuesta, para llegar a una posición deseada, ante una variación de valores en sus constantes K_p y K_v .

GUI (Graphical User Interface) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en dicha interfaz. Tiene las características básicas de todos los programas visuales como **Visual Basic**[®] o **Visual C++**[®] [9].

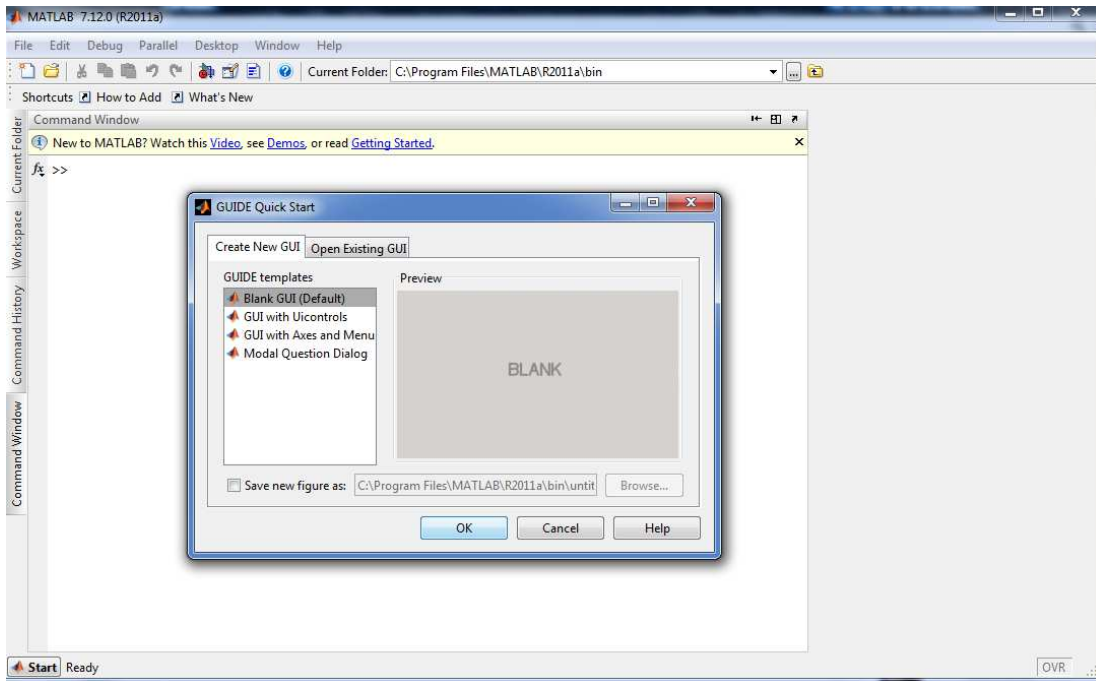


Figura 5.6: GUI

Como se puede observar en el diagrama de flujo, Figura 5.7, tenemos la condición de inicio que inicializará al simulador, con esta condición se producen las llamadas al contenido de **Simulink**[®] y ofrecerá las opciones que tendrá el programa, posteriormente, las siguientes dos opciones que van enlazadas serán las de pausar o detener por completo el proceso que se está efectuando en ese mismo instante.

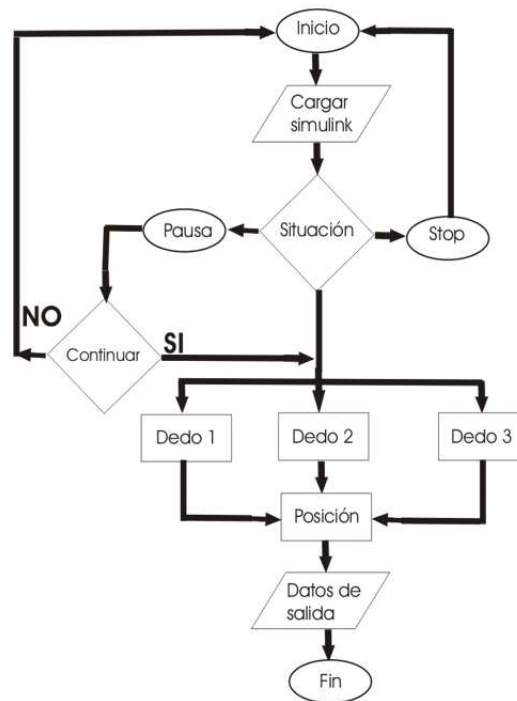


Figura 5.7: Diagrama de flujo

Éstos repercutirán en los grados de libertad (dedos) del efector final, dichos grados de libertad estarán controlados por el usuario final ya sea dando un parámetro de valores o haciéndolo de forma manual para que el sistema realice la tarea que se desea asignar, con lo cual se logrará la posición deseada, la cual contendrá todos los datos necesarios para un buen control de éste mismo, ya que con los datos que se tomen a la salida se comprobará la eficiencia del controlador el cual cumple con los requerimientos para controlar nuestro sistema.

