



Universidad Autónoma de Querétaro
 Facultad de Ingeniería
 Maestría En Ciencias (Instrumentación y Control Automático)

Diseño de un control difuso mediante un PSO para el seguimiento de trayectorias de un robot de 2DOF.

TESIS

Que como parte de los requisitos para obtener el grado de:
 Maestro en Ciencias (Instrumentación y Control Automático)

Presenta:

Ing. Emmanuel Alejandro Rodelas Domínguez

Dirigido por:

M. C. César Oswaldo Mendoza Herbert

M.C. César Oswaldo Mendoza Herbert
 Presidente


 Firma

Dr. Manuel Toledano Ayala
 Secretario


 Firma

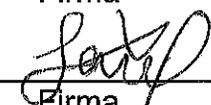
Dr. Edgar Alejandro Rivas Araiza
 Vocal

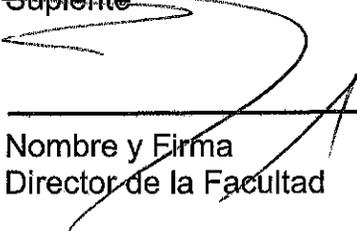

 Firma

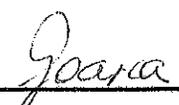
Dr. Roberto Valentín Carrillo Serrano
 Suplente


 Firma

Dr. Saúl Tovar Arriaga
 Suplente


 Firma


 Nombre y Firma
 Director de la Facultad


 Dra. Ma. Guadalupe Flavia Loarca Piña
 Directora de Investigación y Posgrado

RESUMEN

Los robots manipuladores utilizan sistemas de control para realizar sus movimientos, estos permiten llevar a los eslabones del robot a la posición deseada. Entre los controladores de robots manipuladores se encuentran los PD, PD con pre-compensación, PID, difuso, feed forward por nombrar a algunos. De entre los diferentes controladores se analiza el controlador PD y el difuso.

El controlador PD realiza su acción mediante la sintonización de ganancias K_p y K_v que reciben el nombre de ganancias de posición y velocidad respectivamente y para sintonizarlas es necesario seguir ciertas reglas.

Un controlador difuso, es un controlador basado en lógica difusa, esto quiere decir que no necesita conocer completamente el modelo matemático del sistema si no cómo se comporta realmente y a partir de este comportamiento se construye el controlador basado en reglas y funciones de membresía.

Ambos controladores requieren sintonización para su funcionamiento óptimo, esta sintonización se logra mediante un PSO. Un PSO es un algoritmo de optimización que permite encontrar valores óptimos para funciones, en este caso las funciones que describen el movimiento del robot.

Para el control PD se varían las ganancias K_p y K_v , para el controlador difuso se proponen funciones de membresía gaussianas en el centro y funciones trapezoidales a los extremos y el PSO varía los parámetros de estas funciones. En ambos controladores se analiza su respuesta ante variaciones.

La trayectoria propuesta para evaluar los controladores es senoidal y se puede observar que el controlador difuso responde más rápido ante las variaciones de carga, esto quiere decir que al modificarse el peso de un eslabón este controlador reacciona más rápido al reducir el error.

Queda abierta la posibilidad de utilizar un algoritmo MOPSO (Multiple Object Particle Swarm Optimization) y compararlo con el PSO y el controlador clásico, así como utilizar diferentes funciones de membresía en los extremos para contrastar el comportamiento.

Palabras clave: Controlador Difuso, Sistema 2-DOF, PSO, modelo cinemático, trayectoria, controlador clásico, optimización.

TABLA DE CONTENIDOS

1. INTRODUCCIÓN	7
1.1 ANTECEDENTES	8
1.1.1 <i>Controladores para sistemas de dos grados de libertad</i>	8
1.1.2 <i>Lógica difusa</i>	9
1.2 JUSTIFICACIÓN	10
1.3 HIPÓTESIS	11
1.4 OBJETIVO	11
1.4.1 <i>Objetivos Particulares:</i>	11
2. MARCO TEÓRICO	12
2.1 LÓGICA DIFUSA	12
2.1.1 <i>Interfaz de difusificación</i>	13
2.1.2 <i>Base de conocimientos</i>	18
2.1.3 <i>Lógica de decisiones</i>	19
2.1.4 <i>Interfaz de desdifusificación</i>	20
2.1.5 <i>Ganancias de escalamiento</i>	21
2.2 SISTEMA 2DOF (TWO DEGREES OF FREEDOM)	22
2.2.1 <i>Modelo dinámico de sistema 2DOF</i>	22
2.3 CONTROLADOR PD CON PRE-COMPENSACIÓN	24
2.3.1 <i>Procedimiento de sintonía de controlador PD</i>	26
2.4 CONTROLADOR DIFUSO	28
2.4.1 <i>Funciones de membresía</i>	29
2.4.2 <i>Reglas del controlador</i>	30
2.5 PSO	32
2.5.1 <i>Actualización de velocidad</i>	35
2.5.2 <i>Actualización de posición</i>	36
3. DESARROLLO	38
3.1 METODOLOGÍA	38
3.2 DEFINICIÓN DE TRAYECTORIA	40
3.3 MODELO DINÁMICO DEL SISTEMA	42
3.4 MATRICES DE POSICIÓN, VELOCIDAD Y ACCELERACIÓN DESEADAS	43

3.5	DISEÑO DEL CONTROLADOR.....	44
3.5.1	<i>Diseño mediante procedimiento de sintonía.....</i>	45
3.5.2	<i>Diseño de controlador PD con pre-compensación mediante PSO</i>	47
3.5.3	<i>Diseño de controlador difuso mediante PSO.....</i>	49
4.	RESULTADOS	53
4.1	CONTROLADOR PD CON PRE-COMPENSACIÓN Y PSO	53
4.2	CONTROLADOR DIFUSO CON PSO.....	55
4.3	VARIACIÓN DE CARGA.....	58
4.3.1	<i>Gráficas controlador PD con pre-compensación</i>	59
4.3.2	<i>Gráficas de controlador difuso.....</i>	61
5.	CONCLUSIONES	68
6.	BIBLIOGRAFÍA.....	70

TABLA DE FIGURAS

2.1	Estructura de controlador difuso.....	12
2.2	Función de membresía trapezoidal	16
2.3	Función de membresía triangular.....	17
2.4	Función de membresía Gaussiana.....	18
2.5	Robot de 2 DOF	22
2.7	Función de membresía de entrada (derivada de error)	29
2.8	Función de membresía de entrada (error).....	29
2.6	Función de membresía de salida (torque).....	30
2.9	Algoritmo PSO.....	34
2.10	Actualización de velocidad y posición	37
3.1	Metodología propuesta.....	40
3.2	Posiciones deseadas de trayectoria a seguir	41
3.3	Velocidades deseadas de trayectoria a seguir	41
3.4	Aceleraciones deseadas de trayectoria a seguir	42

3.5 Estructura de sintonización de controlador PD con pre-compensación	47
3.6 Estructura de sintonización de controlador PD con pre-compensación	50
4.1 Evolución de la función objetivo	53
4.2 Error de posición PD con pre-compensación	54
4.3 Trayectoria actual y deseada	55
4.4 Inicio de trayectoria PD con pre-cálculo	55
4.5 Evolución de función objetivo	56
4.6 Funciones de membresía de entrada y salida de articulación 1	56
4.8 Trayectoria actual y deseada	57
4.7 Funciones de membresía de entrada y salida de articulación 2	57
4.9 Inicio de trayectoria difuso calculado con PSO	58
4.10 Aumento 10gr en masa 2 en t/4	59
4.11 Aumento 20gr en masa 2 en t/4	59
4.12 Aumento 20gr en masa 2 en t/2	60
4.13 Reducción 10gr en masa 2 t/2.....	60
4.14 Aumento 10gr en masa 1, t/4	61
4.15 Aumento 10gr en masa 2 en t/4	61
4.16 Aumento 20gr en masa 2 en t/4	62
4.17 Aumento 20gr en masa 2 en t/2	62
4.18 Reducción de 10gr en masa 2 en t/2.....	63
4.19 Aumento 10gr en masa 1 en t/4	63
4.20 Error de posición PD abajo, difuso arriba.....	65
4.21 Error de posición 20gr masa 2, t/2, difuso arriba, PD abajo	66
4.22 Error posición 10gr masa 1, PD arriba, difuso abajo.	67

1.INTRODUCCIÓN

El presente trabajo tiene como objetivo la comparación de dos tipos de controladores, un controlador clásico y un controlador inteligente aplicados al seguimiento de una trayectoria para un robot de dos grados de libertad.

El diseño del controlador clásico para estos sistemas está definido ya que es un modelo matemático, en cambio para el controlador inteligente el modelo depende del conocimiento que la persona posea sobre el sistema por lo que la mayoría de las veces es específico del sistema. Para reducir la participación de un experto en el tema se emplea el algoritmo de optimización.

Un algoritmo de optimización permite encontrar los parámetros que tengan la mejor respuesta para el sistema evaluándolo en una función objetivo. Con la optimización del controlador inteligente y el controlador clásico mediante el mismo método se comparan los dos controladores y se observa la respuesta al seguir la trayectoria propuesta.

En el primer capítulo se mencionan brevemente los sistemas de dos grados de libertad, se indica el tipo de controlador que utilizan estos sistemas y se mencionan aspectos relevantes de cada uno. Se muestra la hipótesis del proyecto, la justificación y los objetivos.

En el segundo capítulo se mencionan las herramientas a utilizar para el desarrollo del proyecto. Se conoce el modelo dinámico del sistema de dos grados de libertad, la lógica difusa, los algoritmos de optimización en especial el PSO, los controladores clásicos y el parámetro de obtención de error.

El capítulo tres expone la metodología a tratar, desde la trayectoria a utilizar, las matrices del sistema y los controladores empleados. Aquí se presentan los parámetros escogidos para el proyecto.

El capítulo cuatro abarca los resultados obtenidos del desarrollo, son expuestos y contrastados entre ellos.

El capítulo cinco es el análisis de resultados donde se presenta si el proyecto cumple o no con los objetivos planteados al inicio, en el capítulo uno.

1.1 Antecedentes

Un robot manipulador es una serie de eslabones en forma de brazo articulado unidos mediante juntas que permite el movimiento individual y relativo entre eslabones consecutivos. Los robots manipuladores tienen articulaciones o uniones que pueden ser rotacionales, traslacionales o mixtas. La relación de posición entre articulaciones se obtiene mediante posiciones en forma de ángulos o distancias dependiendo del tipo de articulación del robot.

Los robots con articulaciones rotacionales son manipulados mediante el movimiento de sus ángulos, para poder controlarlos se necesita un marco de referencia base y la relación que existe entre los diferentes eslabones.

Para controlar un robot o un brazo robótico de esta forma se debe conocer el modelo del sistema y usar un controlador que permita que la condición de error sea cero, esto quiere decir que la posición del robot se cumpla.

Las aplicaciones de los robots son tanto industriales como didácticas y su uso es cada vez más frecuente. Debido a que los modelos de los robots son no lineales las técnicas tradicionales de control se vuelven limitadas y nuevas técnicas de precisión son requeridas. Dentro de estas técnicas la aplicación de controladores basados en el modelo dinámico del robot son muy usadas así como controladores inteligentes que permiten lidiar con las no linealidades de los sistemas(Kelly *et al*, 2003).

1.1.1 Controladores para sistemas de dos grados de libertad

En septiembre del 2014 se trabajó con un controlador difuso aplicado a un sistema, este controlador se propuso para tener un seguimiento robusto y se comparó con un controlador PID clásico (Ayas *et al.*, 2014) . El rendimiento de ambos controladores se midió utilizando el error cuadrático integral. Todo el

sistema fue probado en Matlab/Simulink y los resultados de las simulaciones mostraron que el error en el seguimiento es menor con el controlador difuso y que es efectivo controlando la incertidumbre y ambigüedad que presenta el sistema.

Ese mismo año, 2014, en China se desarrolló un controlador incremental difuso para mejorar la respuesta al seguimiento de señales sinusoidales, las reglas del control difuso fueron establecidas por un método determinístico sistemático (Liu *et al.*, 2014). El sistema fue programado en Matlab/Simulink y ejecutado por una computadora usando RT-Lab. Se demostró que el controlador difuso pudo seguir la señal sinusoidal de forma precisa comparándolo con el controlador clásico PID.

En India se realizó investigación con diferentes tipos de controladores difusos para un manipulador paralelo electrohidráulico de dos DOF. Se compararon el controlador proporcional integral derivativo (PID), el PID-difuso híbrido, el PID-difuso auto-ajustado y el PID-difuso auto-ajustado parcial (Ghosh *et al.*, 2015). El controlador se diseñó en Labview, se comunicó el sistema con una tarjeta NI-cRIO para la adquisición de datos y un FPGA fue utilizado para insertar el diseño de control. Se concluyó que el controlador PID-difuso auto-ajustado parcial se comportaba más rápido y suave que los otros controladores.

1.1.2 Lógica difusa

La lógica difusa es un grupo especial de métodos matemáticos que hacen posible incluir en un modelo impreciso o vagamente formulado la información de un experto que la mayoría de las veces es caracterizada como lenguaje natural. El desarrollo de modelos difusos es exitoso debido a que dan una solución en situaciones donde los modelos con matemáticas tradicionales fallan. La idea de incluir información imprecisa en los modelos contradice los requerimientos tradicionales que son precisión en la información, sin embargo, la razón por la que se utilizan es afrontar la discrepancia de relevancia frente precisión. Este principio llamado *principio de incompatibilidad* y la formulación de la lógica difusa fue desarrollada por L. A. Zadeh.

El principio de incompatibilidad menciona que mientras más complejo es un sistema es más difícil hacer declaraciones absolutas y precisas sobre él. En un punto debe haber una relación entre precisión y relevancia. El incremento de precisión puede ser obtenido solamente mediante el decremento de relevancia, el incremento en relevancia puede ser obtenido mediante el decremento en precisión.

Para expresar relevancia en información se necesita el lenguaje natural. Esta es la única herramienta completa que permite trabajar efectivamente con conceptos vagos.

1.2 Justificación

Los controladores utilizados para el movimiento de un sistema de dos grados de libertad emplean controladores clásicos que si bien describen el sistema, su obtención presenta cierto grado de complejidad al incluirse perturbaciones y son poco flexibles a las variaciones de parámetros. Los controladores difusos proporcionan un marco de referencia eficiente y sistemático para tratar imprecisiones y no linealidades en sistemas complejos, debido a esto se propone su uso.

Para implementar un control clásico, se debe conocer el modelo del sistema describiéndolo por medio de ecuaciones diferenciales encontrando una función de transferencia que describa las relaciones del sistema propuesto. Estas ecuaciones la mayoría de las veces tienen un alto grado de complejidad y se realizan suposiciones para simplificarlas (Yin and Yin, 2007).

En los trabajos revisados se han desarrollado sistemas de control que describen superficialmente la integración de los subsistemas, algunos muestran los tiempos de respuesta y gráficas del sistema a los parámetros requeridos (Chen *et al.*, 2007), sin embargo todo esto realizado con un control clásico para el modelo y enfocándose a la obtención de las ecuaciones que describen al sistema

por lo que un controlador difuso para este tipo de sistemas permite flexibilidad a la entrada que se describe como una variación en el peso de sistema.

En el control de sistemas de dos grados de libertad está documentado que los controladores difusos tienen mejor respuesta a seguir trayectorias definidas, las señales sinusoidales son usadas la mayoría de las veces como entradas del sistema (Liu *et al.*, 2014) y estos sistemas son programados en Matlab.

1.3 Hipótesis

Un controlador difuso para un sistema de dos grados de libertad disminuye el error de los eslabones al seguir una trayectoria propuesta comparándolo un controlador clásico.

1.4 Objetivo

Diseñar un controlador basado en lógica difusa para un sistema de dos grados de libertad que reduzca el error de los eslabones al seguir una trayectoria comparándolo con un controlador clásico.

1.4.1 Objetivos Particulares:

- Obtener el modelo del sistema.
- Definir la trayectoria.
- Optimizar el controlador clásico mediante el PSO y obtener las gráficas de respuestas del seguimiento de la trayectoria.
- Obtener el modelo difuso seleccionando las variables de entrada, el proceso de difusificación, la base de reglas y el método de desdifusificación.
- Optimizar el controlador difuso mediante el PSO y obtener las gráficas de respuestas del seguimiento de la trayectoria.

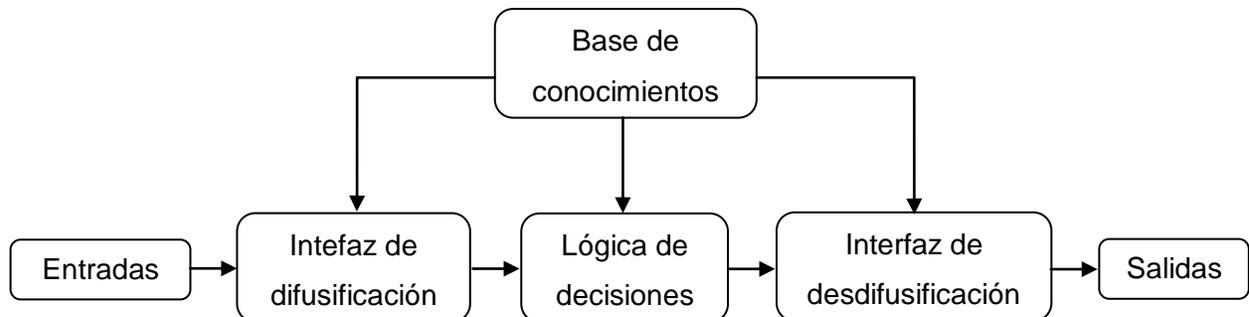
2.MARCO TEÓRICO

La teoría utilizada para este proyecto de investigación enfocado al control de una plataforma abarca diferentes áreas: control clásico, control inteligente, robótica y algoritmos de optimización. Ya que cada tema se utiliza en la metodología es conveniente dar un panorama general de cada uno de los sub temas involucrados en este proyecto.

2.1 Lógica Difusa

La lógica difusa tiene base en valores no específicos como cierto o falso si no en un grado de pertenencia a lo cierto y un grado de pertenencia a lo falso. Este tipo de lógica permite visualizar el problema de forma similar a como una persona lo haría o como el operador de una máquina la controlaría, debido a esto es una ventaja que se tenga la experiencia de conocer el sistema, sin embargo hay métodos que permiten realizar un controlador difuso sin necesidad haber tenido experiencia con el sistema.

Un sistema difuso está compuesto por cuatro diferentes partes, estas son: la interfaz de difusificación, la base de conocimientos, la lógica de decisiones y la interfaz de desdifusificación. La Fig. 2.2 muestra las partes que lo conforman.



2.1 Estructura de controlador difuso

2.1.1 Interfaz de difusificación

La primera fase de la lógica difusa es la interfaz de difusificación, la interfaz de difusificación se encarga de medir los valores de las variables de entrada para escalarlos y pasarlos a valores en un rango que sea adecuado a las variables del universo del controlador. Los valores de la entrada deben variar entre cero y uno, donde el cero significa que la variable no es un miembro del conjunto difuso y el valor de uno significa que la variable es completamente un miembro del conjunto difuso.

Los parámetros de entrada se difusifican mediante el uso de funciones de membresía pre-definidas por el usuario. Esta parte del diseño es fundamental ya que funciones simples harán el cómputo más sencillo evitando sobre-cargar la implementación.

2.1.1.1 Conjuntos difusos y funciones de membresía

Un conjunto difuso es definido por Lotfi A. Zadeh como una clase con un grado de membresía continuo (Bede, 2013). Para comprender de mejor forma este concepto se debe relación el conjunto difuso con un conjunto clásico. Así, la función característica μ_A de un conjunto clásico A se puede generalizar como una función de membresía $\mu_{\bar{A}}$ para un conjunto difuso \bar{A} , el cual corresponde a la ecuación (3).

$$\mu_{\bar{A}} : X \mapsto [0,1] \quad \text{Ec. 3}$$

La ecuación (3) representa el grado que tiene el valor respecto a ese conjunto a partir del punto de vista de conjuntos. En general, un conjunto difuso que se definirá como \bar{A} puede ser expresado por conjuntos de pares que constan de elementos x de un conjunto universal X y un número de grados membresía $\mu_{\bar{A}}$, tal y como se expresa en la siguiente ecuación.

$$\bar{A} = \{(x, \mu_{\bar{A}}(x)) \mid x \in X, \mu_{\bar{A}}(x) \in [0,1]\} \quad \text{Ec. 4}$$

Para el caso particular de conjuntos difusos ya que los conjuntos son contables, los elementos de un conjunto pueden ser listados, y los elementos con grado de membresía cero, son usualmente omitidos. Con los conjuntos difusos tanto como con conjuntos convencionales hay operaciones básicas las cuales se describen suponiendo que se definen el conjunto \bar{A} y \bar{B} discreto sobre un universo de discurso X .

