

Modelado en el accionar con fricción de un
brazo robótico utilizando un controlador PD más
Red Neuronal
TESIS
que para obtener el grado de Maestro en
Ciencias (Matemáticas Aplicadas e Industriales)
presenta
Miguel Isaías Castillo Espínola

Director de tesis:
Dr. Joaquín Delgado Fernández

Junio de 2010

INDICE

1. INTRODUCCION.....	2
1.a. Introducción sobre Control.....	2
1.b. Objeto de la Tesis	7
2. NORMAS VECTORIALES	8
2.a. Normas vectoriales y matriciales.....	8
3. SISTEMAS LINEALES INVARIANTES EN EL TIEMPO LIT	15
3.a. Sistemas Dinámicos	15
3.a Definición de Sistemas Dinámicos Lineales Invariantes en el tiempo.....	16
Un sistema dinámico de control viene descrito por un sistema de la forma	16
3.b. La función de transferencia	17
3.c. Obtención del Modelo de Estado	18
3.d. Solución de la Ecuación de Estado de Sistemas Lineales	20
3.e. Controlabilidad y Observabilidad en Sistemas Lineales Invariantes.....	23
3.f. Estabilidad en sistemas LIT	27
4. ESTABILIDAD EN SISTEMAS NO-LINEALES	29
4.a. Análisis por linealización.....	29
4.b. Análisis de la estabilidad por el método de Lyapunov.....	33
4.b.2. Ejemplos del Método Directo	43
5. CONTROL DE SISTEMAS ADAPTATIVOS	51
5.a. Problema de Identificación: caso escalar	51
5.b. Leyes adaptativas.....	53
5.c. Ganancias adaptativas.....	55
5.d. Simulación de identificación de una planta lineal.....	55
5.e. Identificación de plantas no-lineales	59
5.f. Control directo; el caso escalar.....	63
5.g. Control por retroalimentación	64
6. REDES NEURONALES PARA CONTROL.....	70
6.a. Introducción	70
6.b. Estructura de la Red Neuronal	73
6.c. El algoritmo de Retropropagación	74
6.d. Aproximación a funciones continuas con la Red Neuronal	77
7. PERTURBACIONES	78
7.a. Tipos de perturbaciones.....	78
7.b. Fricción en servomecanismos	80
8. CONTROLADOR DE RETROALIMENTACION LINEAL TIPO PD.....	83
8.a. Un sistema amortiguado con fricción	83
8.b. Ecuaciones Euler-Lagrange para un brazo robótico	86
8.c. Dinámica del brazo robótico	89
8.d. Diseño del controlador de retroalimentación lineal tipo PD	91
9. CONTROLADOR PD Y RED NEURONAL	102
9.a. Estructura del controlador PD y red neuronal.....	102
9.b. Reglas de ajuste de pesos	106
10. CONTROLADOR PD Y RED NEURONAL AUMENTADA ROBUSTA	115
10.a. Ampliación de la red neuronal.....	115
10.a.1. Redes Neuronales para aproximación de funciones con discontinuidades	115
10.b. Control adaptativo robusto en la red neuronal.....	121
10.c. Control por pasividad en la red neuronal	126
10.d. Controlador final PD y red neuronal con redes aumentadas.....	131
10.d.1. Aproximación de la Red Neuronal y la No-linealidad en el problema de Parámetros	131
10.e. Simulaciones.....	139
11. OBSERVACIONES Y CONCLUSIONES.....	143
11.a. Observaciones.....	143
11.b. Conclusiones	144
REFERENCIAS.....	145
ANEXO	147

1. INTRODUCCION

1.a. Introducción sobre Control

La teoría del control comienza a mediados del siglo XIX cuando el funcionamiento de reguladores mecánicos comenzaba a ser analizados de una manera matemática. Aquellos reguladores actuaban de una manera estable si todas las raíces de los polinomios característicos estaban contenidas en la parte izquierda del plano complejo. Hoy en la teoría de control diversos aspectos de estabilidad están presentes.

La teoría de control ha estado fuertemente ligada con las matemáticas a partir de la mitad del siglo pasado y ha tenido considerable influencia sobre el cálculo de variaciones, la teoría de ecuaciones diferenciales y la teoría de procesos estocásticos.

Los sistemas de control automático, que tienen sus orígenes en la más remota antigüedad han sustentado sus raíces en las matemáticas aplicadas y se están convirtiendo, de manera más creciente, en componentes esenciales y críticos de cualquier sistema dinámico. El control es una de las pocas disciplinas que trasciende las fronteras de los campos tradicionales de la ingeniería (mecánica, eléctrica, química, nuclear, etc.)