5.2. Simulación

Las simulaciones y los experimentos son técnicas utilizadas para verificar el desempeño de un controlador [10]. Dentro de la etapa de simulación de los algoritmos de control una parte importante es la selección de ganancias dado que el sistema depende directamente de éstas. Por lo que al realizar una mala sintonía de ganancias puede ocasionar resultados no satisfactorios; aunque por otro lado no existen criterios ó una metodología para poder elegirirlas [10, 11]. Una vez que se cuenta con todas las herramientas tanto matemáticas, CAD y de programación, relacionadas en la plataforma de simulación, el siguiente paso consta de realizar las simulaciones necesarias para analizar el comportamiento del sistema. Una simulación ofrece grandes ventajas como la fácil obtención de la información, la reproducción de numerosas pruebas sin sufrir desgaste, el análisis visual del comportamiento, entre otros beneficios.

5.3. Simulación del modelo dinámico

En primera instancia dentro de la etapa de simulación, se programa el modelo dinámico y se analiza su comportamiento. Para nuestro trabajo se hizo uso del programa de **Simulink**® donde creamos un diagrama, Figura 5.8, en donde se utilizan los valores de la τ para verificar el comportamiento del sistema, y que este nos de la respuesta esperada para la posición deseada en tiempo y velocidad.

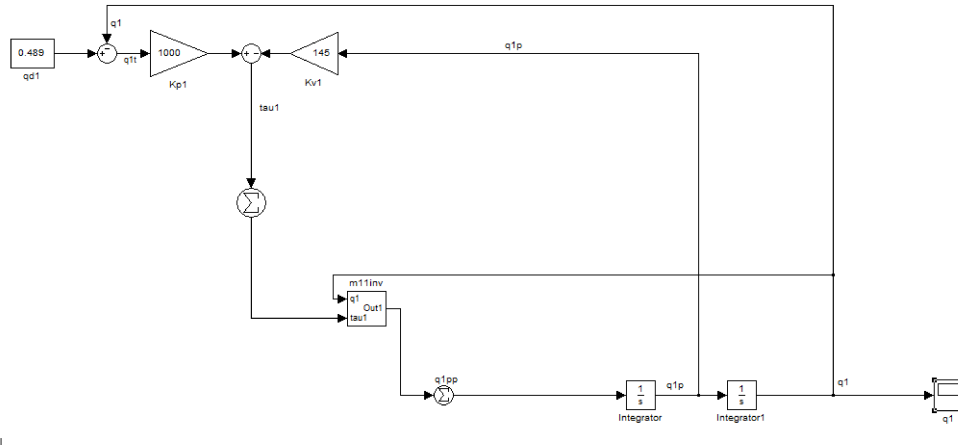


Figura 5.8: Diagrama

En la gráfica siguiente, Figura 5.9, podemos observar la respuesta que nos da nuestro sistema, con los valores exactos en las constantes, y los valores de nuestra τ obtenida.