Intersección o disyunción. La intersección de dos conjuntos basados de su función característica está dada por la ecuación (5) donde el resultado de la intersección de los conjuntos es el valor mínimo del grado de pertenencia que tenga el punto a evaluar para cada valor del conjunto.

$$\bar{A} \cap \bar{B} \rightarrow \mu_{\bar{A} \cap \bar{B}}(x) = \min(\mu_{\bar{A}}(x), \mu_{\bar{B}}(x)) \quad \text{Ec. 5}$$

Unión o conjunción. La unión de conjuntos basados en su función característica está dada en la ecuación (6), donde el resultado de la conjunción es el valor máximo del grado de pertenencia que tenga el punto a evaluar en los conjuntos.

$$\bar{A} \cup \bar{B} \rightarrow \mu_{\bar{A} \cup \bar{B}}(x) = \max(\mu_{\bar{A}}(x), \mu_{\bar{B}}(x)) \quad \text{Ec. 6}$$

Complemento. El complemento de un conjunto difuso \bar{A} sobre el universo de discurso X , se expresa en la ecuación (7) donde el resultado de aplicar este conjunto es un el valor que no queda dentro del conjunto pero si en el universo.

$$\bar{\bar{A}} \rightarrow \mu_{\bar{\bar{A}}}(x) = 1 - \mu_{\bar{A}}(x) \quad \text{Ec. 7}$$

Estas operaciones básicas son fundamentales en la implementación y el funcionamiento de los conjuntos difusos. Junto a estas operaciones se debe mencionar a las funciones de membresía.

Una función de membresía de un conjunto difuso está hecha para simular el pensamiento de una persona cuantificándolo en una variable lingüística, es decir, representa el grado de pertenencia de un elemento x a un subconjunto previamente definido o previamente nombrado. Es importante hacer notar que los

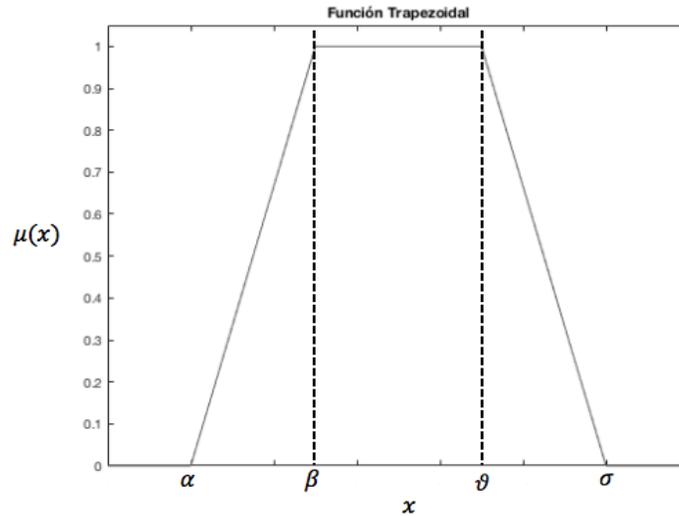
valores de las funciones de membresía o los grados de pertenencia no son probabilidades y no suman necesariamente 1, son formas de pensar de expertos ejemplificadas en estas funciones. Existen diferentes representaciones de dichas funciones y pueden ser elegidas arbitrariamente dependiendo de la experiencia del diseñador (Wang, 1997). Las más comunes son la trapezoidal, la triangular y la gaussiana.

2.1.1.1.1 Función de membresía trapezoidal

Esta función es una generalización de la función triangular. En este caso para la función de membresía se tiene no solo un valor para el cual la pertenencia es unitaria si no todo un conjunto de valores o franja que varia el ancho dependiendo del diseño de la función. La forma de la función se emplea cuando hay rango de valores óptimos.

Queda definida por la ecuación 8 que nos muestra como calcular el grado de pertenencia en algún punto de la función y puede observarse en la Fig 2.2.

$$\mu(x) = \begin{cases} 0 & \text{si } (x \leq \alpha) \text{ o } (x \geq \sigma) \\ \frac{x - \alpha}{\beta - \alpha} & \text{si } x \in (\alpha, \beta] \\ 1 & \text{si } x \in (\beta, \vartheta) \\ \frac{\sigma - x}{\sigma - \vartheta} & \text{si } x \in (\vartheta, \sigma) \end{cases} \quad \text{Ec. 8}$$

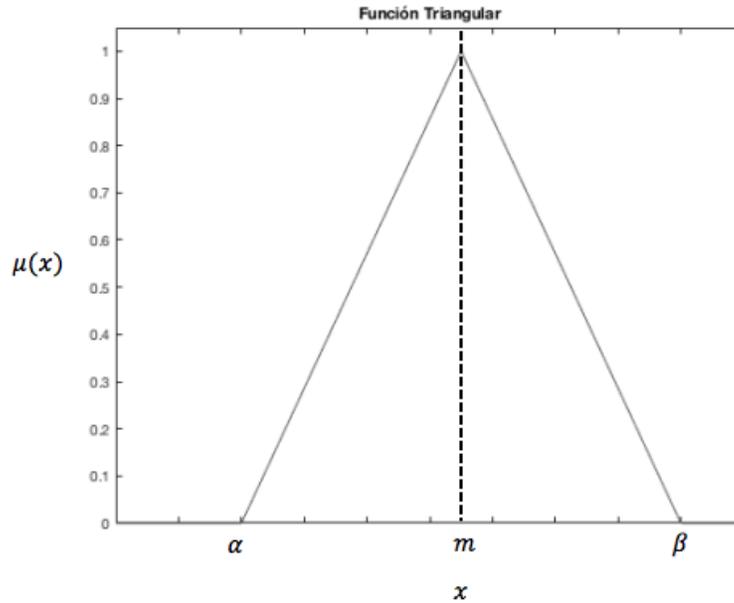


2.2 Función de membresía trapezoidal

2.1.1.2 Función de membresía triangular

Su forma como lo dice el nombre se constituye de una parte de pendiente positiva constante hasta que alcanza la unida y una vez lograda desciende de manera uniforme. El ejemplo de esta función está en la figura 2.3 y el cálculo del grado de pertenencia de un punto está dado por la ecuación 9.

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq \alpha \\ \frac{x - \alpha}{m - \alpha} & \text{si } x \in (\alpha, m] \\ \frac{\beta - x}{\beta - m} & \text{si } x \in (m, \beta) \\ 0 & \text{si } x \geq \beta \end{cases} \quad \text{Ec. 9}$$

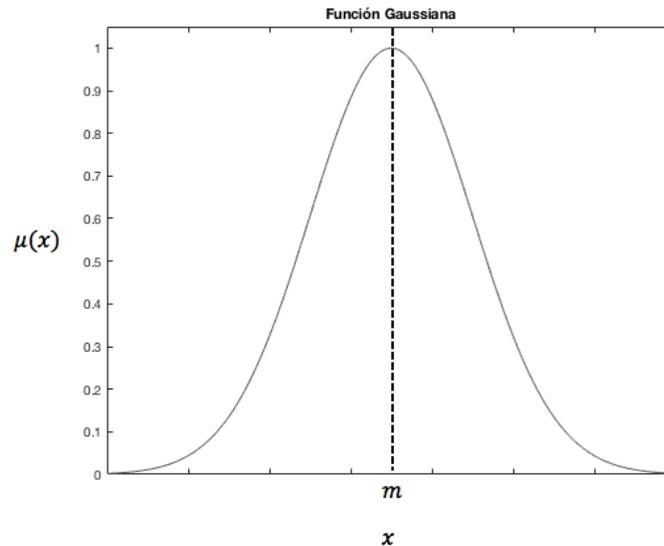


2.3 Función de membresía triangular

2.1.1.3 Función de membresía Gaussiana

Esta función se denomina así ya que es una campana Gaussiana. Es definida por su valor medio m y un valor $\sigma > 0$, representados en la ecuación 10. Entre más pequeño sea el valor de σ , más angosta será la campana, esto quiere decir que σ , controla el ancho de la campana. Un ejemplo de la función implementada sería la figura 9 donde se observa el centro y la variación del parámetro.

$$\mu(x) = \exp\left[\frac{-(x - m)^2}{2\sigma^2}\right] \quad \text{Ec. 10}$$



2.4 Función de membresía Gaussiana

2.1.2 Base de conocimientos

La base de conocimientos tiene la información de las reglas y las metas del controlador. Esta base define las metas de control y el manejo que debe utilizar el sistema para llevar a cabo el control, empleando proposiciones.

La base de conocimientos se obtiene de la experiencia desarrollada de un experto o alguna persona que conozca el proceso tanto del punto de vista práctico como matemático. La base de conocimiento está compuesta de dos partes, la base de reglas difusas y la base de datos.

La base de reglas está dada por el conocimiento previo del sistema por lo que con base en éste se seleccionan las variables de entrada del sistema. La prioridad cuando se quiere controlar un sistema es que la salida de éste sea muy cercana a la salida que se espera, esto quiere decir que el error sea muy cercano o igual a cero por lo que el error es una variable que se utiliza la mayoría de las veces.

Las variables se organizan en reglas de control tipo IF-THEN que son reglas compuestas por variables lingüísticas basadas en la experiencia que se tiene del proceso. Por lo general en cualquier sistema práctico se utiliza un

número de reglas para realizar una tarea, en este tipo de controlador se necesitan varias reglas que especifican acciones a realizar ante diferentes condiciones.

La base de datos parte de la definición de los universos de entrada y las funciones de pertenencia. Las funciones de pertenencia indican la relación que tiene una variable de entrada a cierto conjunto difuso. Estos conjuntos ya están definidos pueden de saturación, de hombro, triangulares, trapezoidales o sigmoidales.

2.1.3 Lógica de decisiones

La lógica de decisiones en el controlador difuso es la parte esencial del controlador ya que a partir de ésta se simula la importancia que una persona le daría a cada argumento de la base de reglas según la base de conocimientos.

La primera función de la etapa de la lógica de decisiones es determinar el grado de impacto de cada regla en la base de conocimientos. La segunda función de la etapa de inferencia es determinar el grado en que la regla se debe ponderar al llegar a la etapa final.

Estas etapas son agregación y composición. Siguiendo con el ejemplo de la temperatura del cuarto se combina ésta con la salida del clima, como la combinación de estas condiciones no es lógica convencional no se puede utilizar una función booleana para dar una respuesta, en vez de esto se utilizan conectores lógicos definidos como AND, OR, NOT expresados en la Tabla 1.

Tabla 1 Operadores difusos

Operadores	Valor Difuso
OR	$\mu_{A \cup B}(u) = \max\{\mu_A(u), \mu_B(u)\}$
AND	$\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}$
NOT	$\mu_{\bar{A}}(u) = 1 - \{\mu_A(u)\}$

2.1.4 Interfaz de desdifusificación

La interfaz de desdifusificación es la encargada de regresar los valores difusos del controlador a los rangos establecidos en el universo del sistema. Este proceso es el que transforma la salida difusa del controlador a una salida nítida o salida que pueda dar un valor definido con base en la unión de las salidas difusas para cada función de membresía.

La interfaz de desdifusificación utiliza funciones de membresía como las descritas en la sección 2.2.1.1. Los métodos que utiliza para encontrar el valor nítido son variados y entre ellos se mencionan los siguientes.

2.1.4.1 Método de centro de área o gravedad

La representación del método de área o gravedad de forma discreta se observa en la ecuación (11).

$$salida = \frac{\sum_{x=a}^b \mu(x) \cdot x}{\sum_{x=a}^b \mu(x)} \quad \text{Ec. 11}$$

Este método es sencillo ya que corta la función de membresía al grado de la membresía respectiva, esto quiere decir que segmenta las funciones de membresía generando en cada función dos áreas. El área inferior que se forma es la que se toma para hacer el cálculo. Se sobreponen todas estas áreas y se obtiene el centroide de la superposición, el cual indica la salida real del sistema.

2.1.4.2 Método de centro máximo.

Primero se calcula el valor más común de cada función de membresía: esto se hace con el máximo de la respectiva función de membresía. Posteriormente con estos valores típicos se les asigna un peso o valor, el cual es proporcional al grado de membresía y la salida se obtiene haciendo un balance entre los distintos valores. La ecuación es la misma solo que con los valores máximo y común de cada función de membresía.

Este método es bastante aplicado cuando se utilizan funciones de membresía trapezoidales en cuyo caso se emplea un valor típico de la mitad del máximo de la función.

2.1.4.3 Método de izquierda y derecha máximo

El método de máximo izquierda y máximo derecha es una variación del método máximo con funciones trapezoidales ya que en vez de utilizar el más común de la mitad del máximo se utiliza el máximo izquierdo o máximo derecho.

Debido a que es práctica común normalizar las entradas y salidas de los controladores difusos muchas veces se utilizan o se requieren ganancias de escalamiento que permiten que el controlador reciba las entradas en el rango del universo de discurso y proporcione las salidas correctas en el mismo universo.

2.1.5 Ganancias de escalamiento

Las ganancias de escalamiento son ganancias que se colocan al inicio y al final del sistema difuso y sirven para convertir las entradas al controlador difuso en parámetros adecuados al contexto lingüístico que el sistema pueda interpretar, las ganancias de salida del controlador permiten escalar el valor de salida y adecuarlo al universo del sistema de control. Estas ganancias también sirven para tener un ajuste fino del controlador.

La selección de valores adecuados de las ganancias de escalamiento se basa en el conocimiento del proceso que va a ser controlador y a veces a través de prueba y error para alcanzar el mejor rendimiento de controlador. Esto es debido a que no hay un método bien definido para un buen ajuste de las ganancias de escalamiento o factor de escalamiento.

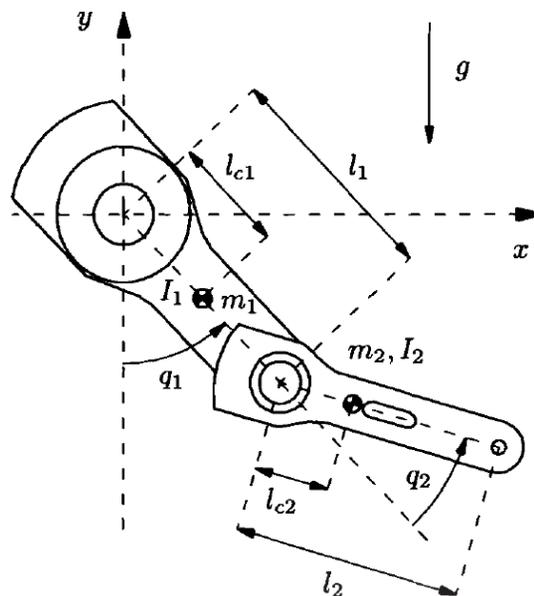
Las ganancias son los parámetros significativos del controlador difuso ya que cambiándolas se cambia el universo del discurso, esto es el dominio del sistema y las funciones de membresía de entrada o salida.

2.2 Sistema 2DOF (Two Degrees of Freedom)

Un sistema 2DOF (two degrees of freedom) se definirá, para este caso, como un mecanismo que consta de dos eslabones unidos mediante articulaciones de revoluta, el primer eslabón unido a una base y el segundo eslabón unido al primero. Este sistema es ampliamente conocido y estudiado por lo que el modelo dinámico es fácil de obtener.

2.2.1 Modelo dinámico de sistema 2DOF

El modelo dinámico de un sistema de 2DOF para un robot serial se obtiene de forma analítica, mediante las leyes de la física, específicamente como se habla de robots de las leyes de la mecánica. Este método de obtención del modelo está basado en ecuaciones diferenciales que describen el comportamiento y la relación de los diferentes componentes del sistema entre sí. Visto desde un enfoque dinámico un robot manipulador de n grados de libertad puede ser considerado como un sistema no lineal multi-variable, teniendo n entradas que se definen como los pares o fuerzas que son aplicados a las articulaciones mediante actuadores, y $2n$ salidas que son la posición q y la velocidad \dot{q} de estas articulaciones.



2.5 Robot de 2 DOF

Basados en el modelo dinámico del robot de 2DOF se obtiene las siguientes ecuaciones:

$$\begin{aligned} \tau_1 = & [m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2] \ddot{q}_1 + \\ & [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_2 - 2m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1 \dot{q}_2 - \\ & m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2^2 + [m_1 l_{c1} + m_2 l_1] \sin(q_1) + m_2 l_{c2} g \sin(q_1 + q_2) \end{aligned} \quad \text{Ec. 12}$$

$$\begin{aligned} \tau_2 = & [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_1 + [m_2 l_{c2}^2 + I_2] \ddot{q}_2 \\ & + m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1^2 + m_2 l_{c2} g \sin(q_1 + q_2) \end{aligned} \quad \text{Ec. 13}$$

Donde q_1 y q_2 se definen como las posiciones de los ángulos para las uniones $i = 1, 2$; \dot{q}_i es la velocidad de las uniones y \ddot{q}_i es la aceleración. El término l_i es la longitud de los eslabones, l_{ci} es la distancia del centro de masa de los eslabones, m_i es la masa de cada eslabón y por último I_i es la inercia relativa al centro de masa del eslabón.

La forma compacta del modelo es:

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} C_{11}(q, \dot{q}) & C_{12}(q, \dot{q}) \\ C_{21}(q, \dot{q}) & C_{22}(q, \dot{q}) \end{bmatrix} \dot{q} + \begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix} = \tau \quad \text{Ec. 14}$$

De donde:

$$M_{11}(q) = m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2 \quad \text{Ec. 15}$$

$$M_{12}(q) = m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2 \quad \text{Ec. 16}$$

$$M_{21}(q) = m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2 \quad \text{Ec. 17}$$

$$M_{22}(q) = m_2 l_{c2}^2 + I_2 \quad \text{Ec. 18}$$

$$C_{11}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2 \quad \text{Ec. 19}$$

$$C_{12}(q, \dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 + \dot{q}_2] \quad \text{Ec. 20}$$

$$C_{21}(q, \dot{q}) = m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1 \quad \text{Ec. 21}$$

$$C_{11}(q, \dot{q}) = 0 \quad \text{Ec. 22}$$

Respecto a las variables, el modelo dinámico se describe como:

$$\frac{d}{dt} \begin{bmatrix} q_n \\ \dot{q}_n \end{bmatrix} = \begin{bmatrix} \dot{q}_n \\ M(q)^{-1} [\tau(t) - C(q, \dot{q})\dot{q} - g(q)] \end{bmatrix} \quad \text{Ec. 23}$$

Conforme a la ecuación 14 se obtienen las aceleraciones del robot, estas aceleraciones se integran para obtener la velocidad mediante el método de Euler y se integran de nuevo mediante el mismo método para obtener la posición que será enviada al servomotor.

2.3 Controlador PD con pre-compensación

Un controlador permite que el sistema al cual controla alcance el punto al que se desee llegar llamado posición de control, esto quiere decir que el error entre el valor que se desea y el valor en el que está el sistema sea igual a cero o muy cercano a cero. Con los controladores PD es posible alcanzar la posición de control deseada en robots manipuladores. Para estos casos la sintonización de las ganancias K_p y K_v se deben seleccionar simétricas y definir las positivas, estas ganancias reciben el nombre de ganancias de posición y velocidad respectivamente.

El controlador PD con pre-compensación queda definido de acuerdo a la ley de control de la ecuación 24.

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + M(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + g(q_d) \quad \text{Ec. 24}$$

$$\begin{aligned}
M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \\
= K_p\tilde{q} + K_v\dot{\tilde{q}} + M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d)
\end{aligned}
\tag{Ec. 25}$$

La ecuación de malla cerrada del controlador puede expresarse en términos del vector de estado $[\tilde{q}^T \quad \dot{\tilde{q}}^T]^T$ como:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ M(q)^{-1}[-K_p\tilde{q} - K_v\dot{\tilde{q}} - C(q, \dot{q})\dot{\tilde{q}} - h(\tilde{q}, \dot{\tilde{q}})] \end{bmatrix}
\tag{Ec. 26}$$

Donde:

$$\begin{aligned}
h(\tilde{q}, \dot{\tilde{q}}) = [M(q_d) - M(q)]\ddot{q}_d + [C(q_d, \dot{q}_d) - C(q, \dot{q})]\dot{q}_d \\
+ g(q_d) - g(d)
\end{aligned}
\tag{Ec. 27}$$

es la llamada dinámica residual.