El enfoque de Poincaré (finales del siglo XIX) al estudio de problemas dinámicos rápidamente se popularizó y se vino a conocer como *método del espacio de estado*. Así el concepto de estado se hace dominante en el estudio de los sistemas dinámicos. Las *variables de estado* representan pues la mínima cantidad de información que nos resume todo el pasado dinámico del sistema y es todo lo que necesitamos conocer para poder predecir su evolución futura frente a cualquier señal de entrada que le apliquemos.

La utilización del tratamiento del espacio de estados condujo rápidamente a una comprensión profunda de los problemas científicos y matemáticos del control y se puede considerar que su introducción marca la emergencia de este como una disciplina científicamente madura.

Por los cuarentas, ingenieros eléctricos, mecánicos y químicos fueron diseñando inventos de control automático en sus respectivos campos para llegar usando métodos muy similares en diferentes rutas y esbozados bajo terminologías completamente diferentes. Aunque al principio no había conexión entre esos desarrollos, fue reconocido gradualmente que los conceptos tenían una base común y para el final de la segunda guerra mundial, vino a aparecer una teoría que era elegante matemáticamente y universal. En los últimos cuarenta años el control por retroalimentación ha evolucionado de un arte a la disciplina de una ciencia, la cual cruza las fronteras extendidas del diseño, desarrollo y producción por una parte y de las matemáticas por el otro.

De hecho, a principios de los sesentas, Bellman decía:

“ la teoría de control ha emergido como una disciplina matemática que puede existir independientemente de sus aplicaciones”.

La teoría del control automático ha dado testimonio del constante esfuerzo hacia el incremento de la velocidad y la precisión. Métodos de respuesta frecuencial fueron desarrollados basados en los esfuerzos de Black, Nyquist y Bode en el diseño de amplificadores de retroalimentación electrónicos. Usando estos métodos que ahora son clasificados bajo el rubro de **Control Clásico**, fue posible llevar a cabo tanto el análisis como la síntesis de sistemas de lazo cerrado de una manera sistemática basados en respuestas frecuenciales de lazo abierto. En el curso del tiempo, estos métodos formaron los fundamentos de la teoría del control por retroalimentación y llegado a adecuarse idealmente para el diseño de sistemas lineales de tiempo invariante. En los cincuenta y sesenta con el desarrollo de la tecnología espacial, el problema de control por retroalimentación creció y se hizo más complejo. Requerimientos estrictos de precisión, peso y costo de las aplicaciones espaciales apresuró el crecimiento de nuevas técnicas para el diseño de sistemas de **Control Óptimo**.

Modelos con mas de una variable dependiente aparecieron comúnmente y anunciaban la era del **Control Multivariable**. Finalmente, la inevitable presencia de ruido tanto en las variables de entrada como en la salida aparecido por la estática de las soluciones para los problemas de estimación y control. Y el campo de prueba de una mezcla de técnicas de control con aquellas que estaban bien establecidas. Así la solución del problema del Regulador Gaussiano Cuadrático Lineal usando el principio de separación y el desarrollo del filtro Kalman llegan a ser hechos sobresalientes en el campo en los sesenta.

Aún cuando los mas grandes esfuerzos fueron hechos en la precisión del control. Los modelos lineales no fueron a menudo encontrados ser de gran validez y más descripciones de precisión del proceso fueron necesarias. Los modelos simples del proceso tenían que ser reemplazados por unos mas complejos y consideraciones de incertidumbre en las entradas, valores de parámetro y estructura del sistema incrementaron su aparición en el cuadro. La importancia en el diseño de controladores rápidos y precisos cambio la atención del control automático hacia áreas tales como los **Sistemas Adaptativos, Sistemas Auto-optimizables y Sistemas Auto-organizables**.

Dos conceptos estructurales fundamentales para la comprensión de los sistemas dinámicos que son **controlabilidad y observabilidad** fueron introducidos.

Se definió un sistema de control como un arreglo de componentes físicos conectados de tal manera que el arreglo se pueda comandar, dirigir o regular a sí mismo o a otro sistema dinámicamente o activamente.

Las entradas y salidas del sistema se definen para un sistema de control como:

La entrada; en un sistema es el estímulo o excitación que se aplica a un sistema de control desde una fuente de energía externa, generalmente con el fin de producir, de parte del sistema de control. Una respuesta previamente especificada.