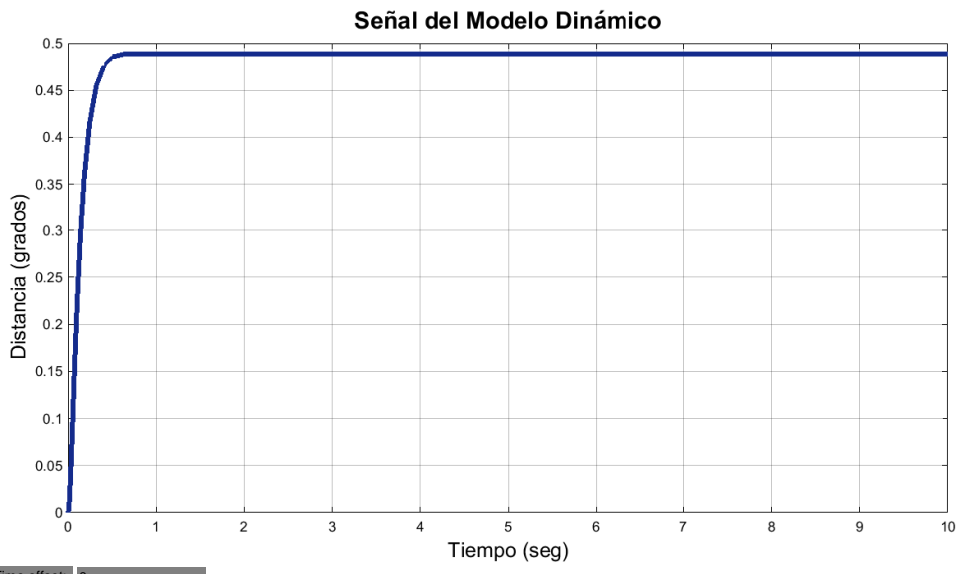


Figura 5.9: Posición deseada

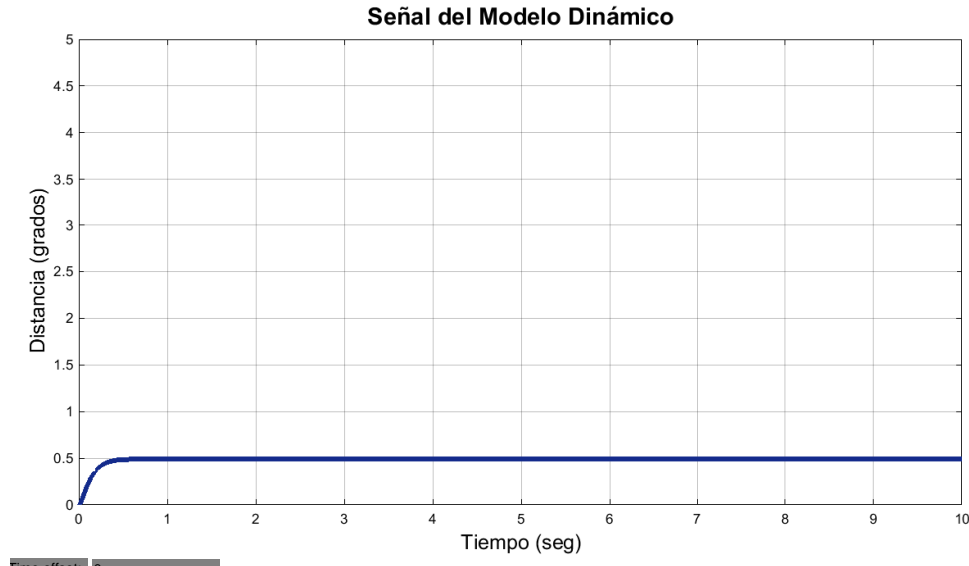


Figura 5.10: Posición deseada

En una sola señal se demuestra el comportamiento de nuestros tres dedos ya que los tres hacen uso del mismo modelo dinámico, solo nuestro tercer dedo tiene la peculiaridad del cambio de signo por lo cual, la misma señal antes mostrada en la Figura 5.9, sólo es invertida, pero sigue conservando el mismo tiempo de respuesta para la simulación del sistema dinámico.

5.4. Simulación del controlador

Ya que tenemos armado nuestro **controlador PD**, proseguimos a hacer su debida simulación para verificar su comportamiento, de igual forma nuestro controlador fue armado y probado con **Simulink®**, en un diagrama a bloques y con sus valores correspondientes que nos darán la señal que deseamos.

Lo que buscamos es que la señal llegue lo mas rápido posible a la posición deseada y con el menor error posible, por tal razón el tiempo se encuentra definido en milisegundos, ya que nuestras constantes estan calculadas para dicho fin, ademas de que nuestro sistema es ideal y no hay fuerzas que generen algun tipo de inestabilidad en la respuesta de salida.

5.5. Simulador propuesto

Inicialmente se genera una ventana en la cual se crea un panel para colocar los botones, características y componentes necesarios para crear la interfaz gráfica con el usuario, esta interfaz será la encargada de permitir manipular la pieza hecha en **SolidWorks®** y realizar el control de la misma a partir del enlace con **Simulink®**. Así pues, permitirá al usuario llevar a diferentes posiciones la garra, de una posición de casa a diferentes ángulos o puntos deseados, modificando los valores de las constantes del controlador PD que es con el cual se realiza el proceso de control.

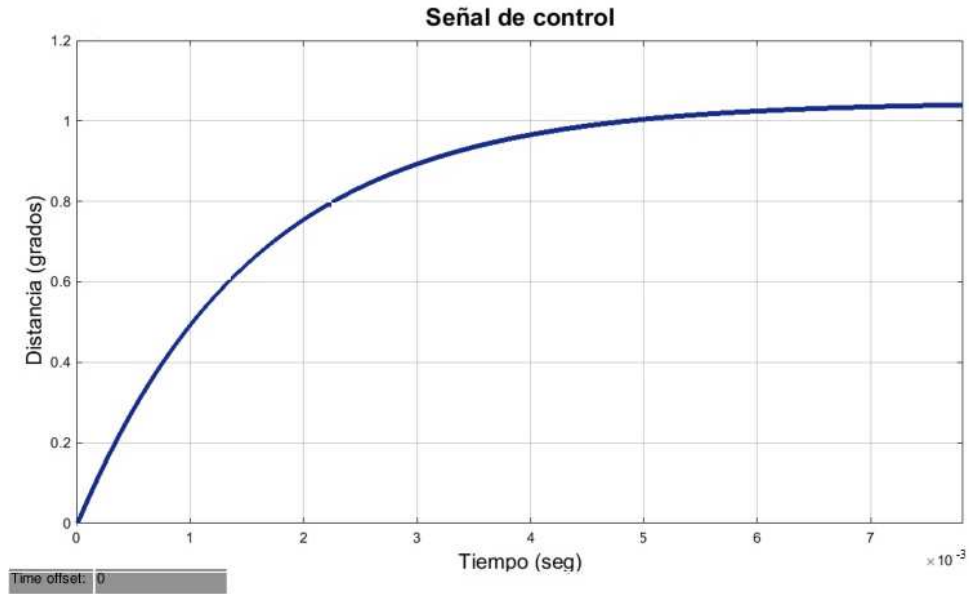


Figura 5.11: Señal del controlador PD

La primera parte consiste en construir nuestro panel mediante el cual daremos las instrucciones debidas a la garra, para lograr una manipulación de esta misma con la precisión debida.