La selección de las ganancias de control no es trivial, se deben obtener mediante un proceso de sintonía que garantice llegar a la posición deseada o posición objetivo de forma local; dicho de otra forma para la trayectoria deseada q_d la sintonía garantiza que $\lim_{t \rightarrow \infty} \tilde{q}(t) = 0$, siempre que el error de posición inicial $\tilde{q}(0)$ y la velocidad inicial de error $\dot{\tilde{q}}(0)$ sean suficientemente pequeñas.

2.3.1 Procedimiento de sintonía de controlador PD

El procedimiento de sintonía del controlador PD con pre-compensación determina los valores propios mínimos de las matrices de posición y velocidad K_p y K_v simétricas de diseño que garantizan el cumplimiento del objetivo de control de movimiento. El procedimiento de sintonía se resume como:

- Obtención del modelo dinámico del robot a ser controlador. Particularmente, determinar $M(q)$, $C(q, \dot{q})$ y $g(q)$.
- Cálculo de las constantes $\lambda_{Max}\{M(q)\}$, $\lambda_{min}\{M(q)\}$, $k_M, k'_M, k_{C1}, k_{C2}, k', k_g^2$.

- Determinar los valores $\|\dot{q}_d\|_{Max}$ y $\|\ddot{q}_d\|_{Max}$ a partir de la especificación de la tarea encomendada al robot.
- Cálculo de las constantes s_1 y s_2 dadas respectivamente por las ecuaciones:

$$s_1 = [k_g + k_M \|\ddot{q}_d\|_{Max} + k_{C2} \|\dot{q}_d\|_{Max}^2] \quad \text{Ec. 28}$$

$$s_2 = 2[k' + k'_M \|\ddot{q}_d\|_{Max} + k_{C1} \|\dot{q}_d\|_{Max}^2] \quad \text{Ec. 29}$$

- Obtener los valores de k_{h1} y k_{h2} dados por las ecuaciones:

$$k_{h1} \geq k_{C1} \|\dot{q}_d\|_{Max} \quad \text{Ec. 30}$$

$$k_{h2} \geq \frac{s_2}{\tanh\left(\frac{s_2}{s_1}\right)} \quad \text{Ec. 31}$$

- Cálculo de las constantes a y b:

$$a = \frac{1}{2} [\lambda_{Max}\{K_v\} + k_{C1} \|\dot{q}_d\|_{Max} + k_{h1}] \quad \text{Ec. 32}$$

$$b = \alpha_4 \lambda_{Max}\{M\} + \alpha_2 k_{C1} \quad \text{Ec. 33}$$

$$\text{con } \alpha_2 = \sqrt{n} \text{ y } \alpha_4 = 1.$$

- Seleccionar $\gamma > 0$ y determinar las matrices de diseño K_p y K_v de manera que sus valores propios mínimos satisfagan las ecuaciones:

$$\lambda_{min}\{K_v\} > k_{h1} + \gamma b \quad \text{Ec. 34}$$

$$\lambda_{\min}\{K_p\} > \alpha_3 \left[\frac{[2\gamma a + k_{h2}]^2}{4\gamma[\lambda_{\min}\{K_v\} - k_{h1} - \gamma b]} + k_{h2} \right] \quad \text{Ec. 35}$$

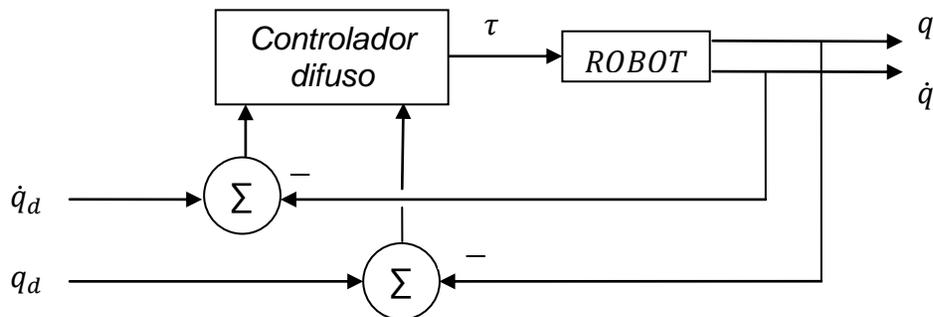
$$\lambda_{\min}\{K_p\} > \gamma^2 \left[\frac{\alpha_1^2 \lambda_{\max}^2\{M\}}{\lambda_{\min}\{M\}} \right] \quad \text{Ec. 36}$$

con $\alpha_1 = 1$, $\alpha_3 = 1$.

2.4 Controlador Difuso

Un controlador difuso, como su nombre lo indica, es un controlador basado en lógica difusa, esto quiere decir que no necesita conocer completamente el modelo matemático del sistema si no cómo se comporta realmente y a partir de este comportamiento se construye el controlador.

Es necesario que el controlador difuso sea desarrollado por o con apoyo de una persona experta en el sistema que se quiera controlar, esto significa que pueda decir cómo se comporta el sistema ante determinadas entradas; esto hace que el controlador sea confiable y garantice el cumplimiento del objetivo de control. Para un sistema robótico el controlador puede ser descrito de acuerdo a la figura siguiente.



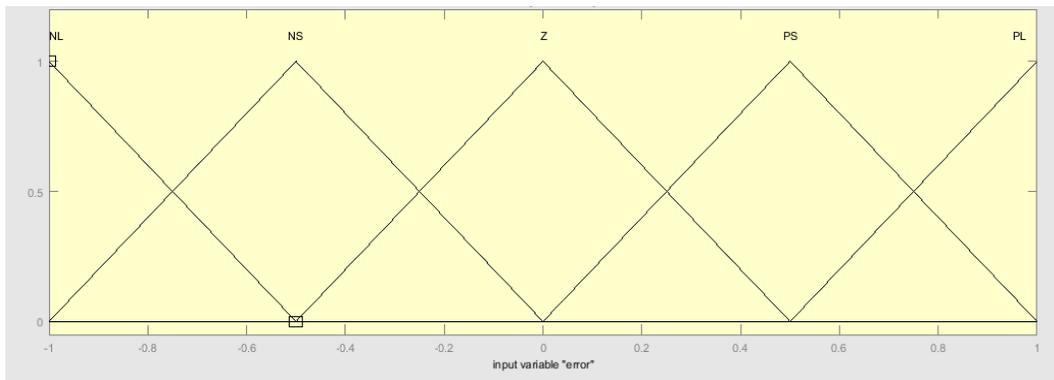
2.6 Controlador difuso

Los controladores propuestos generales basan su control en entradas definidas mediante el error de posición y el error de velocidad representados como

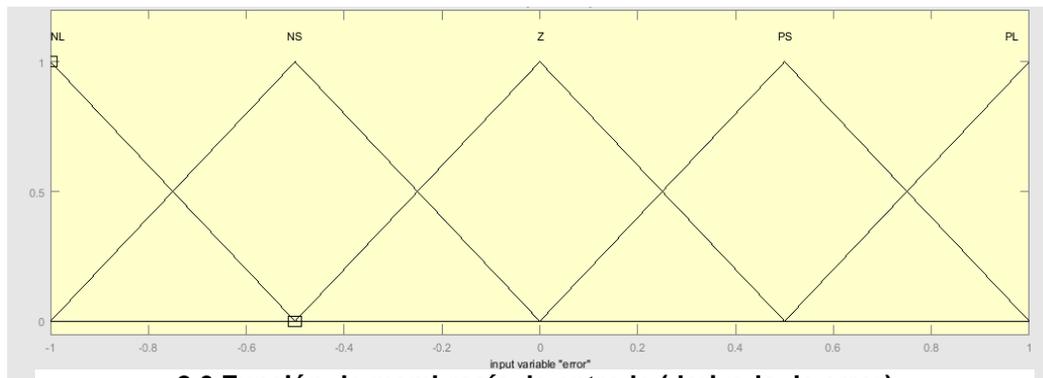
$$\tilde{q} = q_d - q \text{ y } \dot{\tilde{q}} = \dot{q}_d - \dot{q}.$$

2.4.1 Funciones de membresía

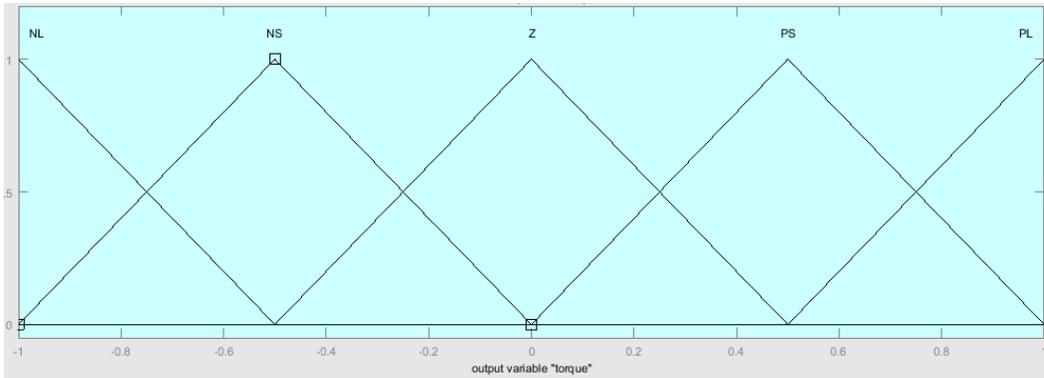
La estructura del controlador difuso consta de lo mostrado en la fig 2.3. Ejemplificado en términos de funciones de membresía triangulares tanto para la entrada como para salida, se tiene:



2.7 Función de membresía de entrada (error)



2.6 Función de membresía de entrada (derivada de error)



2.8 Función de membresía de salida (torque)

2.4.2 Reglas del controlador

La base de conocimientos se muestra en la tabla 2, esta tabla es una forma compacta que permite verificar que todas las condiciones tengan una consecuencia, dicho de otra forma que todas las combinaciones de entradas tengan una salida.

Tabla 2 Reglas controlador difuso

Torque		Error				
		NB	NS	Z	PS	PB
Derivada de error	NB	NB	NB	NS	NS	Z
	NS	NB	NS	NS	Z	PS
	Z	NS	NS	Z	PS	PS
	PS	NS	Z	PS	PS	PB
	PB	Z	PS	PS	PB	PB

Estas reglas también pueden ser escritas en formato IF-THEN, como se muestra en la tabla 3. Si el error es NS y la derivada del error es PS entonces el torque o salida es Z, si el error es Z y la derivada del error es PB entonces el torque es PS y así para cada regla.

Tabla 3 Reglas lingüísticas

Si el error es:	Y el cambio de error es:	Entonces la salida es
NB	NB	NB
NB	NS	NB
NB	Z	NS
NB	PS	NS
NB	PB	Z
NS	NB	NB
NS	NS	NS
NS	Z	NS
NS	PS	Z
NS	PB	PS
Z	NB	NS
Z	NS	NS
Z	Z	Z
Z	PS	PS
Z	PB	PS
PS	NB	NS
PS	NS	Z
PS	Z	PS

PS	PS	PS
PS	PB	PB
PB	NB	Z
PB	NS	PS
PB	Z	PS
PB	PS	PB
PB	PB	PB

Cada sistema difuso requiere de su propia base de conocimientos ya que cada problema presenta una solución distinta, sin embargo en general las reglas de la tabla 3 son aplicables a controladores de posición o trayectoria por lo que estas reglas sirven como base para la construcción de sistemas difusos para este tipo de aplicaciones.

2.5 PSO

PSO son las siglas para Particle Swarm Optimization, un algoritmo de evolución que es utilizado para encontrar la solución óptima o más cercana a la óptima de un conjunto en un espacio de búsqueda. El PSO es un algoritmo útil para parámetros de optimización en espacios de búsqueda multi-dimensional y continuos.

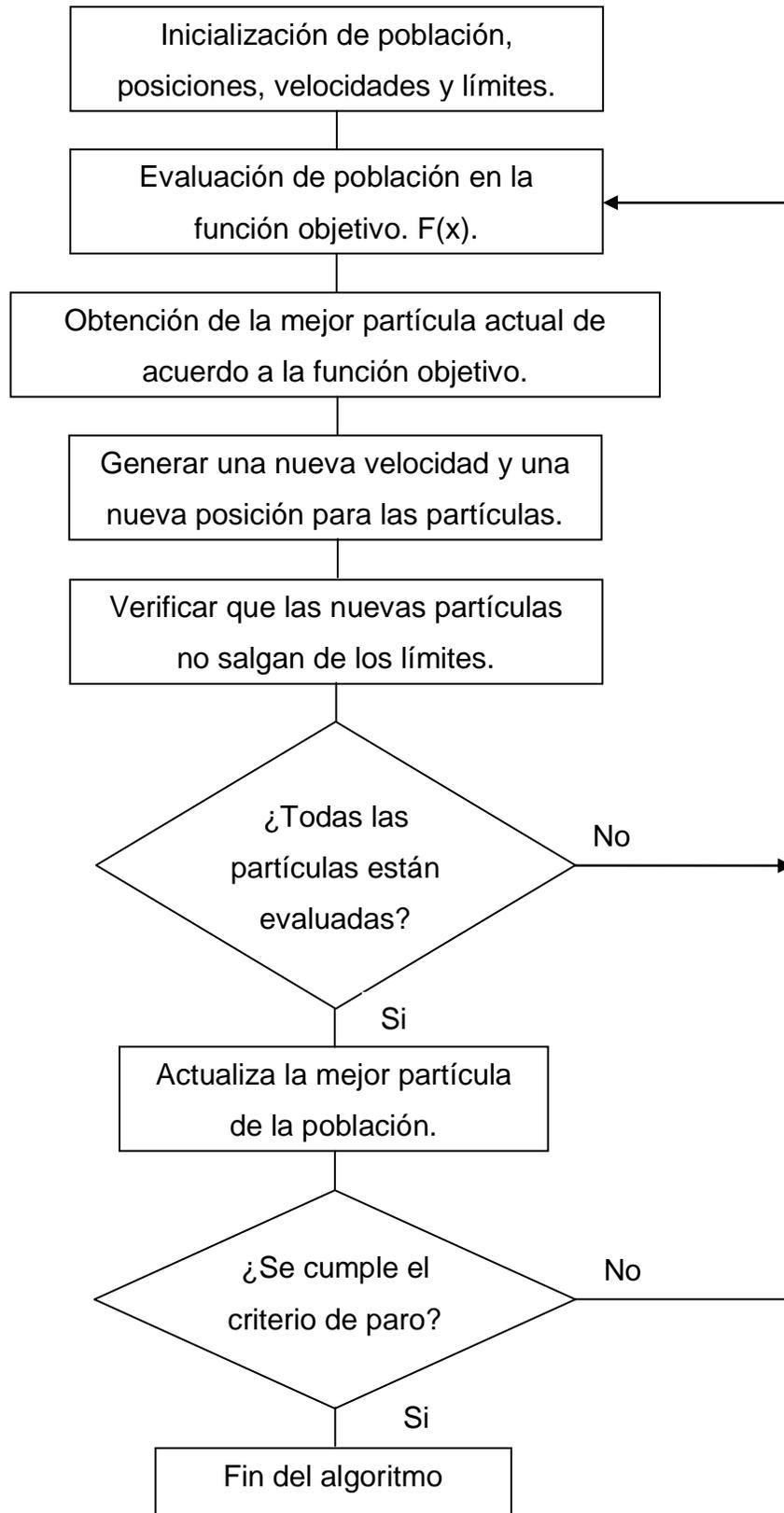
La organización de este algoritmo es la siguiente:

- Se define la dimensión del espacio de búsqueda y el tamaño de las partículas.
- Se asignan límites para cada elemento del espacio de búsqueda.
- Se evalúa la función objetivo para el tamaño de las partículas y se crea el óptimo local y global.

- Se define el número total de iteraciones o el valor máximo permitido para la optimización.
- Se obtienen velocidades y posiciones de las partículas.
- Se verifica que las nuevas posiciones no se salgan de los límites.
- Se evalúan las nuevas posiciones en la función objetivo.
- Si las posiciones son mejores se actualiza el mejor valor local.
- Se actualiza el mejor global.
- Se devuelve el mejor global al cumplirse el criterio de paro, este puede ser un número de iteraciones o un valor mínimo permitido para la función objetivo.

Estos pasos puede ser comprendidos visualmente mediante en el diagrama de flujo de la figura 2.8 donde se observa el algoritmo. La mejor partícula actual se define al evaluación la posición de la partícula en la función objetivo con una nueva partícula generada si al evaluarse en la función es mejor la partícula se actualiza. La mejor partícula de la población es la partícula que durante un ciclo se acerca más al criterio de paro al ser evaluada en la función objetivo. Esta partícula se toma como la óptima al cumplirse el criterio de paro del algoritmo ya que resulta la mejor a nivel tanto local como global.

Cada partícula se actualiza al modificar su velocidad y su posición en el espacio de búsqueda, esto se realiza mediante operaciones vectoriales simples de suma por lo que el proceso y la complejidad del algoritmo está en la evaluación del valor óptimo de cada partícula en cada iteración y que depende del problema a optimizar.



2.9 Algoritmo PSO

2.5.1 Actualización de velocidad

La nueva posición de cada una de las partículas en el espacio de búsqueda se obtiene a partir del cálculo de su velocidad. Cuando el sistema comienza las partículas parten del reposo por lo que su velocidad inicial es cero, conforme se desplazan generan una velocidad que se va actualizando en función de las mejores posiciones de cada partícula tanto locales como globales. La fórmula utilizada para el cálculo de la velocidad se encuentra en la ecuación 30.

$$v^{t+1} = w \times v^t + C_1 \text{random1} \times (P - X^t) + C_2 \text{random2} \times (G - X^t) \quad \text{Ec. 37}$$

Donde :

- v^{t+1} es el valor a calcular de la velocidad, la velocidad siguiente que se calcula para la iteración t+1.
- w es el peso inercial el cual tiene un valor entre 0.4 y 0.9.
- v^t es la velocidad actual, la velocidad t.
- C_1 es el componente de la atracción personal de la partícula, normalmente con un valor de 2.
- random 1 y 2 son números aleatorios que forman parte del rango de 0 a 1, distribuidos uniformemente.
- P es el vector que contiene las mejores posiciones actuales de cada partícula.
- X^t es el vector que contiene las posiciones actuales de cada partícula.
- C_2 es el componente de la atracción global de la partícula, normalmente con un valor de 2.
- G es el vector que contiene la mejor partícula a nivel global.

De la ecuación 30 podemos deducir que si la posición X^t es la mejor local, o sea, si X^t es P entonces la partícula no se mueve con respecto a su posición

local ya que ya está en la mejor posición posible y solamente se mueve respecto a la mejor posición global. Todas las partículas se mueven entonces siguiendo a la mejor partícula local y global dentro del algoritmo. Si X^t resulta ser la mejor posición global/local la partícula continúa su movimiento en la misma dirección y todas las demás partículas se mueven respecto a esta.

Los valores random 1 y 2 pretenden dar la posibilidad a la ecuación 30 el dirigirse en cualquier trayectoria siempre tomando en cuenta la dirección del óptimo local y global. El cálculo de velocidad es el tema más recurrente al alterar un PSO y es donde la mayoría de las modificaciones suceden con respecto a este algoritmo.

2.5.2 Actualización de posición

Una vez que la actualización de velocidad se ha realizado las partículas deben moverse a sus nuevas posiciones con esa velocidad. Para moverse deben sumarse los vectores de posición actual y la velocidad calculada para que la nueva posición quede definida. La ecuación 31 describe esta actualización de posición.