La salida; en un sistema de control se define como la respuesta obtenida del sistema de control. Puede ser o no igual a la respuesta especificada que la entrada implica.

Los sistemas de control se clasifican en dos grandes categorías.

Sistema de Control de Lazo Abierto. Es aquel en el cual la acción de control es independiente de la salida.

Sistema de Control de Lazo Cerrado. Es aquel en el cual la acción de control es de cierto modo dependiente de la salida.

Calibrar. Significa establecer o restablecer una relación entre la entrada y la salida con el fin de obtener del sistema la exactitud deseada.

Los sistemas de lazo abierto tienen tres rasgos sobresalientes.

- 1.- La habilidad que estos tienen para ejecutar una acción con exactitud está determinada por su calibración.
2. Estos problemas no tienen problemas de inestabilidad.
- 3.- Son susceptibles a tener variaciones importantes causadas por el efecto que causan las perturbaciones cuando ellas afectan al sistema.

Por otra parte, la **Retroalimentación** es la característica principal de los sistemas de lazo cerrado, la cual permite que la salida (o cualquier otra variable controlada del sistema) sea comparada con la entrada al sistema (o con una entrada a cualquier componente interna del sistema o con un subsistema de este) de tal manera que se pueda establecer la acción de control apropiada como función de la entrada y la salida y evitar así el no poder alcanzar los objetivos deseados.

El problema de los sistemas de control es de dos tipos. El análisis y el diseño del sistema deseado.

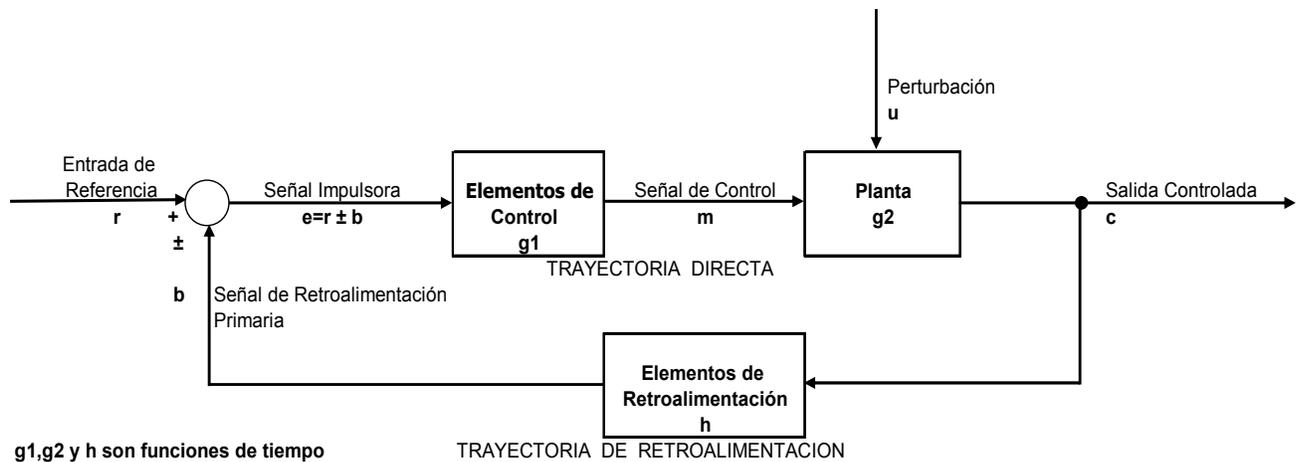
Diseño por análisis. Se lleva acabo modificando las características de un sistema existente ó de un modelo estándar del sistema.

Diseño por síntesis. Definiendo la forma del sistema directamente a partir de sus especificaciones

El Modelo, forma susceptible a ser sometida a análisis, diseño y evaluación que puede representarse por medio de:

- 1.- Ecuaciones diferenciales y otras relaciones matemáticas.
- 2.- Diagramas de bloque
- 3.- Gráficas de flujo de señales

A continuación se presenta la configuración básica de un sistema de control simple de lazo cerrado (por retroalimentación) que está representado por diagrama de bloques.