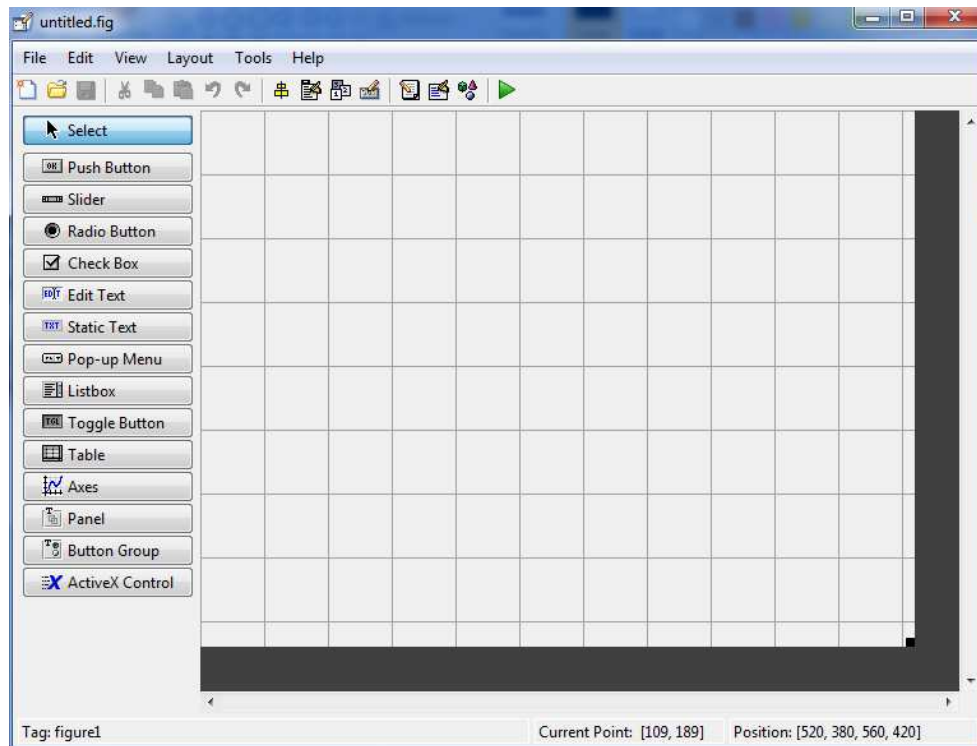


Figura 5.12: Ventana de nuevo panel

A continuación se describe el panel creado en el GUI para la manipulación de la garra:

- **Slider.** Para dar un rango de valores aleatorios a la posición deseada del sistema.
- **Static Text.** Para poner texto fijo y que no pueda ser cambiado por el usuario, en el cual desplegará los valores que sean asignados por el slider.
- **Edit Text.** Para editar texto, a diferencia del Static Text, puede ser modificado por el usuario, y pueden introducirse valores que el usuario desee ingresar, lo contrario que el slider que da un rango de valores predeterminados.
- **Push Button.** Botones para ejecutar ordenes de comando como, pausar, continuar, detener, posición de casa, iniciar simulación, y gráficas.
- **Axes.** Comandos para desplegar imágenes que en nuestro caso la imagen de la garra y el escudo de la facultad.