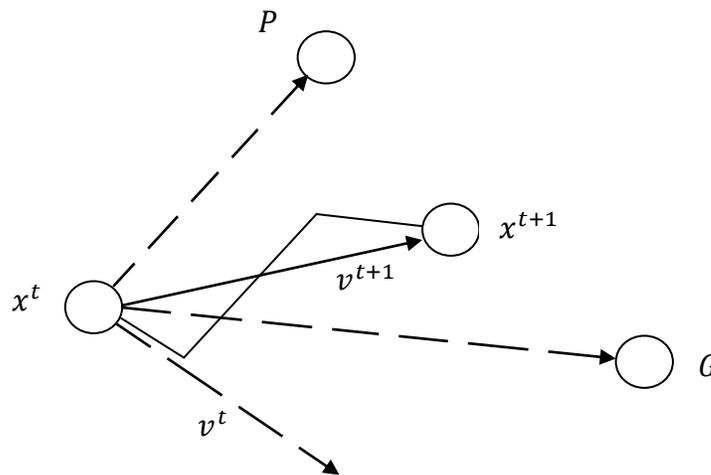
$$X^{t+1} = X^t + v^{t+1} \quad \text{Ec. 38}$$

Donde:

- X^{t+1} es el nuevo vector de posiciones de las partículas
- X^t es el vector de posiciones actuales del sistema
- v^{t+1} es la velocidad calculada anteriormente

La figura 2.10 describe el movimiento de una partícula; como puede observarse la posición actual x^t se desplaza siguiendo una velocidad v^t , para la nueva

velocidad se toman en cuenta los vectores v^t , P y G . por lo que v^{t+1} será la resultante de estos y la nueva posición es una adición de la posición actual y el vector de velocidad calculada.



2.10 Actualización de velocidad y posición

La posición a la que la partícula se acercó es mejor en comparación a la posición anterior ya que su desplazamiento está orientado a seguir la mejor partícula local y general del universo del problema.

3.DESARROLLO

Los temas mostrados en el marco teórico son aplicados en esta sección para llevar a cabo el proyecto. La metodología a seguir se compone de diferentes pasos todos enfocados a cumplir con los objetivos específicos del proyecto, esto permite cumplir con el objetivo general y a su vez corroborar la hipótesis planteada.

Es importante aclarar que debido a que en una etapa media del proyecto los recursos se limitaron y no fue diseñada ni construida la plataforma se optó por un sistema de dos grados de libertad que era parte del diseño de la plataforma y tiene un modelo conocido lo que permitía continuar con el objetivo principal de diseñar el controlador y compararlo. La metodología utilizada tiene como base el uso de un sistema tipo brazo de dos grados de libertad.

3.1 Metodología

La estructura del proyecto parte del modelo del sistema de dos grados de libertad visto en la figura 2.5. Una vez el modelo dinámico del sistema es obtenido se debe seleccionar un controlador para el seguimiento de la trayectoria por lo que se optó por el controlador PD con pre-compensación ya que se ha reportado por diferentes autores que este controlador ha demostrado excelentes prestaciones y es comparable incluso con el empleo de controladores del tipo Par-Calculado.

Una vez que se tiene el controlador se debe implementar en el entorno virtual por lo que se utilizó el programa Matlab para esta tarea. Para implementar el controlador PD pre-calculado se debe conocer la trayectoria a seguir ya que una de las matrices de este controlador es la matriz de valores deseados en posición velocidad y aceleración por lo que se propuso una trayectoria senoidal para ambos eslabones.

La trayectoria propuesta permite obtener las posiciones, velocidades y aceleraciones deseadas con el fin de obtener las matrices de inercia, de Coriolis y gravedad que son el pre-cálculo del controlador.

Teniendo estos datos se utiliza el procedimiento de sintonía propuesto en la sección 2.3.1 para obtener los valores mínimos de ganancias K_p y K_v que serán las que lleven al sistema por la trayectoria deseada. Una vez obtenidas las gráficas de respuesta del controlador a la trayectoria del robot de 2DOF se diseña el controlador difuso.

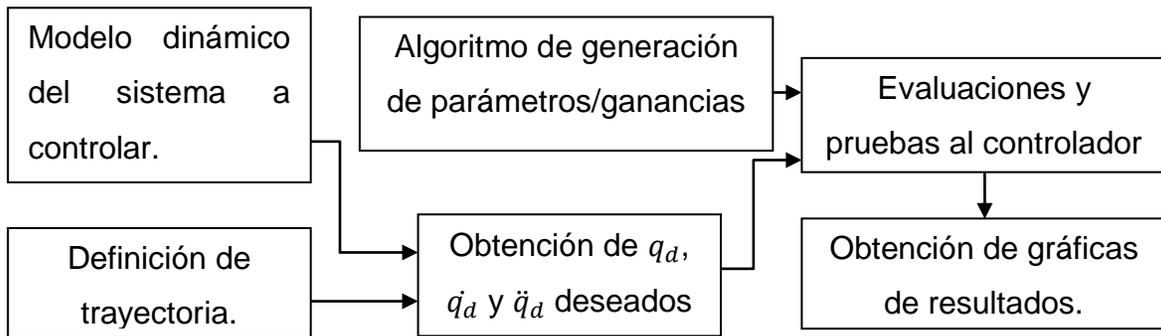
Para el sistema difuso se emplean funciones de membresía gaussianas ya que a diferencia de una función triangular los cambios de pertenencia a cierto conjunto no son tan drásticos y se acercan a más a la realidad. Las funciones de membresía se definen tanto de entrada como de salida y la base de reglas utilizadas para la relación es descrita en la Tabla 2. Esto nos lleva al diseño de dos controladores difusos, uno por cada eslabón del sistema de 2DOF y cada uno con parámetros de entrada de error de posición y error de velocidad, y la salida de cada controlador es el torque necesario para llevar el sistema a la trayectoria deseada.

La forma de implementar el controlador difuso es muy general y sigue una estructura similar a la propuesta con controladores de posición en base a las reglas donde si el error es muy grande y la derivada del error también el torque o salida debe ser también grande. El controlador tiene funciones de membresía gaussianas moderadamente abiertas y distribuidas uniformemente a través del rango del universo de entrada y salida por lo que para mejorar la respuesta se sugiere el uso de un algoritmo de optimización. El algoritmo de optimización utilizado es el PSO.

Para la implementación del PSO se propuso la modificación de los valores que establecen a la función de membresía, estos valores son el valor medio definido como m y la amplitud definida como σ de la función gaussiana vista en la

sección 2.1.1.3. Estos valores se modifican para todas las funciones de membresía tanto de entrada como de salida; el algoritmo PSO pretende generar los mejores puntos medios y amplitudes para cada función de membresía que permitan cumplir con el objetivo de control y seguir la trayectoria con un error menor al obtenido en el controlador PD con pre-compensación.

El esquema de la figura 3.1 muestra de forma muy general lo descrito anteriormente, esta metodología aplica tanto para el controlador PD con pre-compensación como para el controlador difuso.



3.1 Metodología propuesta

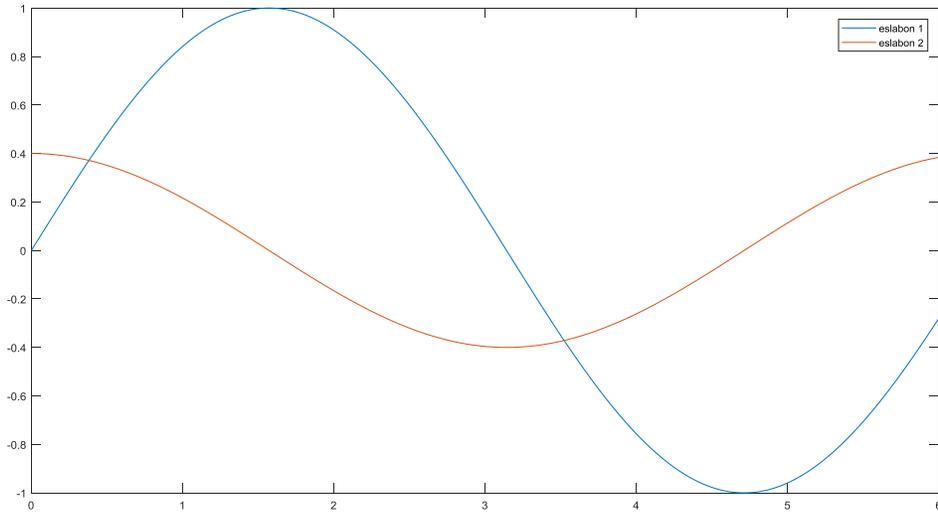
Cada parte de este esquema se verá con detalle a lo largo de esta sección.

3.2 Definición de trayectoria

Para evaluar las prestaciones de cada controlador se eligió la siguiente trayectoria:

$$q_d = \begin{bmatrix} q_{d1} \\ q_{d2} \end{bmatrix} = \begin{bmatrix} \text{sen}(t) \\ 0.4\text{cos}(t) \end{bmatrix} \quad \text{Ec. 39}$$

Las posiciones de la trayectoria para cada eslabón se almacenan en el vector q_d y la trayectoria de cada articulación puede verse en la figura 3.2.

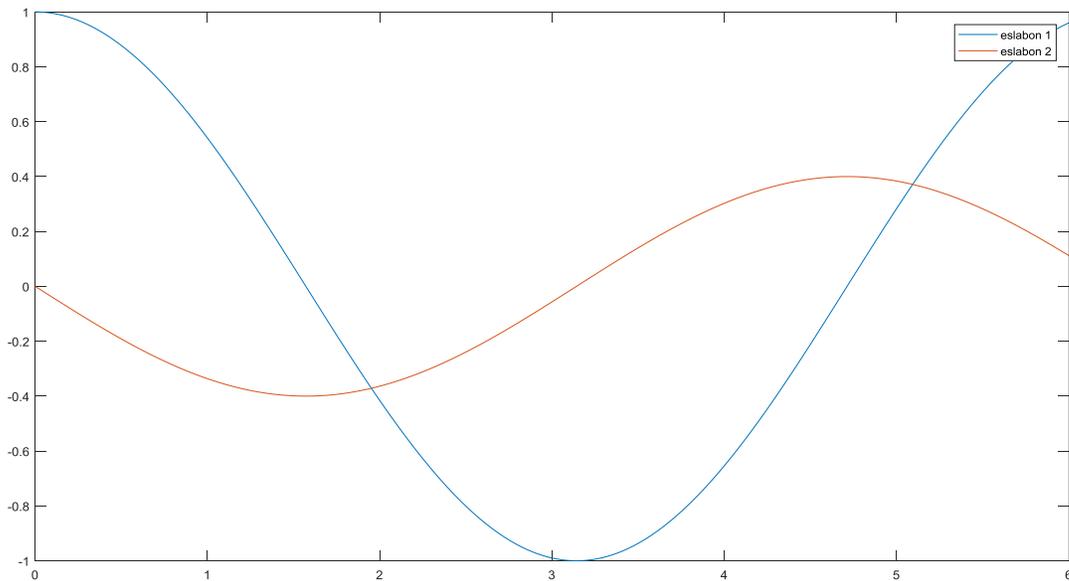


3.2 Posiciones deseadas de trayectoria a seguir

Derivando la ecuación 39 obtenemos la velocidad para cada eslabón, esta puede verse en la ecuación 40.

$$q_d = \begin{bmatrix} \dot{q}_{d1} \\ \dot{q}_{d2} \end{bmatrix} = \begin{bmatrix} \cos(t) \\ -0.4\text{sen}(t) \end{bmatrix} \quad \text{Ec. 40}$$

La figura 3.3 muestra las velocidades deseadas para las articulaciones.

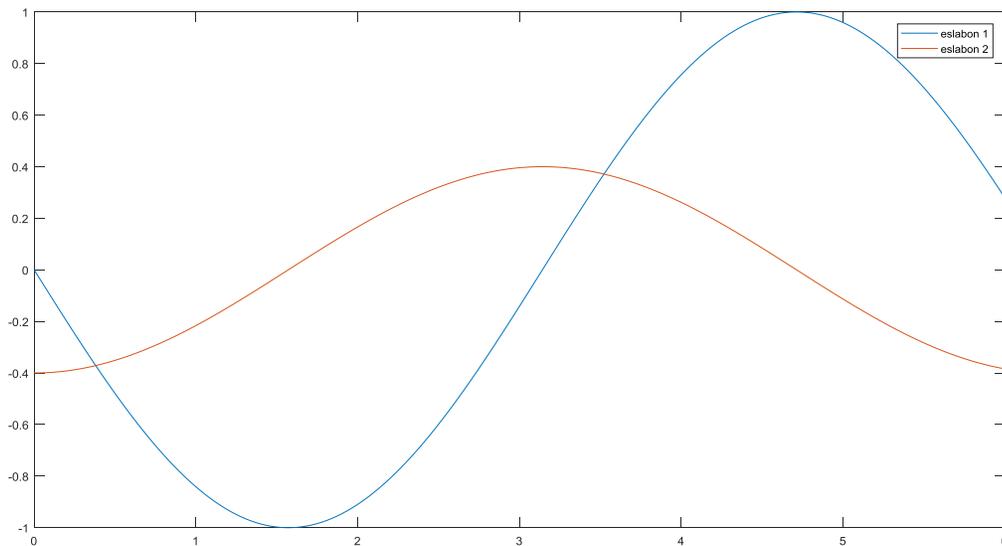


3.3 Velocidades deseadas de trayectoria a seguir

Derivando la ecuación 40 obtenemos la aceleración para cada eslabón, esta puede verse en la ecuación 41

$$q_d = \begin{bmatrix} \ddot{q}_{d1} \\ \ddot{q}_{d2} \end{bmatrix} = \begin{bmatrix} -\text{sen}(t) \\ -0.4\text{cos}(t) \end{bmatrix} \quad \text{Ec. 41}$$

La figura 3.4 muestra las aceleraciones deseadas para cada articulación.



3.4 Aceleraciones deseadas de trayectoria a seguir

Esta trayectoria tiene un término suave para el primer y segundo eslabón, este es un movimiento senoidal que evalúa la respuesta ante señales periódicas y presenta un término que crece suavemente para mantener al robot en estado de operación sin saturar el sistema.

3.3 Modelo dinámico del sistema

Los parámetros utilizados para obtener el modelo dinámico del sistema de 2DOF se muestran en la tabla 3.

Tabla 4 Parámetros de robot de 2DOF

Parámetros	Valores	Unidades
l_1	0.093	Mts
l_1	0.073	Mts
l_{c1}	0.0326	Mts
l_{c2}	0.0126	Mts
m_1	0.065	Mts
m_2	0.055	Mts
I_1	0.0113	Kgm ²
I_2	0.01116	Kgm ²
g	9.810	$\frac{m}{s^2}$

Los valores fueron obtenidos a partir del modelo físico del sistema, l corresponde a la longitud del eslabón, l_c es la longitud al centro de masa del eslabón, m es la masa del eslabón, I es la inercia también del eslabón y g es el término de la gravedad.

Estos valores son los que se utilizan para resolver todas las ecuaciones que se desarrollen u obtengan a lo largo del proyecto.

El modelo dinámico esta descrito en la ecuación 23 y los parámetros de la tabla 3 se colocan en las ecuaciones 15-22 para obtener la matriz de inercia, la matriz centrífuga y de Coriolis y la matriz de gravedad del sistema. Las matrices quedan en función de los ángulos que tome el eslabón en el tiempo.

3.4 Matrices de posición, velocidad y aceleración deseadas

El cálculo de la trayectoria permite determinar los valores deseados q_d , \dot{q}_d y \ddot{q}_d que son utilizados como parte del pre-cálculo del controlador. Estos

valores permiten obtener las matrices de inercia, de Coriolis y las matrices de gravedad deseadas y definidas como:

$$M(q_d)\ddot{q}_d \quad \text{Ec. 42}$$

$$C(q_d, \dot{q}_d)\dot{q}_d \quad \text{Ec. 43}$$

$$g(q_d) \quad \text{Ec. 44}$$

Estas matrices junto con los valores reales obtenidos para cada posición del sistema conforma el sistema en lazo cerrado descrito en la ecuación 26. En esta ecuación puede observarse que el componente de la dinámica residual es quien utiliza las matrices 42-44. La ecuación 26 da una respuesta en error de aceleración que se integra mediante el método de Euler para obtener error de posición. Este error es igual a $\tilde{q} = q_d - q$ donde la variable a despejar es la posición q y será también la variable de entrada del sistema para completar el lazo cerrado.

Establecidas las matrices y de donde se obtienen los valores de realimentación al sistema el diseño del controlador es el siguiente paso en el desarrollo del proyecto.

3.5 Diseño del controlador

La parte de diseño del controlador abarca la selección de las ganancias K_p y K_v que permiten que el error al seguir la trayectoria sea muy bajo o cero. Para este diseño primero se optó por seguir el procedimiento definido para la sintonía de los valores propios de estas ganancias. Este resultado ofrece un punto de partida para dichas ganancias ya que el método de sintonía asegura estabilidad. Una vez que se tiene un parámetro base que asegura estabilidad se utiliza un método de optimización para llevar las ganancias a sus valores óptimos tanto para el controlador PD con pre-compensación.

Esta sub sección abarca estos métodos de optimización, todos los controladores tienen un tiempo de muestreo de 0.01s y un tiempo total de ejecución de 5 segundos. El sistema se programa en Matlab y todos los comandos que se utilicen en el documento son aplicables al software.

3.5.1 Diseño mediante procedimiento de sintonía.

Como de menciona en la sección 2.3.1 existe un procedimiento definido para la sintonía de los valores propios mínimos de las matrices de posición K_p y la matriz de velocidad K_v . Los resultados obtenidos al aplicar los valores de la tabla 3 para las variables son:

$$\lambda_{Max}\{M(q)\} = 0.0133 \quad \text{Ec. 45}$$

$$\lambda_{min}\{M(q)\} = 0.0010$$

$$k_m = 5.1559e^{-04}$$

$$k'_m = 0.0133$$

$$k_{c1} = 2.5780e^{-04}$$

$$k_{c2} = 5.1559e^{-04}$$

$$k_g = 0.0242$$

Para los valores deseados de velocidad y aceleración tiene:

$$\|\dot{q}_d\|_{Max} = \begin{bmatrix} 1 \\ 7.6358 \end{bmatrix} \quad \text{Ec. 46}$$

$$\|\ddot{q}_d\|_{Max} = \begin{bmatrix} 1.07 \\ 35.2183 \end{bmatrix}$$

El cálculo de las constantes s_1 y s_1 queda definido como:

$$s_1 = 0.2037 \quad \text{Ec. 47}$$

$$s_2 = 0.9912$$

Los valores de de k_{h1} y k_{h1} son:

$$k_{h1} = 0.002 \quad \text{Ec. 48}$$

$$k_{h2} = 0.9913$$

Las constantes a y b, definidas como:

$$a = 0.0166 \quad \text{Ec. 49}$$

$$b = 0.0136$$

Con estos valores las matrices quedan definidas como.

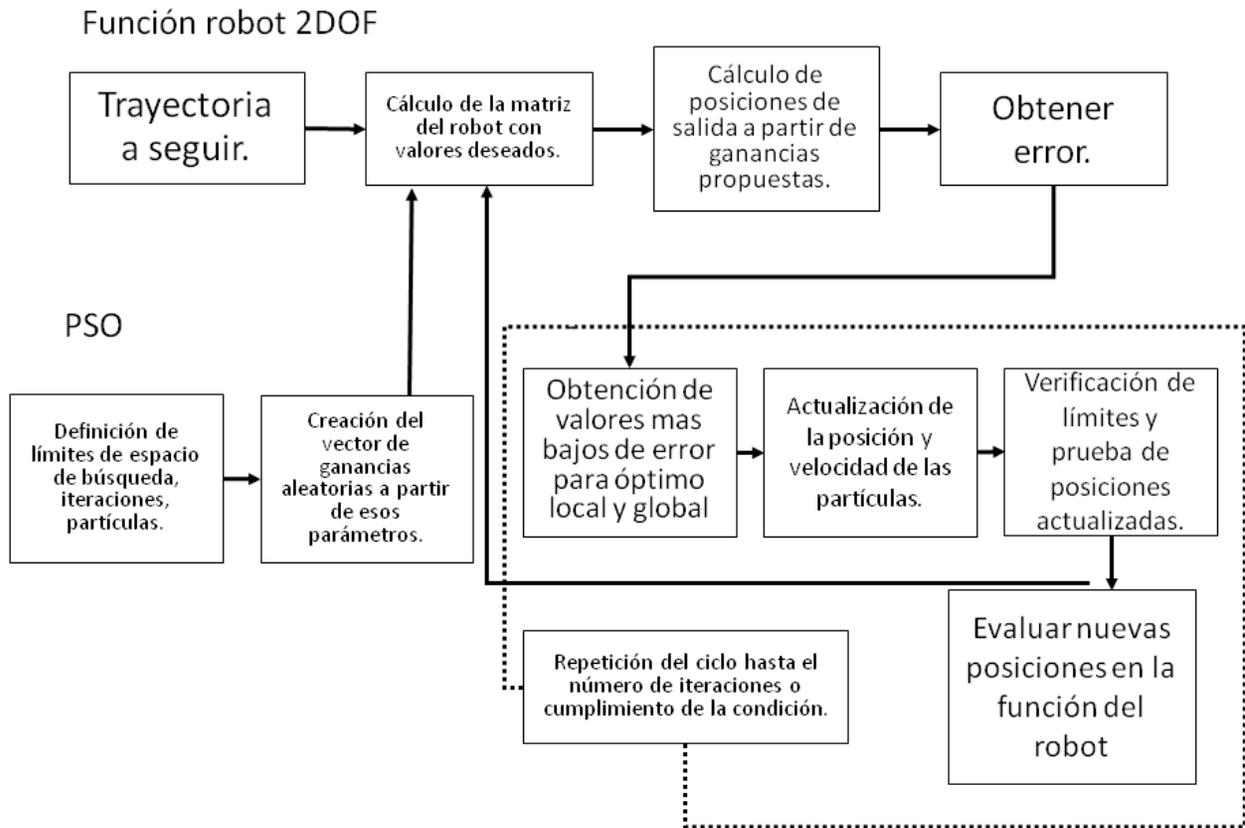
$$Kp = \begin{bmatrix} 5.051 & 0 \\ 0 & 3.9751 \end{bmatrix} \quad \text{Ec. 50}$$

$$Kv = \begin{bmatrix} 0.0621 & 0 \\ 0 & 0.0292 \end{bmatrix} \quad \text{Ec. 51}$$

Este método de sintonización pretende establecer un parámetro inicial para las ganancias del controlador que asegure estabilidad asintótica y equilibrio único del sistema. Debido a esto se descarta la utilización de estas ganancias como parámetros finales ya si bien aseguran estabilidad no aseguran un tiempo de respuesta óptimo y la sintonización a partir de aquí es prueba tras prueba. Para esto se pretende utilizar el algoritmo PSO ya que mediante este se pretende optimizar la respuesta del sistema basándose en los parámetros iniciales de estabilidad.