SISTEMA DE CONTROL RETROALIMENTADO

La *Planta* denotada por **g2** llamada también sistema controlado, es el cuerpo, proceso o máquina de la cual se va a controlar una cantidad o condición particular. Los elementos de control **g1**, también llamados *el Controlador* son los componentes requeridos para generar la señal de control apropiada **m** que se aplica a la planta. Los elementos de retroalimentación **h** son los componentes que se requieren para establecer la relación funcional entre la señal de retroalimentación primaria **b** y la salida controlada **c**; es esa cantidad o condición de la planta que se controla. La entrada de *referencia* **r** es una señal externa aplicada a un sistema de control por retroalimentación con el fin de ordenar a la planta una acción especificada, a menudo representa un comportamiento ideal de la salida de la planta. La señal de retroalimentación primaria **b** es una señal que es función de la salida controlada **c**, y que se suma algebraicamente a la entrada de referencia **r** para obtener la señal impulsora **e**. La señal impulsora **e**, también denominada el error o la acción de control, es la suma algebraica de la entrada de referencia **r** mas o menos (usualmente menos) la retroalimentación primaria **b**. La variable manipulada **m** (señal de control) es esa cantidad o condición que los elementos de control **g1** aplican a la planta **g2**. Una *perturbación* **u** es una señal de entrada indeseable que afecta el valor de la salida controlada **c**. Puede entrar a la planta sumándose con **m** o a través de un punto intermedio, como se muestra en el diagrama en bloque de la figura. La Trayectoria Directa es la vía de transmisión desde la señal impulsora **e** hasta la salida controlada **c**. La Trayectoria de Retroalimentación, es la vía de transmisión desde la salida controlada **c** hasta la señal de retroalimentación primaria **b**.

Generalmente un controlador o compensador es requerido para filtrar la señal de error, para que el criterio de control sea cierto o las especificaciones sean satisfechas, este criterio puede involucrar, pero sin ser limitado a lo siguiente:

- 1.- Rechazo de perturbaciones
- 2.- Errores del estado estable
- 3.- Características de las respuestas transitorias

4.- Sensibilidad a cambios en los parámetros de la planta.

Hay tres aspectos importantes sobre el modelado matemático: **Representación, Precisión y Generalización** en relación al diseño del modelo que de solución al problema físico.

La **representación matemática** del fenómeno físico:

El fenómeno físico debe estar representado adecuadamente por un sistema dinámico, los estados de las variables, las entradas, las salidas. Poder saber el estado del sistema para distintos momentos. Otro aspecto importante es la **precisión** del modelo que permite lograr los objetivos consistentemente a pesar de la presencia de perturbaciones. La **generalización** nos permite llegar a conclusiones en relación al comportamiento del sistema ante la consideración inicial de los parámetros y con esto delimitar el conocimiento del comportamiento del sistema dinámico.

El incremento reciente en la complejidad de los sistemas hechos por el hombre, ha forzado a posturas severas sobre el diseño de las técnicas de los controladores modernos; requerimientos de desempeño más rigurosos en cuanto a velocidad de respuesta y precisión en objetivos, han cambiado los límites del control moderno. Además, la tolerancia actual a faltas y fallas demanda que los controladores tengan características de sistemas inteligentes.

Los sistemas complejos tienen perturbaciones desconocidas, dinámicas no modeladas e inestabilidades que no están estructuradas; por ejemplo, los actuadores que dirigen los sistemas modernos tienen importantes no-linealidades debidas a la fricción, zonas muertas, retrocesos por holguras en los mecanismos y retrasos en tiempo.

Aquí se presenta el diseño mejorado de un controlador el cual se basa en redes neuronales y que se dirige junto con la dinámica del sistema de control con PD. Los sistemas de **Redes Neuronales RN** pertenecen a una gran categoría llamada estructuras de redes no-lineales que tienen propiedades de gran importancia para sistemas de control por retroalimentación.

Estas redes artificiales están formadas por interconexiones de múltiples nodos y pueden aprender por la modificación de los pesos de los nodos interconectados; este proceso es el llamado *tuning* (ajuste). Cambiando los pesos, se cambia el conocimiento almacenado en la red, en efecto, los valores de los pesos representan la memoria de la red. Aquí veremos que algunos aspectos de control están vinculados a algunos aspectos de la teoría de las Redes Neuronales, su rápida convergencia del error de seguimiento es comparable a la de controladores adaptativos directos.

Por "Control de un Proceso" debemos comprender, cualitativamente, la habilidad para dirigir, alterar o mejorar su funcionamiento, y un sistema de control es aquel en el cual algunas cantidades de interés son mantenidas en un valor prescrito con más o menos precisión. Un control se hace verdaderamente automático cuando los sistemas son hechos para que se autoregule. La esencia del concepto de *Retroalimentación* esta en la terna: **medición, comparación y corrección**. Por medición

de las cantidades de interés, comparación con el valor deseado y uso del error para corrección del proceso, la cadena acostumbrada de causa y efecto en el proceso se convierte en un lazo cerrado hacia el interior de sucesos interdependientes.