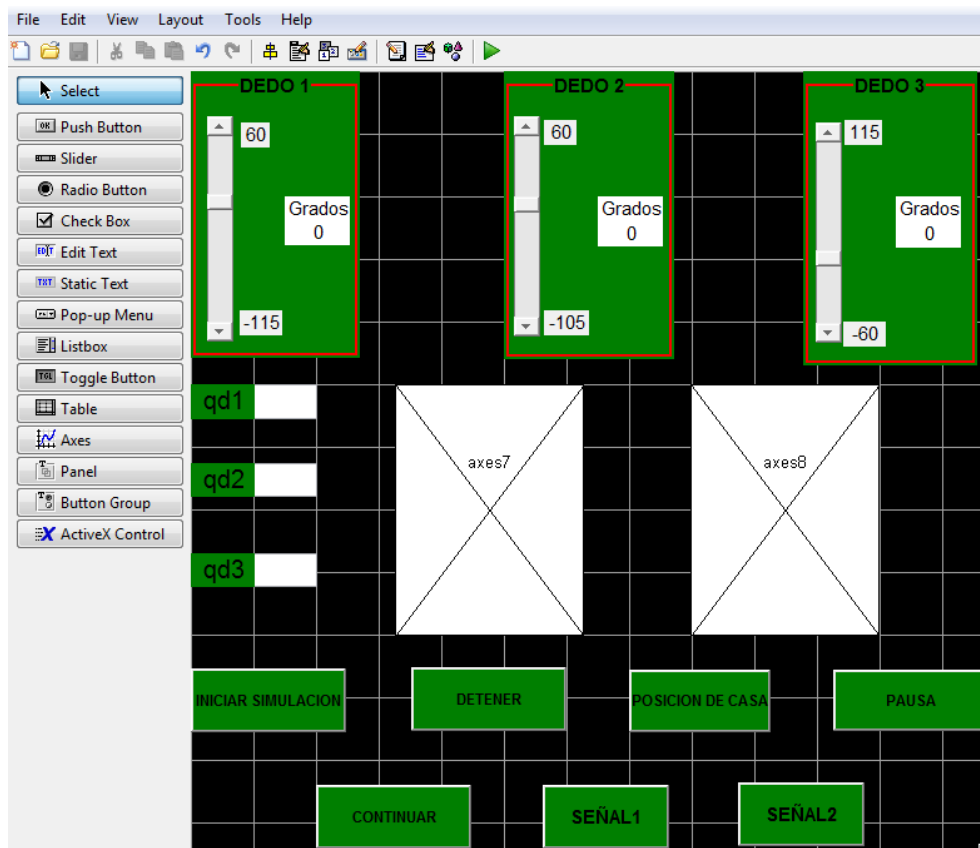
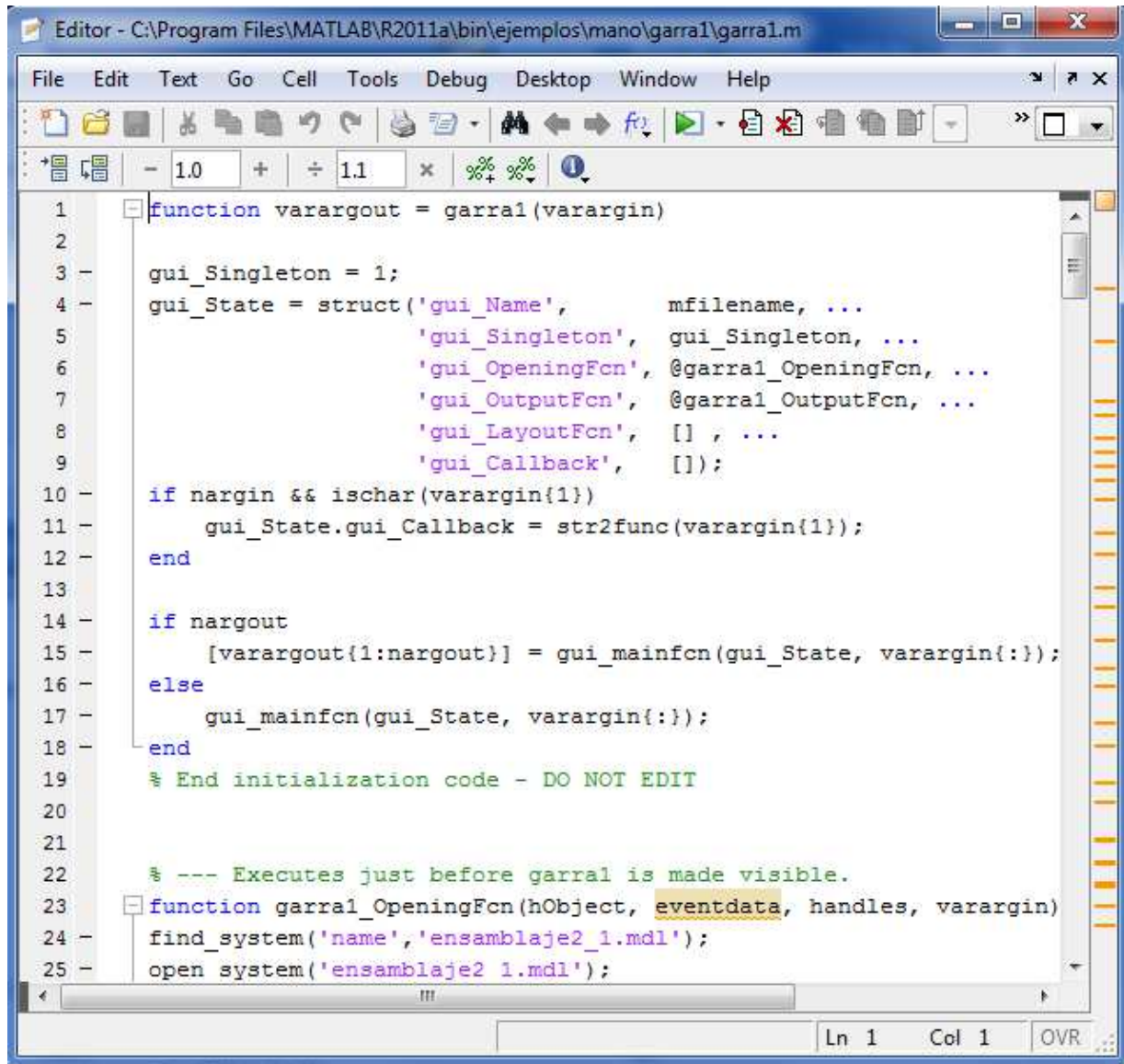


Figura 5.13: Archivo .fig

Una vez que se ha llevado a cabo este proceso se tendrán dos ventanas una de ellas en la cual se tienen los botones para dar las instrucciones al sistema y ésta quedará con extensión `.fig`, Figura 5.13, de acuerdo con la configuración del programa, la segunda ventana nos mostrará el código del programa en el cual se desarrolla el proceso de control, la cual se generará como un archivo `.m`, 5.14 propio de **MATLAB®**.



```

1  function varargout = garra1(varargin)
2
3  gui_Singleton = 1;
4  gui_State = struct('gui_Name',       mfilename, ...
5                    'gui_Singleton',  gui_Singleton, ...
6                    'gui_OpeningFcn', @garra1_OpeningFcn, ...
7                    'gui_OutputFcn',  @garra1_OutputFcn, ...
8                    'gui_LayoutFcn',  [], ...
9                    'gui_Callback',    []);
10
11  if nargin && ischar(varargin{1})
12      gui_State.gui_Callback = str2func(varargin{1});
13  end
14
15  if nargin
16      [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
17  else
18      gui_mainfcn(gui_State, varargin{:});
19  end
20  % End initialization code - DO NOT EDIT
21
22  % --- Executes just before garra1 is made visible.
23  function garra1_OpeningFcn(hObject, eventdata, handles, varargin)
24      find_system('name','ensamblaje2_1.mdl');
25      open_system('ensamblaje2_1.mdl');

```

Figura 5.14: Archivo `.m`

Al ejecutar nuestro programa se ejecutan dos ventanas, la del panel de control que se ha creado para dar las instrucciones deseadas, y la de nuestro sistema a controlar, la cual es enlazada desde **Simulink**[®] hacia nuestro GUI.