3.5.2 Diseño de controlador PD con pre-compensación mediante PSO

La estructura del controlador PD con pre-compensación puede observarse en la figura 2.4 sin embargo un esquema general que incluye el algoritmo implementado con la sintonización mediante el PSO se puede observar en la figura 3.5.



3.5 Estructura de sintonización de controlador PD con pre-compensación

Como puede observarse el sistema esta dividido en funciones donde el programa principal es el que tiene al algoritmo PSO como base.

Se empieza definiendo la dimensión del espacio de búsqueda que en este caso es cuatro ya que son los parámetros a optimizar. También se definen los límites que pueden tener los valores de la función, esto quiere decir los mínimos y máximos que los valores de las ganancias pueden tener ya que es el parámetro

con el que se optimiza. Estos quedan definidos de acuerdo a 50 y 51 para su menor valor, para el valor mayor se tomo un valor acorde a los límites del propio sistema. Estos valores son nombrados l y u definidos como *lower* y *ur bound* del sistema quedando definidos como:

$$l = [5 \quad 4 \quad 0.06 \quad 0.03] \quad \text{Ec. 52}$$

$$u = [8 \quad 8 \quad 0.8 \quad 0.5] \quad \text{Ec. 53}$$

Siguiendo el diseño de la figura 3.5 para el número de iteraciones en el PSO se utilizaron 8 y las partículas fueron 25. Una vez inicializada la población mediante la ecuación 54 se procede a obtener una primera función objetivo analizando las 50 partículas que se crearon. Esto lleva al sistema a tener un óptimo global que es la partícula que mejor responde en la primera iteración.

$$x = l + rand(Part_N,d).*(u-l) \quad \text{Ec. 54}$$

Después de obtener el óptimo global y local se mueven las partículas de acuerdo a la ecuación 47 y 48 y se verifica que no salgan de los límites l y u .

$$v = v + rand(Part_N,d).*(Loc_opt-x) + rand(Part_N,d).*(G_opt-x) \quad \text{Ec. 55}$$

$$x = x + v \quad \text{Ec. 56}$$

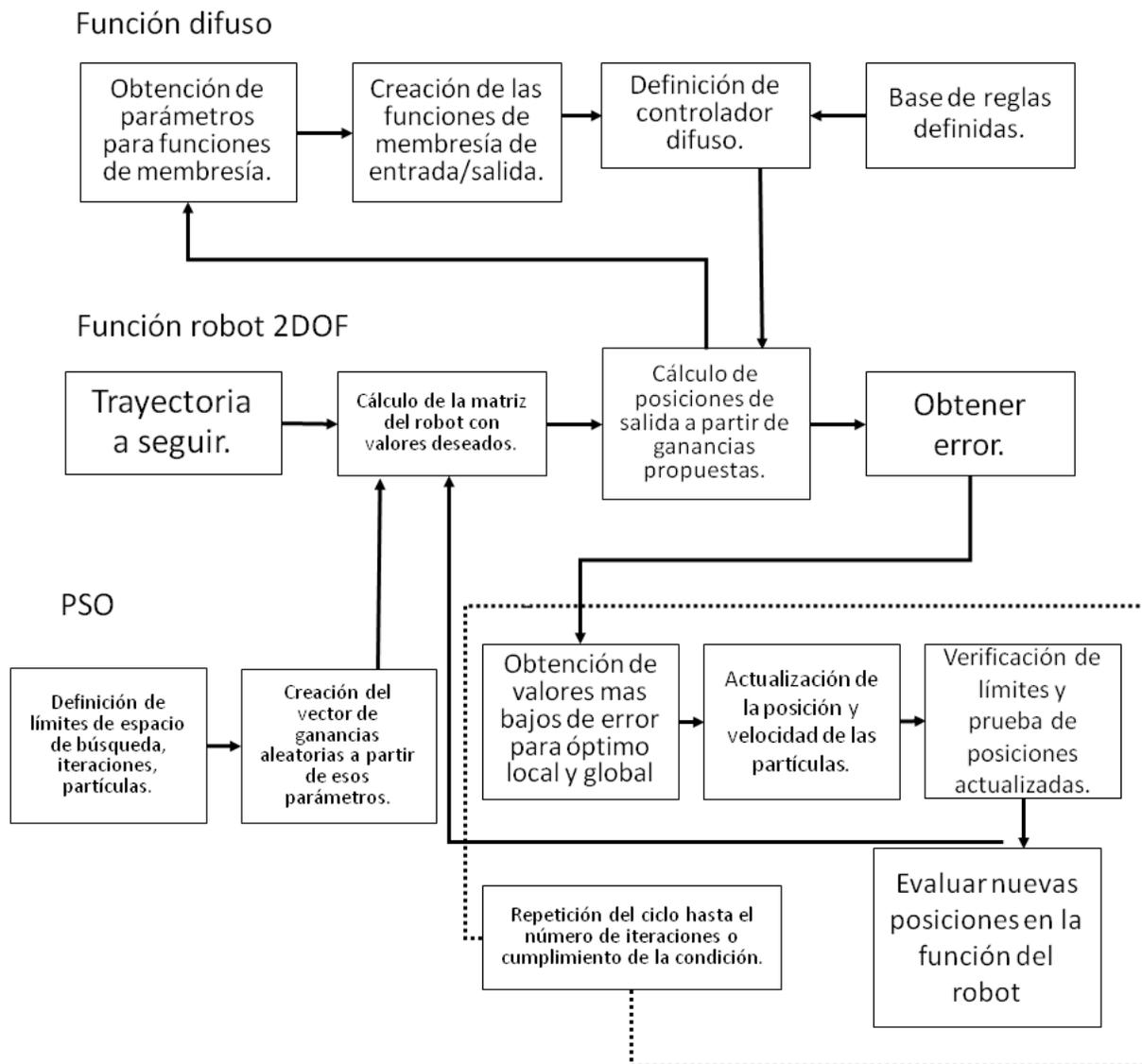
Estas nuevas posiciones se analizan en la función del controlador PD con pre-compensación y se obtiene un nuevo valor que es comparado con el valor de la función objetivo anterior, si el valor nuevo es menor al anterior se sustituye en la función objetivo y se vuelve el óptimo local. Este procedimiento se aplica para cada partícula en la función objetivo, hasta terminarlas donde se encuentra la mejor partícula global y se compara con la global anterior, si es mejor se sustituye en caso contrario se queda. El sistema repite la operación hasta que el número total de iteraciones se haya cumplido.

El valor que retorna la función del robot es el MRSE (mean of root of squared error) y se obtiene de la ecuación 57. Como puede observarla el MRSE es el error medio del sistema basado en el error del eslabón 1 y el error del eslabón 2. Conforme el algoritmo avanza el error debe ir reduciéndose ya que al incrementar de iteración tiene el mejor valor de las partículas analizadas anteriormente y comparadas con las actuales, esto lleva, por la naturaleza del algoritmo a un MRSE menor.

$$MRSE = E(k) = \frac{1}{N} \sum_1^N \sqrt{e_1^2(i) + e_2^2(i)} \quad \text{Ec. 57}$$

3.5.3 Diseño de controlador difuso mediante PSO

La estructura del controlador difuso puede observarse en la figura 2.6 sin embargo un esquema general que incluye el algoritmo implementado con la sintonización mediante el PSO se puede observar en la figura 3.6.



3.6 Estructura de sintonización de controlador PD con pre-compensación

Tiene la misma estructura que el controlador PD con pre-compensación con el añadido del ajuste de las funciones de membresía para evaluar la entrada de la trayectoria.

Se empieza definiendo la dimensión del espacio de búsqueda, para el control PD con pre-compensación este número eran las ganancias a optimizar, para este sistema los parámetros que se modifican son la amplitud y el centro de

las funciones de membresía tanto de entrada como de salida. Las funciones de membresía a utilizar son gaussianas, sección 2.1.1.3, por lo que el total de dimensiones contando funciones de entrada y salida es de 84.

Se debe definir el universo de entrada y salida de las funciones de membresía. De acuerdo con (Ross, 2010) este universo está definido por los parámetros máximos y mínimos que puede tomar el valor de la señal de entrada y salida. En caso de los eslabones el error máximo que pueden tener va de $[-1 \ 1]$ y el valor máximo de la derivada del error es $[-40 \ 40]$, para la salida tenemos $[-17 \ 17]$.

También se definen los límites que pueden tener los valores de la función, esto quiere decir, los mínimos y máximos que las funciones de membresía pueden modificarse dentro del universo del discurso. Estos valores son nombrados l y u definidos como *lower bound* y *upper bound* del sistema quedando definidos como:

$$L = [-1.3 \ -1.1 \ -0.9 \ -0.6 \ 0.04 \ -0.6 \ 0.04 \ -0.29 \ 0.04 \ 0.3 \ 0.5 \ 0.7 \ 1.0 \ 1.2]; \quad \text{Ec. 57}$$

$$u = [-1.2 \ -1.0 \ -0.7 \ -0.5 \ 0.80 \ -0.3 \ 0.80 \ 0.29 \ 0.80 \ 0.6 \ 0.6 \ 0.9 \ 1.1 \ 1.3]; \quad \text{Ec. 58}$$

Es importante señalar que estos valores están definidos en el rango de $[-1 \ 1]$ y aplican para todas las funciones de membresía con la diferencia de un parámetro considerado modificador que asegura las funciones en rangos correspondientes a sus entradas o salidas. En estos casos los modificadores de acuerdo al universo del discurso serían 1, 40 y 17 para el error, la derivada del error y el torque.

Siguiendo el algoritmo para el número de iteraciones en el PSO se utilizaron 25 y las partículas fueron 10. Una vez inicializada la población mediante la ecuación 59 se procede a obtener una primera función objetivo y un óptimo local analizando las partículas que se crearon. Esto lleva al sistema a localizar un

óptimo global de entre todos los locales, esta es la partícula que mejor responde de todas las 50 partículas en la primera iteración.

$$x = l + rand(Part_N,d).*(u-l) \quad \text{Ec. 59}$$

Después de obtener el óptimo global y local se mueven las partículas de acuerdo a la ecuación 60 y 61 y se verifica que no salgan de los límites l y u mencionados anteriormente.

$$v = w*v + C1*rand(Part_N,d).*(Loc_opt-x) + C2*rand(Part_N,d).*(G_opt-x) \quad \text{Ec. 60}$$

$$x = x + v \quad \text{Ec. 61}$$

Estas nuevas posiciones se analizan en la función del controlador difuso y se obtiene un nuevo valor de partícula de la función objetivo que es comparado con el valor de la partícula de la función objetivo anterior, si el valor nuevo es menor al anterior se sustituye en la función objetivo y se vuelve el óptimo local. Este procedimiento se aplica para cada partícula en la función objetivo, hasta terminarlas todas después se encuentra la mejor partícula global y se compara con la global anterior, si es mejor se sustituye en caso contrario se queda la anterior. El sistema repite la operación hasta que el número total de iteraciones se haya cumplido.

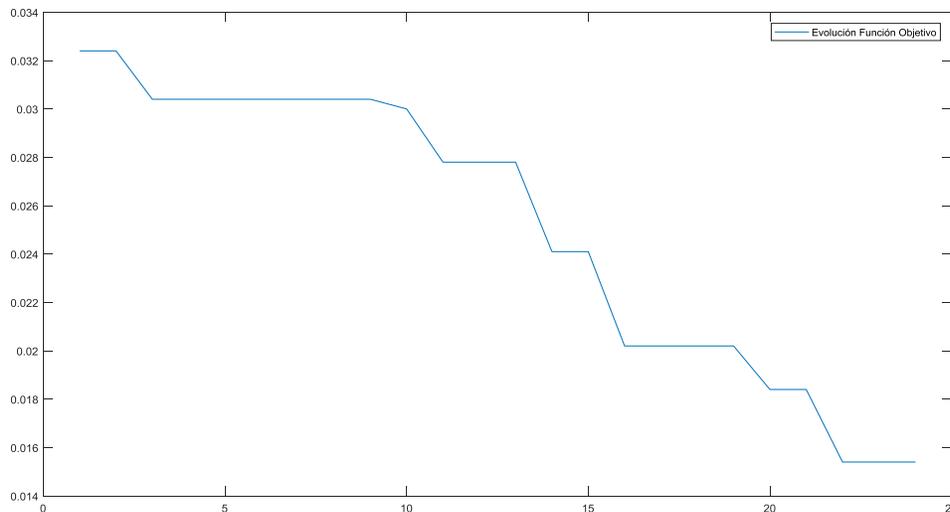
El valor que retorna la función del robot es el MRSE (mean of root of squared error) se obtiene de la ecuación 57. Como puede observarla el MRSE es el error medio del sistema basado en el error del eslabón 1 y el error del eslabón 2. Conforme el algoritmo avanza el error debe ir reduciéndose ya que al incrementar de iteración se tiene el mejor valor de las partículas analizadas anteriormente esto quiere decir el menor error de la iteración que se compara con los actuales dejando siempre el menor, esto lleva, por la naturaleza del algoritmo a un MRSE menor iteración con iteración.

Una vez desarrollados los algoritmos en Matlab para cada controlador se obtienen las gráficas del sistema que se analizan en la siguiente sección.

4.RESULTADOS

4.1 Controlador PD con pre-compensación y PSO

Las gráficas de respuesta muestran que las ganancias optimizadas del controlador PD con pre-compensación mediante el PSO dan un valor de la función objetivo bajo. Las figuras 4.1, 4.2, 4.3 muestran el comportamiento del PSO y la trayectoria seguida.



4.1 Evolución de la función objetivo

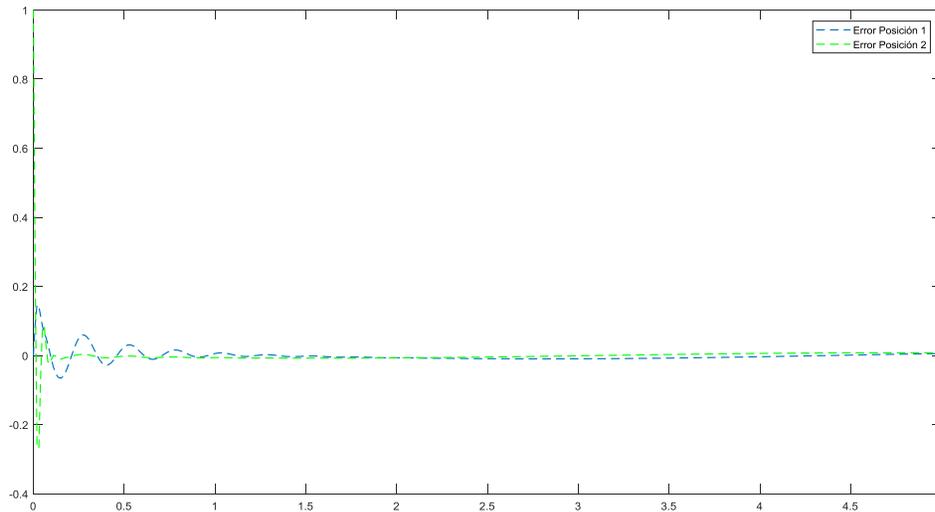
El valor óptimo de la función para el total de partículas se encontró en la iteración 22. El total de partículas utilizadas fueron 8 y 25 iteraciones. Las ganancias para cada eslabón quedaron definidas de la siguiente forma:

$$K_p = \begin{bmatrix} 8.00 & 0 \\ 0 & 8.00 \end{bmatrix} \quad \text{Ec. 62}$$

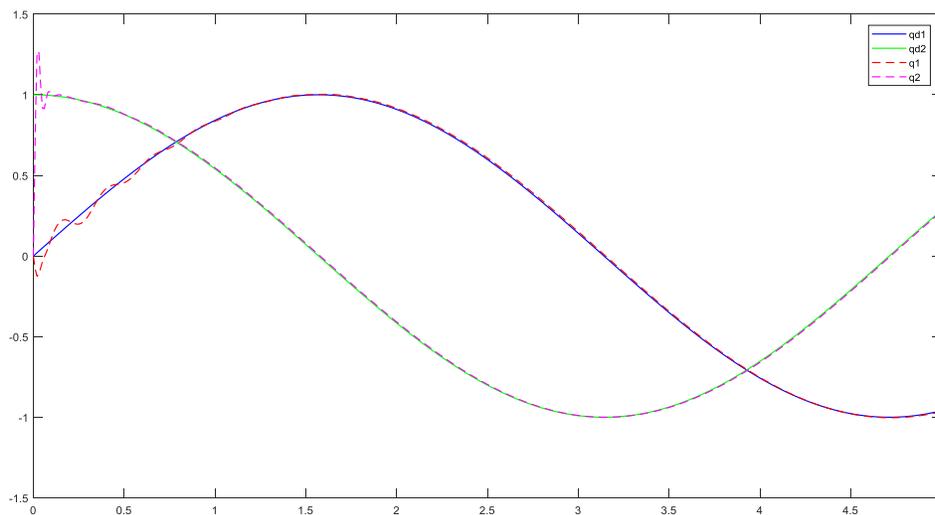
$$K_v = \begin{bmatrix} 0.0759 & 0 \\ 0 & 0.0595 \end{bmatrix}$$

Ec. 63

En la figura 4.2 se tiene el error por eslabón al seguir la trayectoria. Los valores de estos errores están en el rango de 0.0154 rad lo que demuestra un buen diseño del controlador y la aplicación del PSO. La trayectoria actual y la trayectoria deseada se encuentran en la figura 4.3.

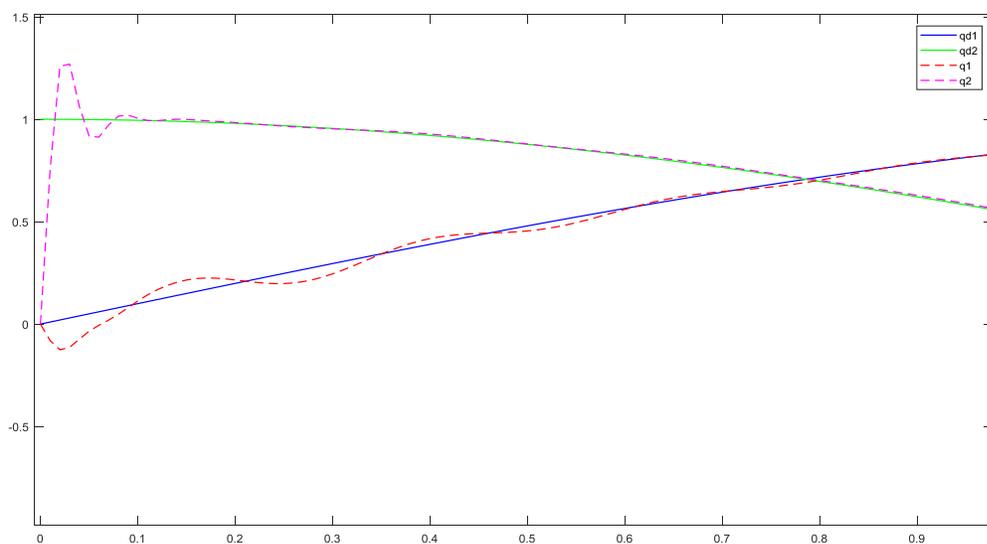


4.2 Error de posición PD con pre-compensación



4.3 Trayectoria actual y deseada

La trayectoria propuesta y la actual son muy similares, haciendo énfasis en la parte inicial de la trayectoria, dentro del primer segundo, puede observarse que el seguimiento tarda en llegar al objetivo para el primer eslabón y existe un sobrepaso en el segundo eslabón. La figura 4.4 muestra la etapa principal del controlador.



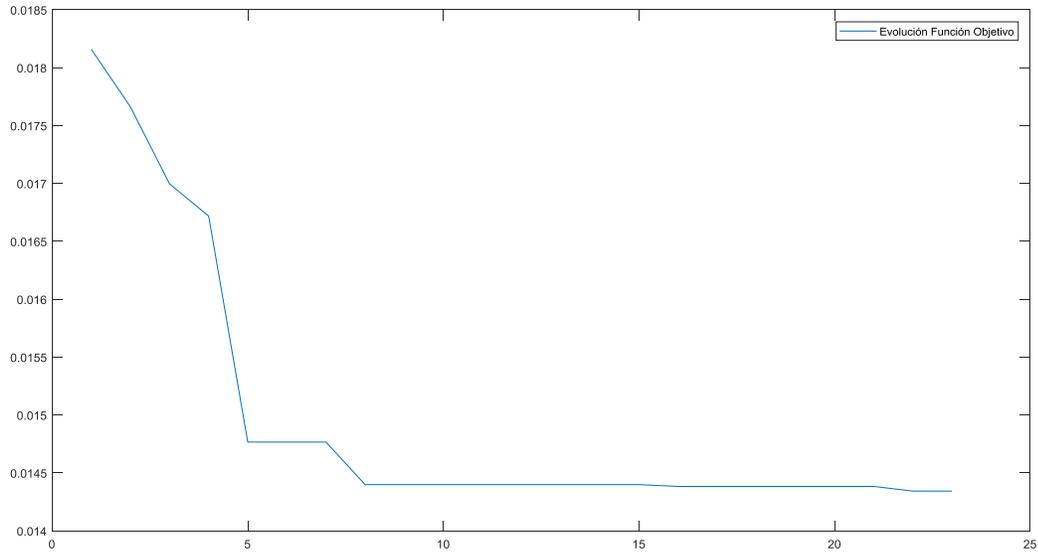
4.4 Inicio de trayectoria PD con pre-cálculo

Para este sistema el MRSE que se obtuvo fue de 0.0154 rad.

4.2 Controlador difuso con PSO

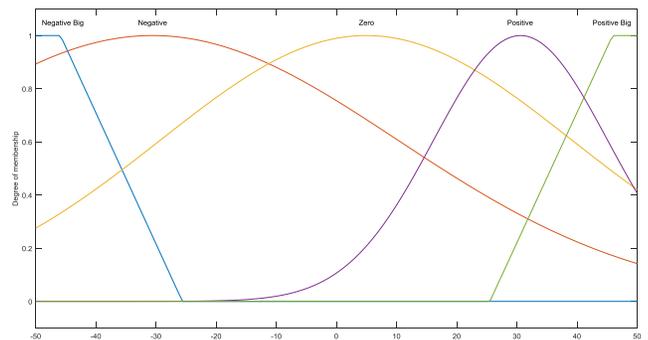
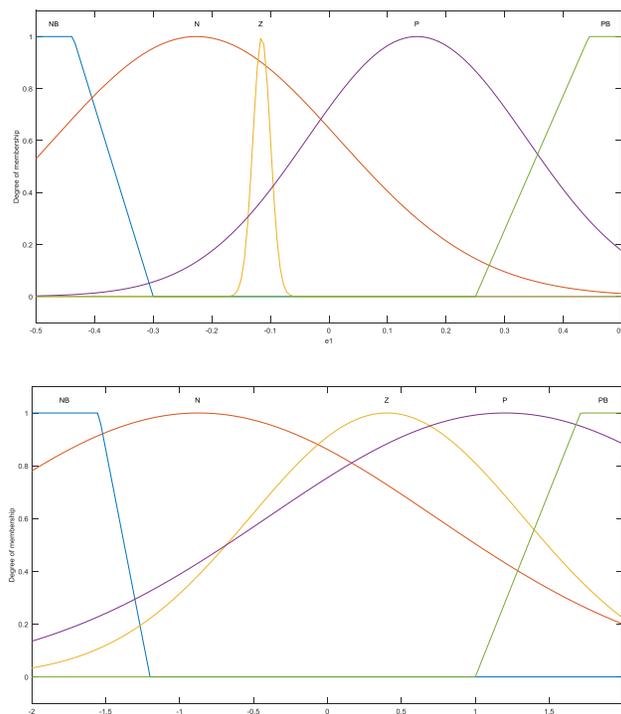
La sintonización del controlador difuso mediante el PSO no es sencilla ya que son 84 parámetros los que se modifican en cada iteración por lo que el costo computación es exigente. Las gráficas de respuesta muestran que las ganancias optimizadas del controlador PD con pre-compensación mediante el PSO dan un

resultado satisfactorio. Las figuras 4.1, 4.2, 4.3 muestran el comportamiento del PSO y la trayectoria seguida.



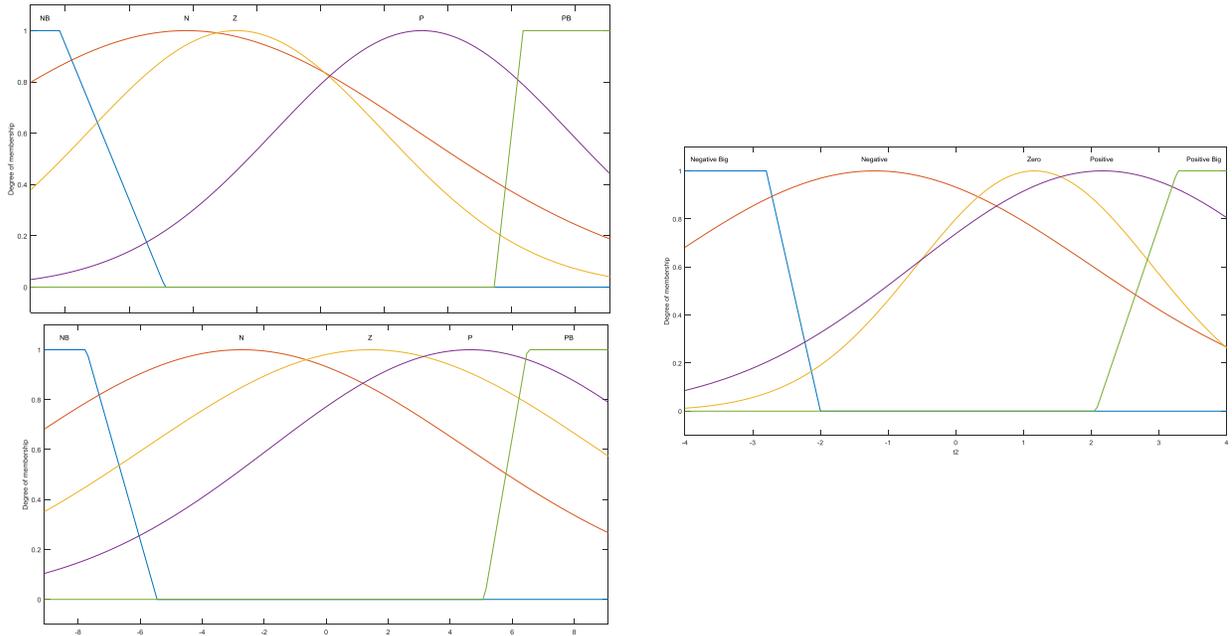
4.5 Evolución de función objetivo

En este caso puede observarse que el óptimo lo encuentra cerca de las últimas iteraciones, en la 2. Las partículas utilizadas fueron 8 y 25 iteraciones. Las funciones de membresía del sistema quedaron definidas como:

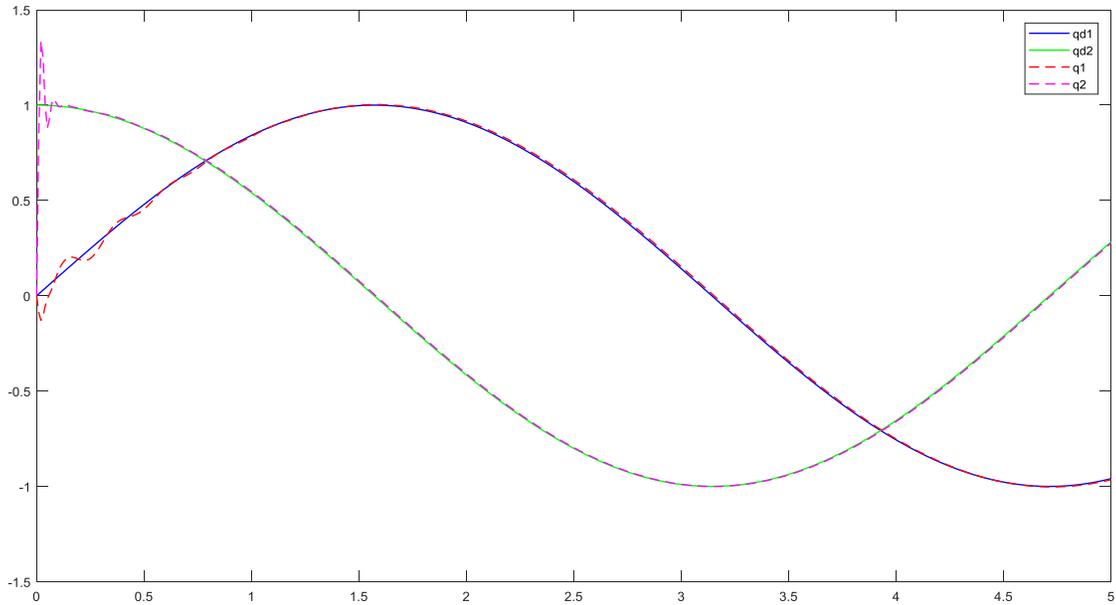


4.6 Funciones de membresía de entrada y salida de articulación 1

En la figura 4.2 se tiene el error por eslabón al seguir la trayectoria. Los valores de estos errores están en el rango de 0.04 rads lo que demuestra un buen diseño del controlador y la aplicación del PSO. La trayectoria actual y la trayectoria deseada se encuentran en la figura 4.3.

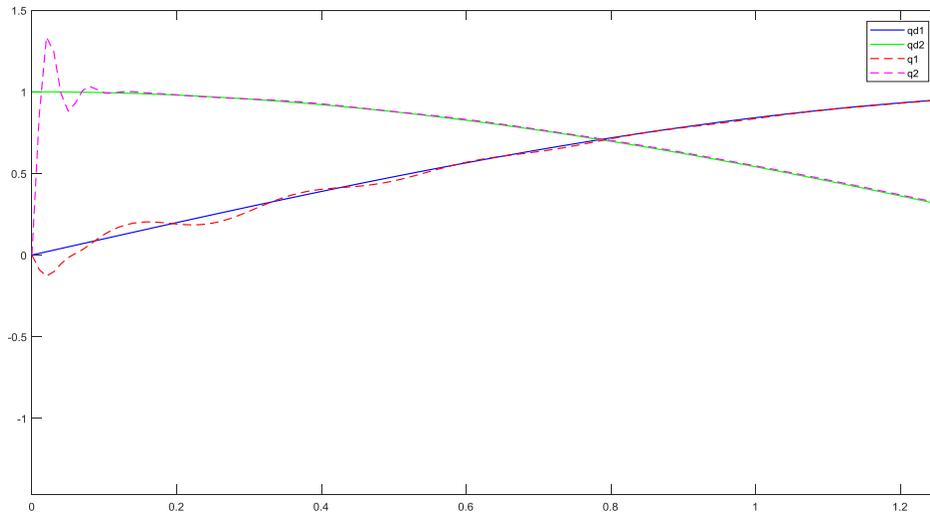


4.7 Funciones de membresía de entrada y salida de articulación 2



4.8 Trayectoria actual y deseada

La trayectoria propuesta y la actual son similares sin embargo el primer eslabón genera un rizo en la parte superior, al igual que el PD con pre-compensación puede observarse que el seguimiento tarda en llegar al objetivo y posteriormente se pudo acoplar. La figura 4.4 muestra la etapa principal del controlador.



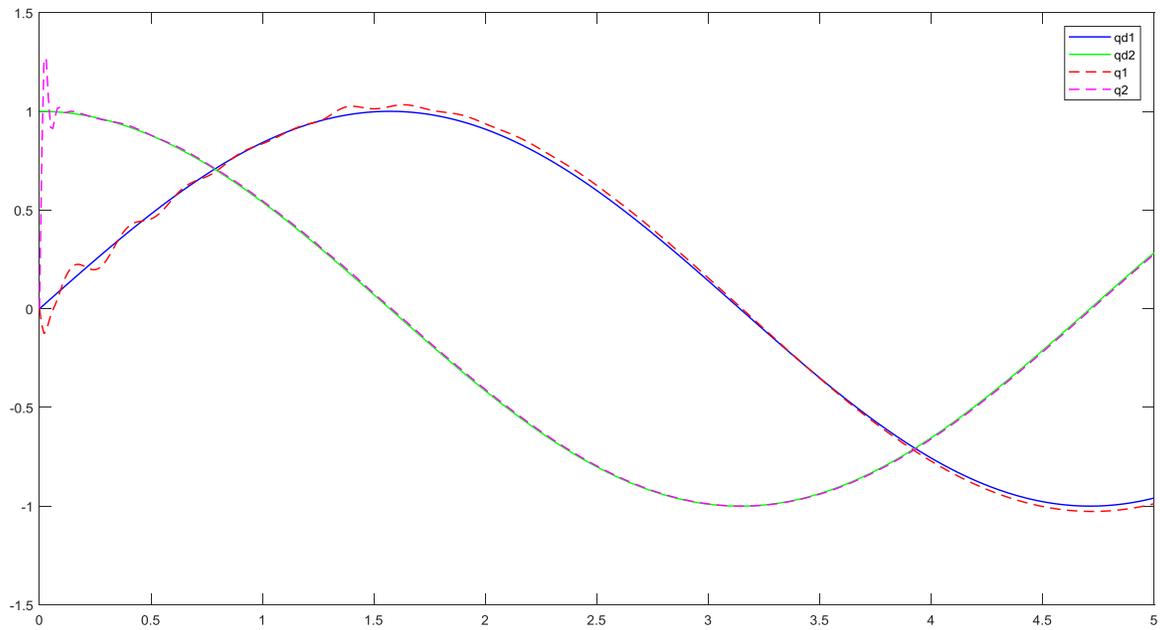
4.9 Inicio de trayectoria difuso calculado con PSO

El MRSE que se obtuvo fue de 0.0143. Se puede apreciar que el controlador llega más rápido a la posición en el eslabón dos mientras que en el eslabón uno tiene una pequeña oscilación.

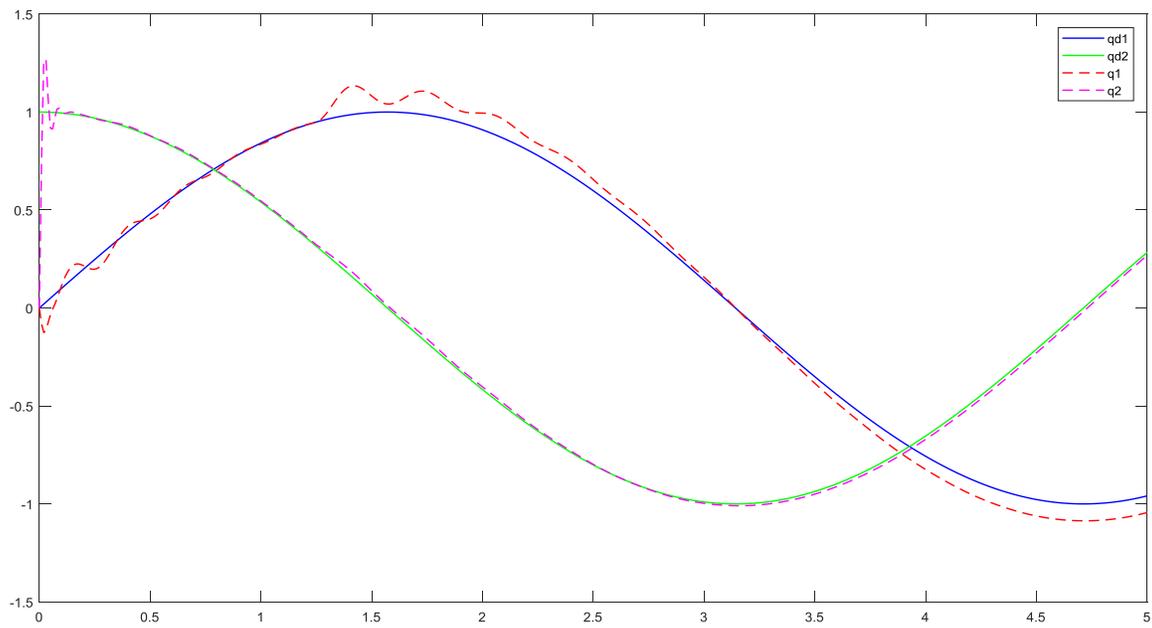
4.3 Variación de carga

La ventaja de un controlador difuso es la robustez que puede ofrecer al sistema ante variaciones de parámetros por lo que se llevó a cabo la modificación del peso en el eslabón final del sistema de 2DOF. Fueron tres variaciones en la masa del eslabón dos y una variación de la masa del eslabón uno. Las gráficas se presentan a continuación.