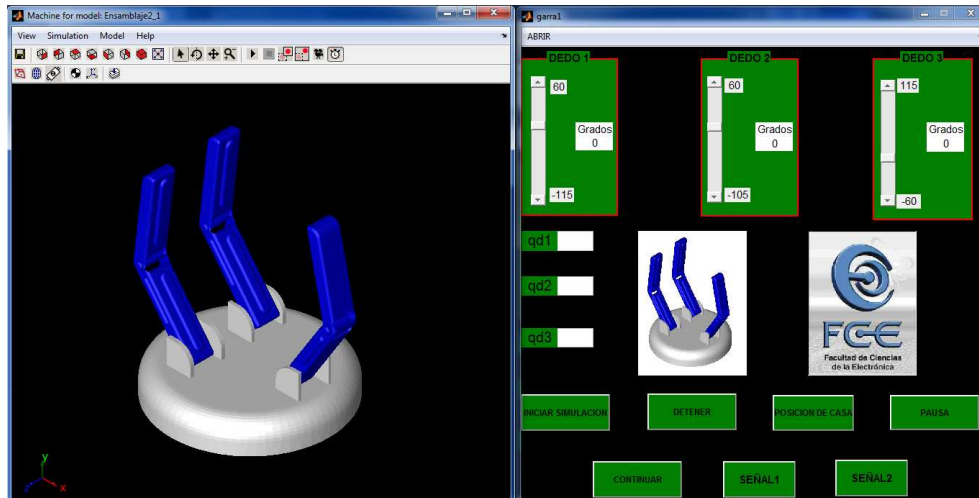


Figura 5.15: Sistema y Panel de Control

Logrando este enlace de **Simulink**[®] con *GUI* logramos tener el control sobre las posiciones a las que deseamos que nuestro sistema llegue, sin la necesidad de recurrir a **Simulink**[®] de una forma directa, los sliders nos ayudarán a dar la posición que deseamos como se muestra en la 5.16, nuestro sistema alcanza su máxima posición al hacer el cierre de la garra.

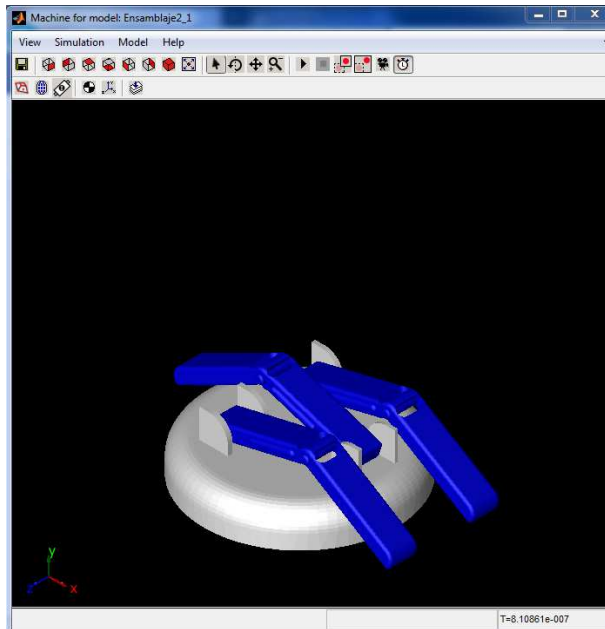


Figura 5.16: Garra cerrada

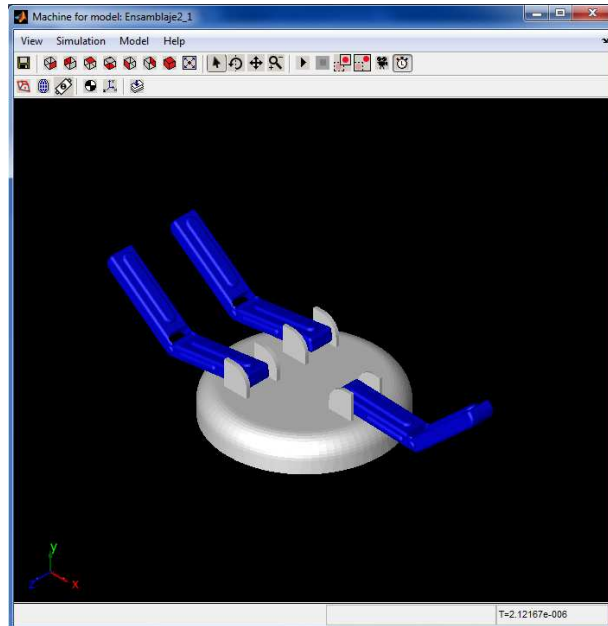


Figura 5.17: Garra abierta

A continuación mostramos nuestra garra en su posición máxima al ser abierta para poder tomar un objeto, las articulaciones deben tener los grados precisos para que no hagan un choque con la base o un force con la misma, para ello el slider está definido con sus grados máximos y mínimos. De igual forma para declarar las posiciones deseadas a nuestro sistema lo podemos hacer introduciendo los grados a los que desearemos que nuestro sistema llegue, en las respectivas q_1, q_2, q_3 .

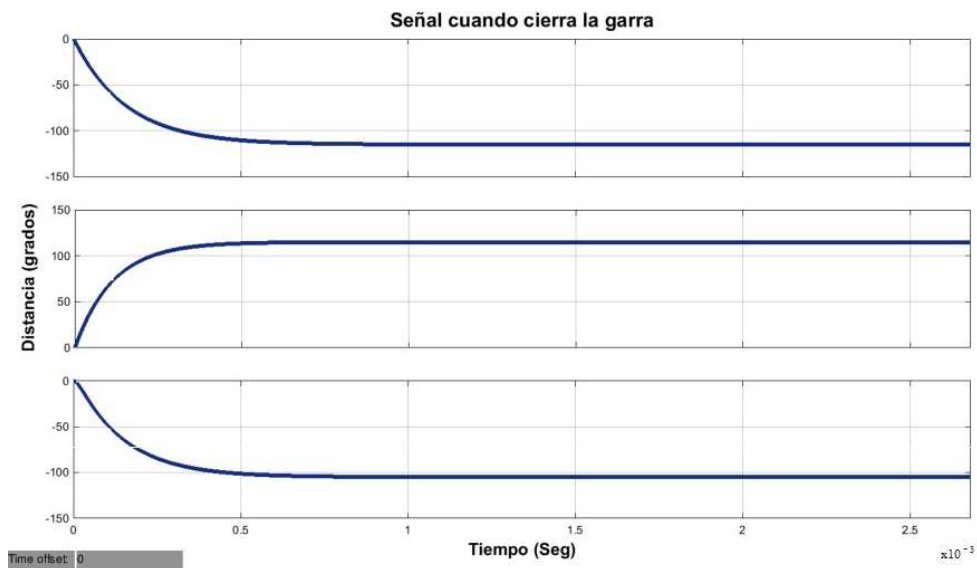


Figura 5.18: Señal cuando cierra la garra

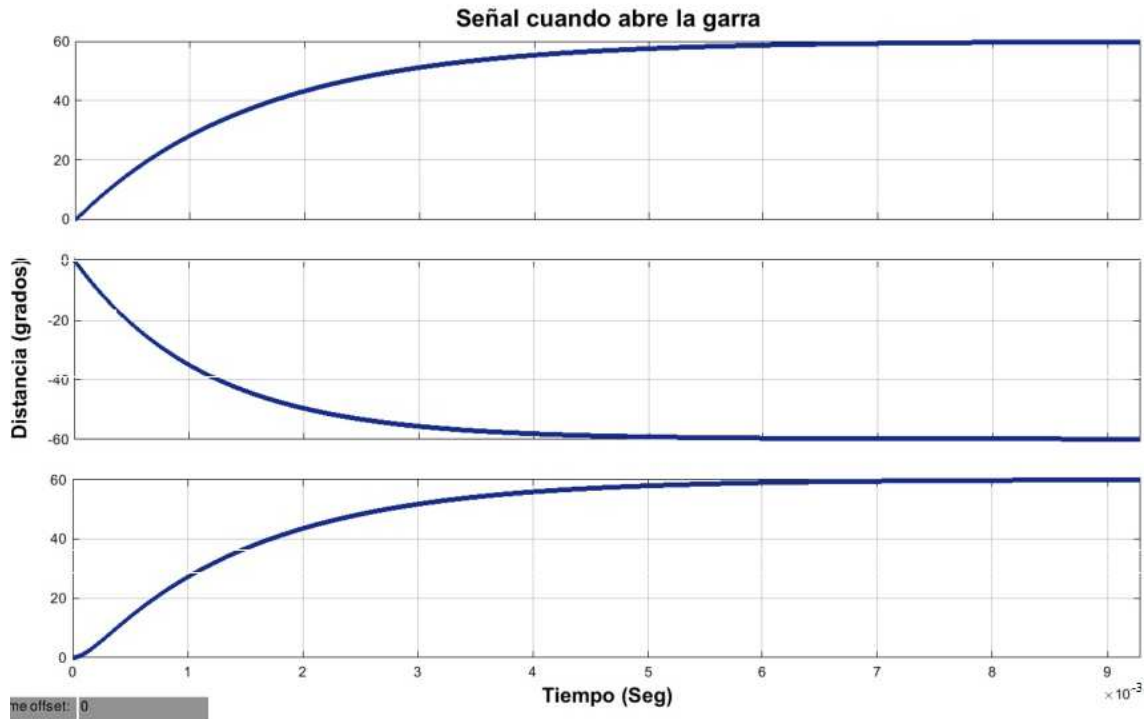


Figura 5.19: Señal cuando abre la garra

En la Figura 5.18 podemos observar las señales desplegadas de acuerdo a la trayectoria seguida por nuestro sistema, las cuales tienen un tiempo de respuesta en milisegundos, sin perturbaciones durante su trayectoria, lo cual hace a nuestro controlador ideal para el manejo de nuestro sistema.

La Figura 5.19 nos muestra las señales cuando nuestra garra realiza el movimiento de abierto hasta su punto máximo, y vemos como las señales se despliegan invertidas por ser el movimiento opuesto de nuestra garra, y que son señales depuradas con un tiempo de respuesta en milisegundos sin presentar perturbaciones en su trayecto.