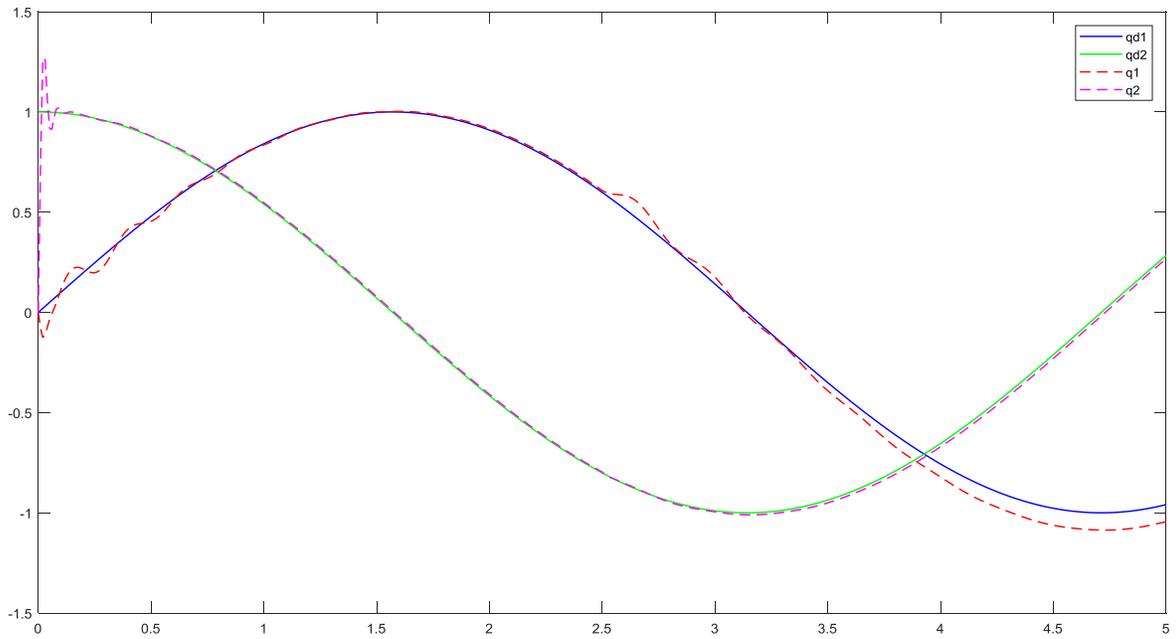
4.3.1 Gráficas controlador PD con pre-compensación



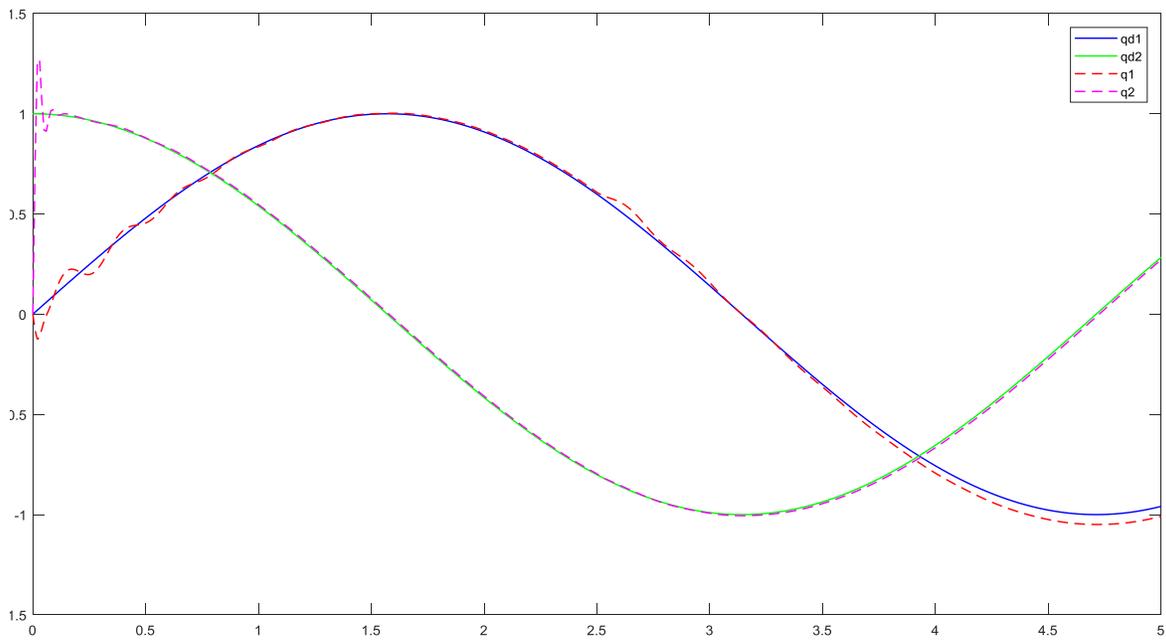
4.10 Aumento 10gr en masa 2 en t/4



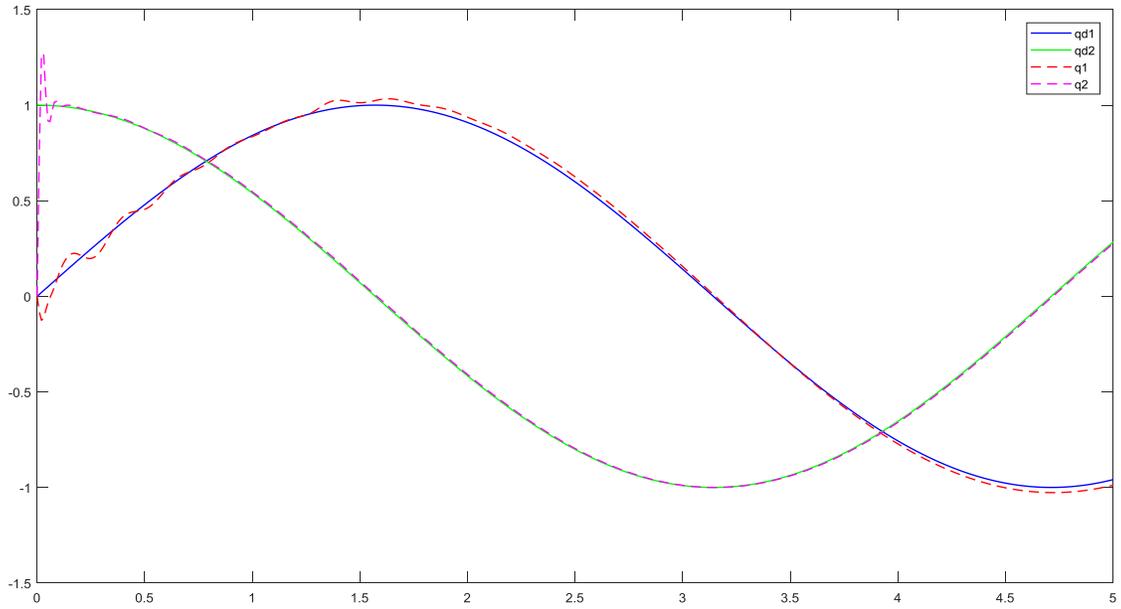
4.11 Aumento 20gr en masa 2 en t/4



4.12 Aumento 20gr en masa 2 en $t/2$



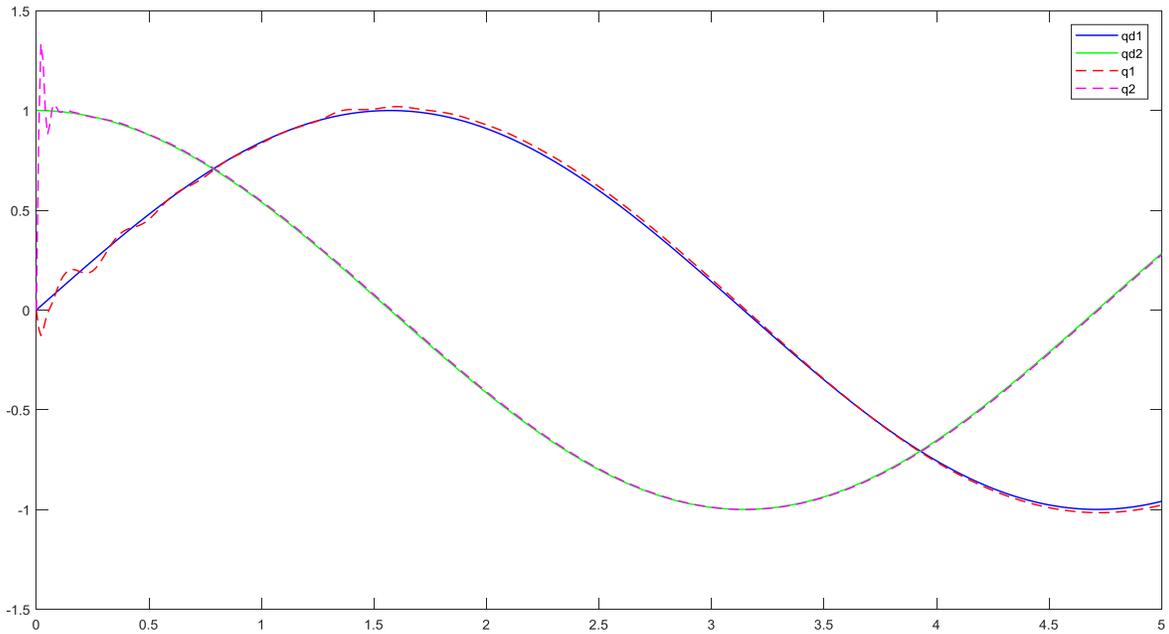
4.13 Reducción 10gr en masa 2 $t/2$



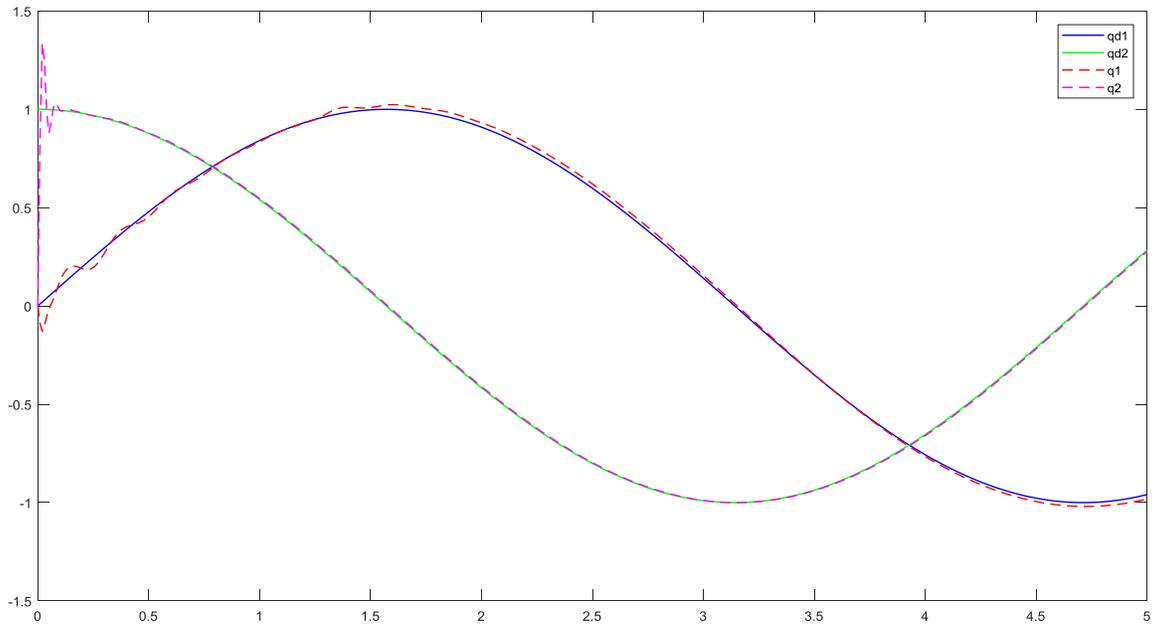
4.14 Aumento 10gr en masa 1, $t/4$

4.3.2 Gráficas de controlador difuso

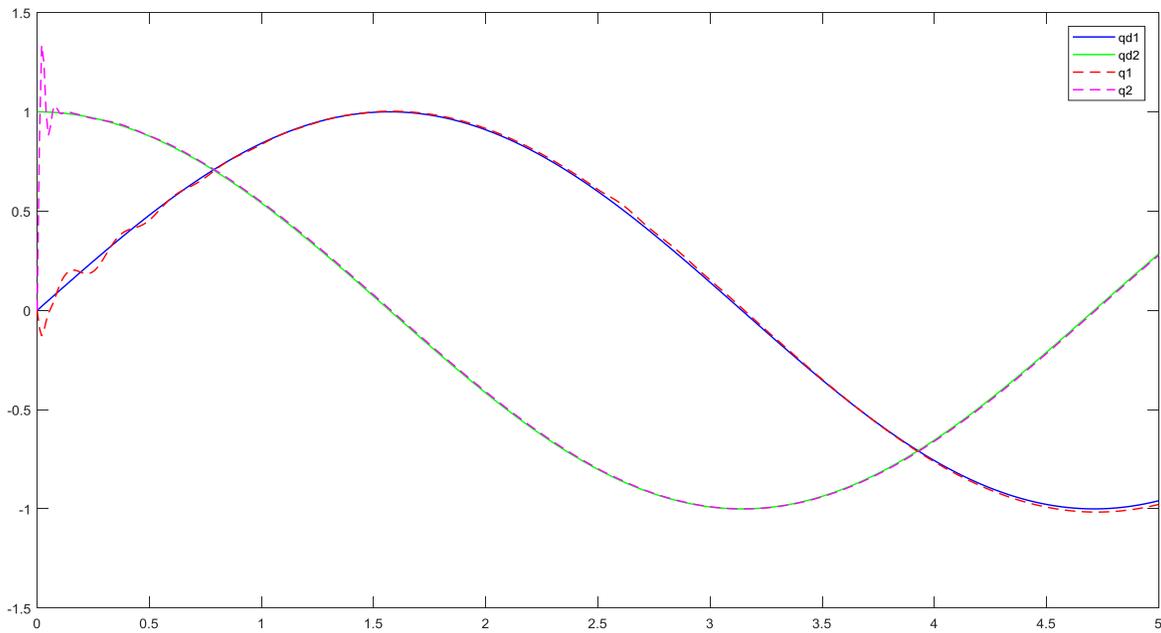
Los valores para el controlador difuso se pueden ver en las tablas anexas:



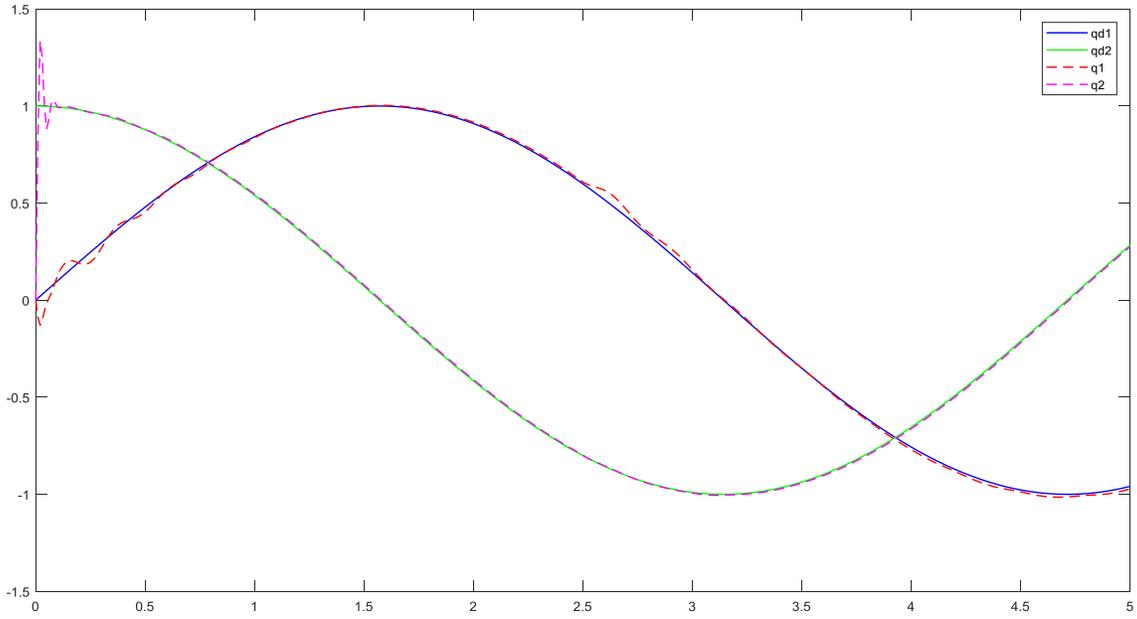
4.15 Aumento 10gr en masa 2 en $t/4$



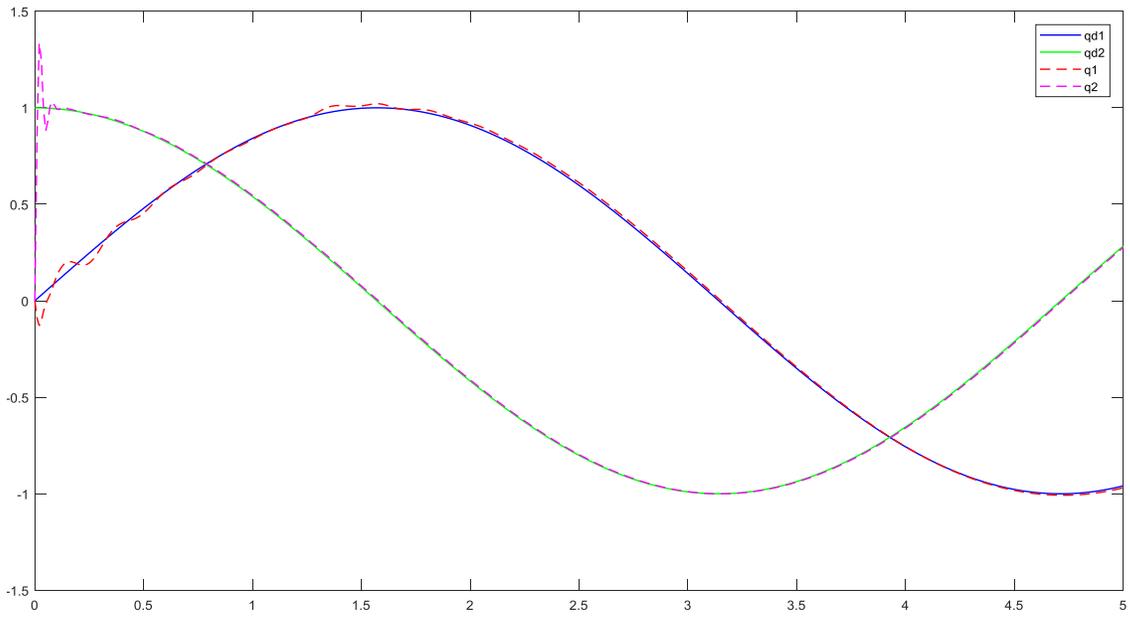
4.16 Aumento 20gr en masa 2 en $t/4$



4.17 Aumento 20gr en masa 2 en $t/2$



4.18 Reducción de 10gr en masa 2 en $t/2$



4.19 Aumento 10gr en masa 1 en $t/4$

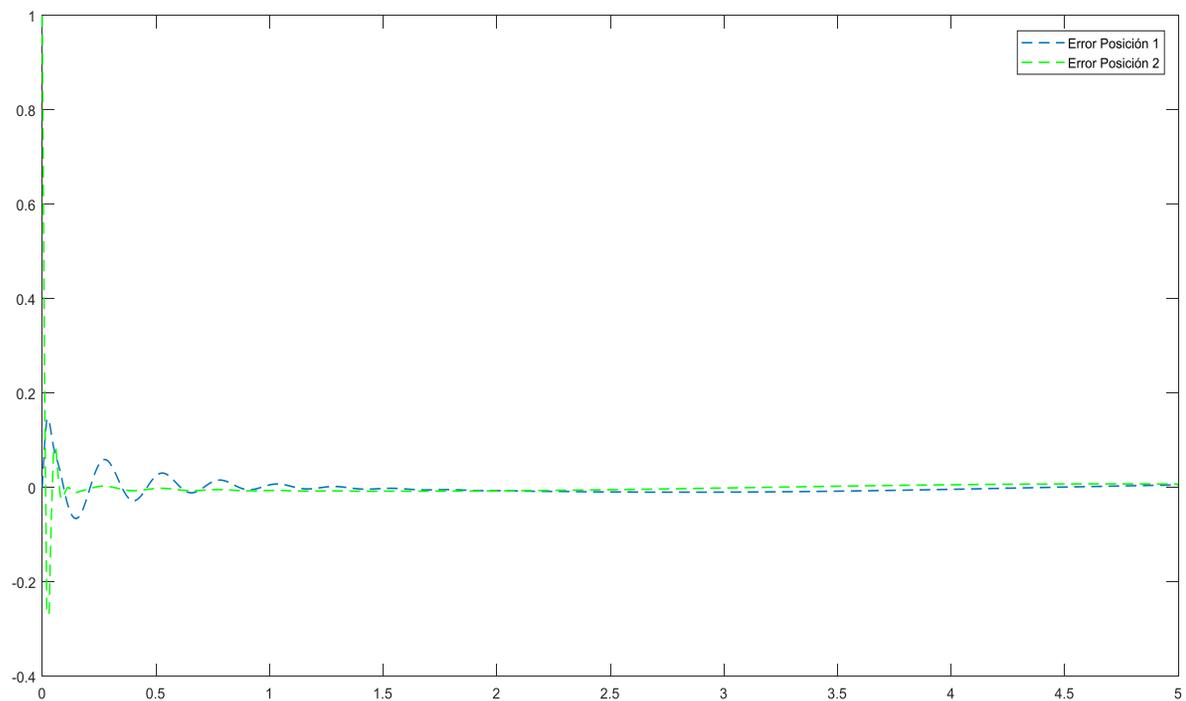
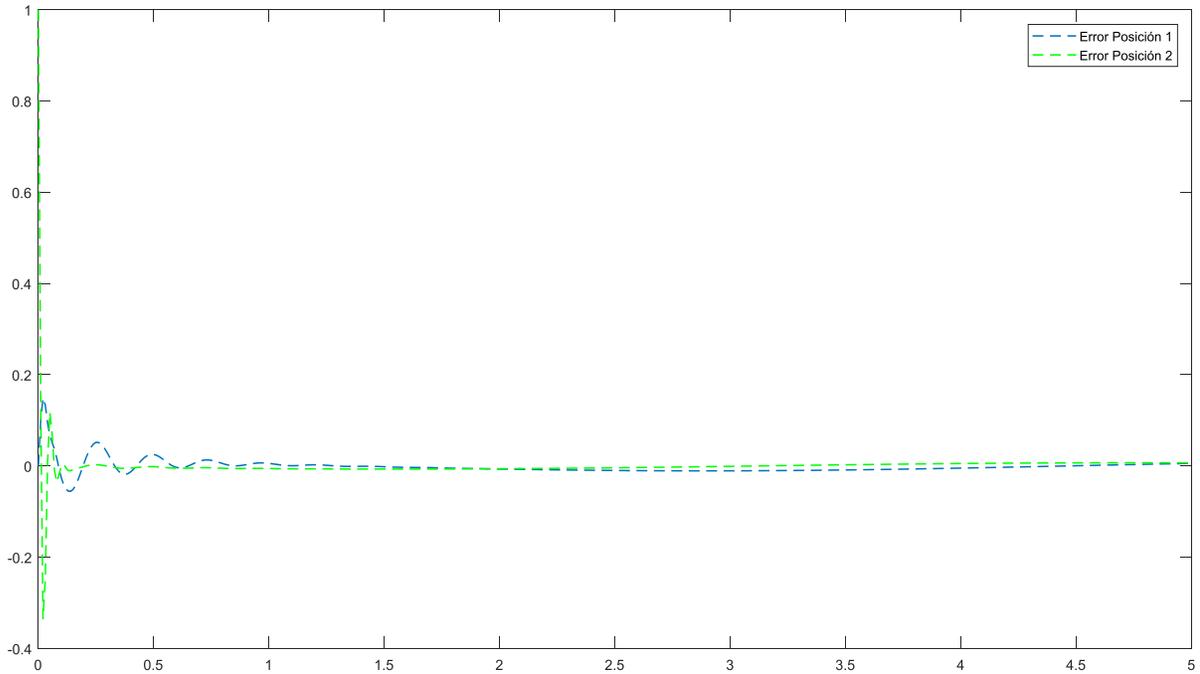
La tabla 4 muestra la diferencia entre el controlador PD con pre-compensación al enfrentar la carga y del controlador difuso en la misma situación.

Tabla 5 Respuesta a variación de carga

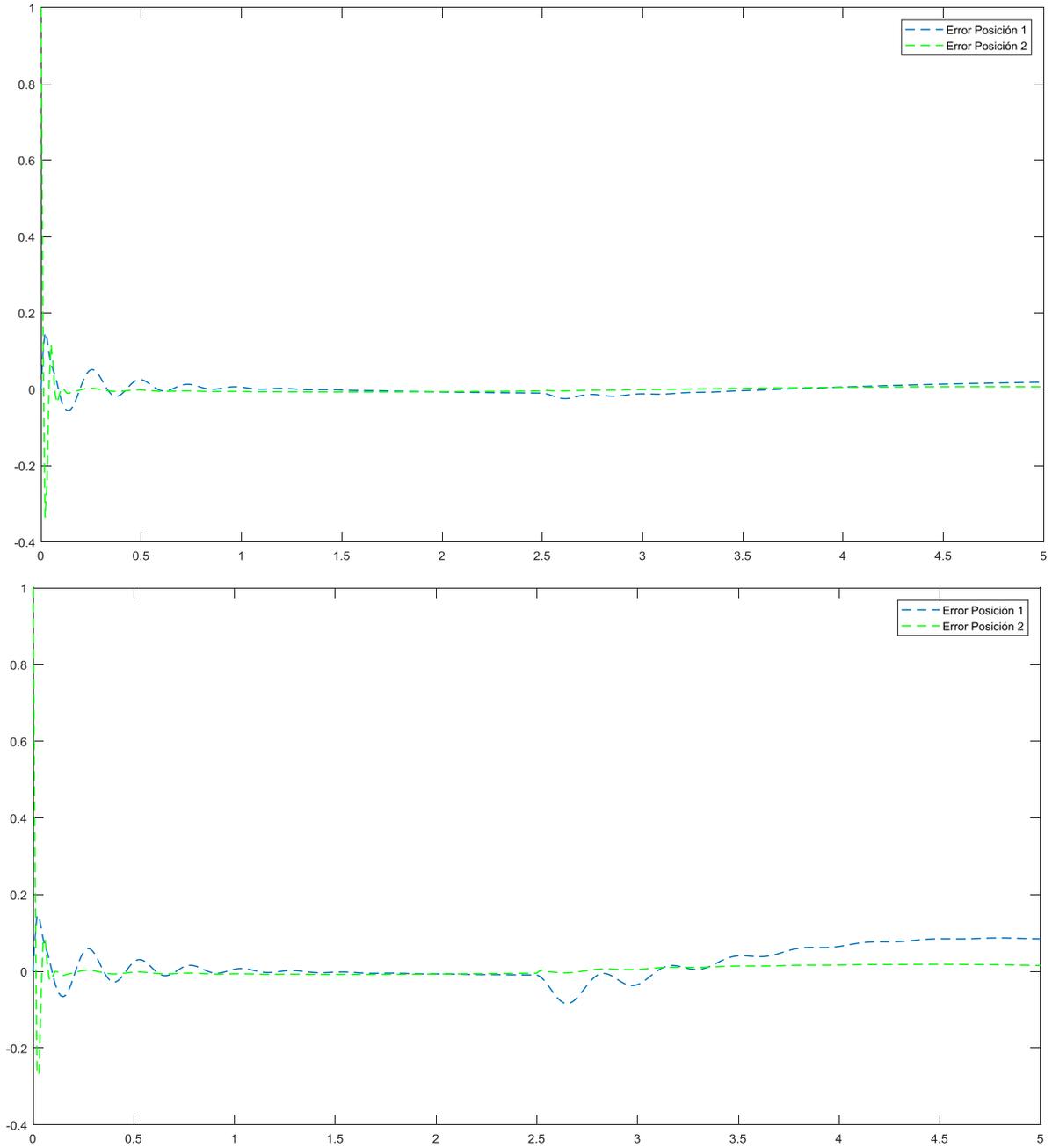
	MRSE PD Pre-compensado	Porcentaje aumento respecto a valor sin carga	MRSE Difuso	Porcentaje aumento respecto a valor sin carga
Sin carga extra	0.0154	0%	0.0143	0%
Aumento de 10gr en eslabón 2 en t/4	0.0221	50.6%	0.0186	30%
Aumento de 20gr en eslabón 2 en t/4	0.0569	269.4%	0.0203	41.9%
Aumento de 20gr en eslabón 2 en t/2	0.0393	155.1%	0.0193	34.9%
Reducción de 10gr en eslabón 2 en t/2	0.0271	75.9%	0.0177	23.7%
Aumento de 10gr en eslabón 1 en t/4	0.0247	60.3%	0.0161	12.5%

Como puede observarse los porcentajes varían más en el controlador PD mientras que en el difuso no llega más allá de 50% de incremento en el MRSE.

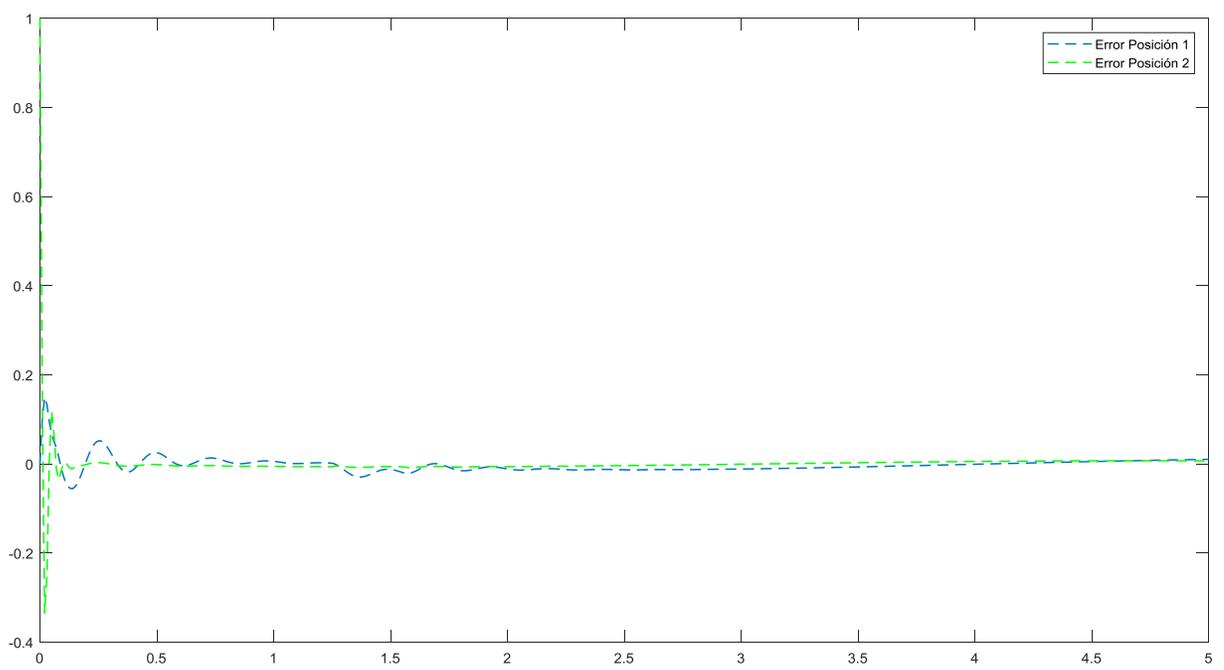
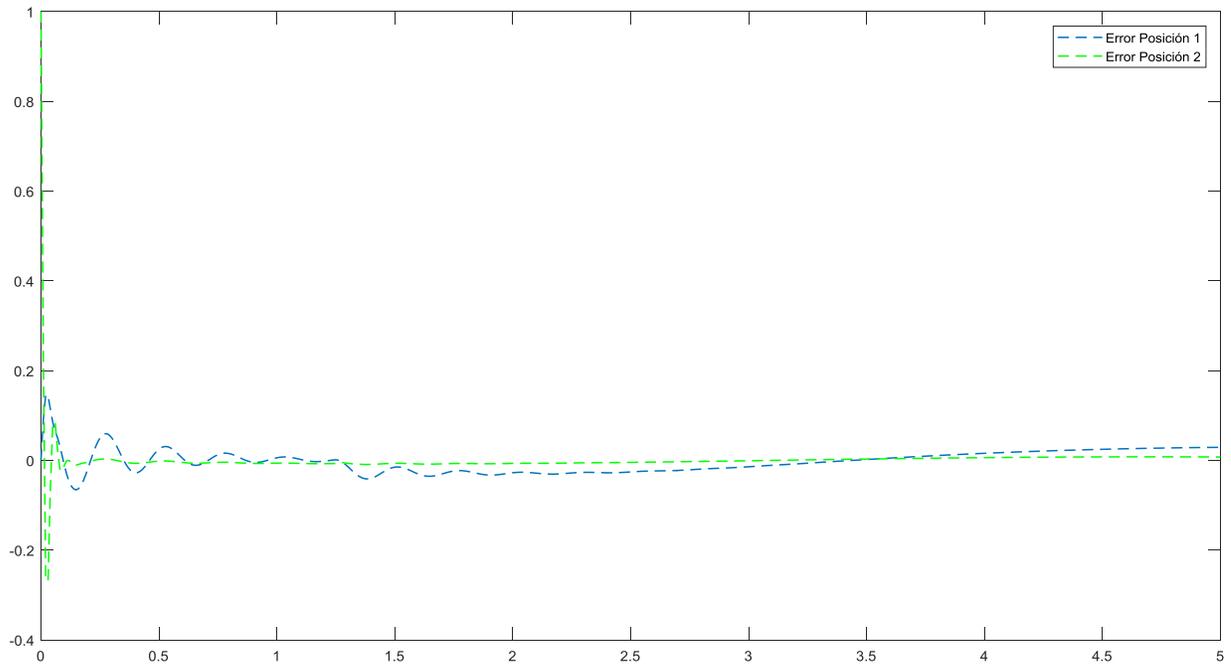
Algunas de las gráficas de error pueden apreciarse en las siguientes figuras:



4.20 Error de posición PD abajo, difuso arriba



4.21 Error de posición 20gr masa 2, $t/2$, difuso arriba, PD abajo



4.22 Error posición 10gr masa 1, PD arriba, difuso abajo.

5. CONCLUSIONES

La presente tesis tuvo como objetivo el diseño de un controlador difuso para el seguimiento de trayectoria de una plataforma de tres grados de libertad; debido a que no se tuvo el recurso para desarrollarla el proyecto se enfocó en un eslabón de dos grados de libertad presente en la plataforma.

Pudo observarse que los controladores de movimiento, en particular el PD con pre-compensación, requieren de la trayectoria para calcular las matrices que se deben tomar en cuenta al hacer el control.

El método de sintonización de ganancias es un método base del cual puede partirse para sintonizar eficientemente el controlador. Las ganancias k_p y k_d obtenidas mediante la sintonía del controlador son la base para ajustar el sistema ya que partiendo de estos parámetros los valores son ajustados manualmente por el usuario.

El método del PSO es un algoritmo de optimización que funciona perfectamente para sistemas mono-objetivos, en este caso el mono objetivo se obtuvo utilizando la Ec. 57 que permite obtener el mean square root error basado en los errores de los eslabones.

La lógica difusa en comparación con el controlador clásico presenta un error menor y permite variaciones en los elementos del sistema, también evita que el error crezca dándole robustez y evitando que se deba recalcular algún parámetro.

La implementación del controlador PD con pre-compensación y PSO permite tener un sistema de seguimiento de trayectoria eficiente sin embargo al introducir perturbaciones el error incrementa y el controlador no es capaz de manejar estos escenarios.

Las trayectorias propuestas en este documento son simples ya que para trayectorias más complejas el sistema necesita más puntos a analizar lo que implica costo computacional y tiempo de ejecución.

El modelo difuso propuesto fue general, posteriormente mediante el algoritmo PSO se modificó para adecuarlo a las necesidades de la trayectoria evitando tener un experto en la materia ya que el PSO a través de la optimización obtendrá los mejores valores del sistema.

Aunque los dos modelos respondieron de forma similar, al aplicar carga a uno de ellos la variación que se presenta no es trivial y el tiempo en recuperar la trayectoria puede ser crucial en algunos proyectos por lo que dependiendo de la aplicación un controlador puede funcionar mejor que el otro.

La sintonización de un controlador difuso no es un trabajo trivial y si bien se pueden generalizar las reglas si no se tiene un algoritmo como el PSO es necesario conocer o tener un experto del tema como soporte para el diseño del controlador.

El objetivo del proyecto establece que los controladores difusos no solamente siguen la trayectoria de forma más eficiente que los controladores difusos si no que tienen la capacidad de adaptarse a la variación de parámetros lo que les da robustez frente a controladores clásicos.

Queda abierta la posibilidad de utilizar un algoritmo MOPSO (Multiple Object Particle Swarm Optimization) y compararlo con el PSO y el controlador clásico, este algoritmo permite analizar cada error como una partícula y evitar utilizar el MRSE.

6. Bibliografía

Abdalla, M. 2018. Optimal Fuzzy Controller: Rule Base Optimized Genetarian. Journal of Control Engineering and Applied Informatics.

Åström, K. J., R. E. Klein, and A. Lennartsson. 2005. Dynamics and control. IEEE Control Syst. Mag. 25:26–47.

Ayas, M. S., E. Sahin, and I. H. Altas. 2014. Trajectory tracking control of a stewart platform. 16th Int. Power Electron. Motion Control Conf. Expo. PEMC 2014:720–724.

Bingül, Z. and Karahan, O. 2011. A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control. Expert Systems with Applications.

Bouallegue, S., Haggége, J., Ayadi, M. and Benrejeb, M. 2012. PID-type fuzzy logic controller tuning based on particle swarm optimization. Engineering Applications of Artificial Intelligence.

Brooks, F. P. 1988. Grasping reality through illusions - Interactive graphics serving science. :15.

Dong-Soo Kwon, Gi-Hun Yang, Chong-Won Lee, Jae-Cheol Shin, Youngjin Park, Byungbo Jung, Doo Yong Lee, Kyungno Lee, Soon-Hung Han, Byoung-Hyun Yoo, Kwang-Yun Wohn, and Jung-Hyun Ahn. 2001. KAIST interactive bicycle simulator. Proc. 2001 ICRA. IEEE Int. Conf. Robot. Autom. (Cat. No.01CH37164) 3:2313–2318.

Force, A. I. R., S. Command, U. States, and A. I. R. Force. 1977. Proceedings of the image conference.

Ghosh, B. B., B. K. Sarkar, and R. Saha. 2015. Realtime performance analysis of different combinations of fuzzy-PID and bias controllers for a two degree of freedom electrohydraulic parallel manipulator. Robot. Comput. Integr. Manuf. 34:62–69.

Hassan, Rania., Cohanim, Babak., De Weck Olivier. 2014. A comparison of particle swarm optimization and the genetic algorithm.

He, Q., X. Fan, and D. Ma. 2005. Full Bicycle Dynamic Model for Interactive Simulator. J. Comput. Inf. Sci. Eng. 5:373.

Heilig, M. 1962. 3,050,870 8. :16.

Heim, M. 2011. The metaphysics of virtual reality.

Herbuś, K., and P. Ociepka. 2016. Integration of the virtual model of a Stewart platform with the avatar of a vehicle in a virtual reality. IOP Conf. Ser. Mater. Sci. Eng. 145:42018. Available from: <http://stacks.iop.org/1757->

Jantzen, Jan., 2013. Design of Fuzzy Controllers

Kelly, R., and Santibañez, V. 2003. Control de movimiento de robots manipuladores.

Kennedy, James., and Eberhart, Russell., 1995. Particle Swarm Optimization

Kwon, D.-S., G.-H. Yang, Y. Park, S. Kim, C.-W. Lee, J.-C. Shin, S. Han, J. Lee, K. Wohn, S. Kim, D. Y. Lee, K. Lee, J.-H. Yang, and Y.-M. Choi. 2002. KAIST interactive bicycle racing simulator: the 2nd version with advanced features. "Intelligent Robot. Syst. 2002. IEEE/RSJ Int. Conf. on" 3:2961.

Liu, G., Z. Qu, X. Liu, and J. Han. 2014. Tracking Performance Improvements of an Electrohydraulic Gough-Stewart Platform using a Fuzzy Incremental Controller. *Ind. Robot An Int. J.* 41:225–235.

Maiti, R., Das Sharma, K. and Sarkar, G. 2018. PSO based parameter estimation and PID controller tuning for 2-DOF nonlinear twin rotor MIMO system. *International Journal of Automation and Control*

Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., and Valdez, M. 2013. Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. *Expert Systems with Applications*.

NASA, and Ames. 1986. Virtual Environment Display System.pdf.

Precup, Radu-Emil., Preitl, Stefan, M. Petriu, Emil., K. Tar József., Tomescu, Marius and Pozna Claudiu. 2009. Generic two-degree-of-freedom linear and fuzzy controllers for integral processes. *Journal of the Franklin Institute* 346.

Song, J., and U. Jung. 2003. Washout Algorithm with Fuzzy-Based Tuning for a Motion Simulator Copyright (C) 2003 NuriMedia Co ., Ltd . Copyright (C) 2003 NuriMedia Co ., Ltd . 17.

Surender, S. and Panigrahi, B. 2017. Application of swarm intelligent techniques with mixed variables to solve optimal problems. *International Journal of Bio-Inspired Computation*.

Sutherland, I. E. 1965. The ultimate display. *Proc. Congr. Int. Fed. Inf. Process.*:506–508. Available from: <http://citeseer.ist.psu.edu/viewdoc/summary>

Sutherland, I. E. 1968. A head-mounted three dimensional display. *Proc. AFIPS '68 (Fall, part I)*:757–764. Available from: <http://dl.acm.org/citation.cfm?id=1476686>

Vijay, M. and Jena, D. 2017. PSO based neuro fuzzy sliding mode control for a robor manipulator. *Journal of Electrical Systems and Information Technology*.

Yap, H. J., J. G. Ng, Z. A. Zakaria, Z. Taha, S. Chang, K. S. Yap, I. Manufacturing,

C. Engineering, and U. T. National. 2016. Design and Development of a 6-Dof System for Virtual Simulator. 5:31–39.

Yeh, S.-W., J.-J. Lo, J.-J. Huang, and Z.-Y. Fan. 2007. The development of fuzzy systems. Technol. E-Learning Digit. Entertain. 4469:829–840.

Yin, S., and Y. Yin. 2007. Study on virtual force sensing and force display device for two DOF robot. Sensors Actuators, A Phys. 140:65–74.

Zakaria, Z. A., and H. J. Yap. 2017. Design and Development of Fuzzy Systems. In: F. Ibrahim, editor. IFMBE Proceedings. p. 175.