Conclusiones

Finalmente solo nos resta realizar las conclusiones de éste trabajo de tesina, cuyo objetivo se ha cumplido el cual fue el desarrollo de una plataforma gráfica de simulación que resuelva la problemática de control de posición, mediante el uso de controladores y la utilización de técnicas de programación.

Debemos entender que para desarrollar un efector final necesitamos conocer en primera instancia la historia que hay detrás de ello, los distintos tipos de efectores finales que existen, así como sus distintos tipos de aplicaciones a los que están sujetos, entender lo más básico de este tipo de sistemas para posteriormente llegar a comprender lo más complejo de estos mismos, como han ido evolucionado y cuales son sus limitantes.

Así como debemos conocer lo más básico de los efectores finales, de la misma forma debemos tener conocimiento de la historia que hay detrás de las plataformas gráficas de simulación, las cuales brindan un entorno de realidad al usuario final.

El uso de conceptos físicos y matemáticos y su respectivo desarrollo, los cuales nos darán una amplia comprensión del comportamiento que tendrá nuestro sistema a controlar.

Apéndice A

Manual de usuario

A continuación presentamos un manual de usuario, el cual nos mostrará cómo manipular el panel de control para la manipulación de una garra robótica de tres dedos.

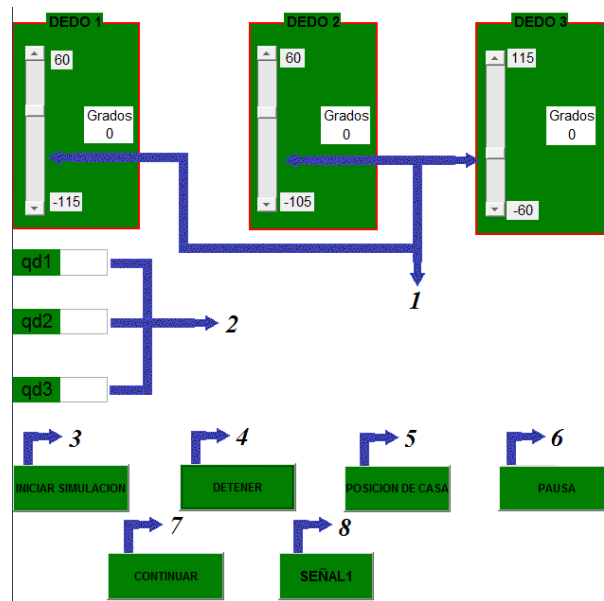


Figura A.1: Panel de control

- B1.** Sliders . Dan un parámetro de valores predeterminados que van de un máximo de 60 grados para la abertura de la garra, hasta un mínimo de -115 grados para realizar el cierre.
- B2.** q_1 , q_2 , q_3 . El usuario final puede dar valores más exactos, manejando números decimales para una mejor manipulación de la garra.
- B3.** Iniciar simulación. Inicia la simulación de la garra de acuerdo a los valores previamente dados.
- B4.** Detener . Detiene el proceso en su totalidad sin importar que este no halla finalizado.

- B5.** Posición de casa. Regresa los dedos de la garra a su posición de inicio después de haber completado la simulación o de haberla detenido a mitad de proceso.
- B6.** Pausa. Detiene momentáneamente el proceso que está efectuando la garra.
- B7.** Continuar. Vuelve a reanudar el proceso de la garra después de haber sido pausado por el usuario final.
- B8.** Señal. Después de haber conseguido la trayectoria deseada, presionamos el botón detener, seguido del comando seal, el cual nos muestra las señales de las trayectorias por separado.

Bibliografía

- [1] Fundamentos de robótica 2Ed. Antonio Barrientos, Luis Felipe P. Carlos Balaguer, Rafael Aracil. Ed. Mc Graw Hill.
- [2] F.M Sánchez Martín, F. Millian Rodríguez and F. Salvador Bayarri. Historia de la Robótica I.
- [3] Masters of Manufacturing: Joseph F. Engelberger. Society of Manufacturing Engineers.
- [4] R.Kelly and V. Santibañez, Control Automiático e Informática Industrial, (Ensenada, Mex.: Pearson Education, 2003).
- [5] Ingeniería de Control Moderna. Katsuhiko Ogata. Segunda Edición, Ed. Prentice Hall.
- [6] Ascher, Uri M; Petzold, Linda Ruth (1998). Computers methods for ordinary differential equations and differential-algebraic equations (1 edition). Philadelphia (USA).
- [7] A. Barrientos, L.F. Peñín and C. Balaguer, Fundamentos de Robótica, (Madrid: Mc Graw Hill, 1997).
- [8] S. Zemansky and Y. Freedman, Física Universitaria, (Madrid, Esp,: Pearson Educación, 1999).
- [9] Manual de Interfaz Gráfica de Usuario en Matlab, Parte 1. Diego Orlando Barragán Guerrero.
- [10] Matlab para Ingenieros. Holly Moore, Ed. Pearson, Prentice hall.
- [11] Modern Control Design with Matlab and Simulink. Ashish Tewari
- [12] Simulink Aplicado a Sistemas de Control. Paula Andrea Ortiz Valencia y Alexander Arias