Para el pensamiento común de las personas, automatización significa reemplazar trabajo humano por máquinas, esto trae consigo miedo y desconfianza, sin embargo mientras la automatización elimina algunos trabajos en soldadura, maquinado o ensamble en líneas de trabajo; se crean otros tales como: instalación y mantenimiento de robots, desarrollo de software y demás. Al mismo tiempo el incremento en la productividad traída por la automatización hace capaz a la sociedad de competir exitosamente en una economía global con el resultado que el nivel de vida se eleva. En el presente el robot representa la más elevada forma de automatización. Una clase general de procesos industriales tiene la estructura de un sistema dinámico precedido por las no-linealidades del actuador. Problemas en el control de estos procesos son exacerbados cuando la precisión requerida es alta, como en mecanismos de microposicionamiento. Debido a la naturaleza no-analítica de las no-linealidades del actuador y el hecho que sus funciones no-lineales exactas son desconocidas, tales procesos representan todo un reto para el diseñador matemático de control.

1.b. Objeto de la Tesis

La motivación principal en la elaboración del presente trabajo estriba en mostrar las bases teóricas y técnicas recientes en el diseño de controladores mejorados con redes neuronales que permiten que los brazos manipuladores actúen siguiendo una trayectoria a una determinada velocidad, logrando establecerse los márgenes y valores para la estabilidad y el error que permitan un desempeño que alcance valores importantes en cuanto a precisión y confiabilidad; considerando la presencia de no linealidades por perturbaciones internas propias de dinámicas no modeladas y las perturbaciones externas en este caso representadas por el fenómeno de la fricción.

El perfilado general del Objeto de la Tesis está en lograr los siguientes puntos:

- a) Dar las bases para el diseño estable de tres distintos controladores para el accionar de un brazo manipulador.
- b) Comparar sus desempeños bajo condiciones ideales (sin perturbación) y bajo condiciones reales (con perturbación).
- c) Analizar sus distintos desempeños conforme a requisitos impuestos ya sea en cuanto a una posición deseada o en cuanto al seguimiento de una trayectoria de referencia.
- d) Establecer conclusiones y ventajas.

En el primer diseño de controladores se trata con un Controlador de Retroalimentación Lineal Proporcional-Derivativo, llamado solamente Controlador PD, el segundo un controlador compuesto,

formado por un Controlador PD mas un Controlador de Redes Neuronales y el tercero también es un controlador compuesto de un Controlador PD mas un Controlador de Redes Neuronales Aumentadas.

Se consideran en el primer estado condiciones ideales de funcionamiento en el sistema, es decir sin perturbaciones externas, en el segundo estado se incluyen en el sistema perturbaciones externas debidas a la fricción.

Los modelos y los diseños están en base las teorías de Sistemas Lineales, de Sistemas No-Lineales y los Sistemas Adaptativos Estables. La estabilidad de los sistemas es comprobada utilizando principalmente el método directo de Lyapunov. Adicionalmente se utilizan varios conceptos y algoritmos de algunas técnicas de control como son: Control por Retroalimentación, Control con Redes Neuronales, Control Robusto, Control por Pasividad y Control Multivariable.

Se analizan las ventajas y limitaciones de los tres diferentes controladores en cuanto al modelado y reducción del error así como su capacidad de rechazo a las perturbaciones

2. NORMAS VECTORIALES

2.a. Normas vectoriales y matriciales

El espacio Euclideo n-dimensional se denota por \mathbb{R}^n ; \mathbb{R}^1 es denotado también como \mathbb{R} . \mathbb{R}^+ es el conjunto de números reales no negativos. \mathbb{C} denota el conjunto de todos los números complejos. \mathbb{C}^- se refiere al semiplano izquierdo del plano complejo, y $\mathbb{C}^+ = \mathbb{C} - \mathbb{C}^-$. El símbolo s sirve para denotar una variable compleja y también para denotar el operador diferencial $\frac{d}{dt}$.

$\mathbb{R}^{n \times n}$ es el conjunto de todas las matrices de reales ($n \times n$). denota la matriz de identidad se denotar por I , cuyo tamaño puede inferirse del contexto. Todos los vectores son vectores columna y el índice T denota la transpuesta. El símbolo \triangleq define la variable anterior. Si $x = [x_1, x_2, \dots, x_n]^T$, su norma, denotada por $\|x\|$, es la norma Euclidea $\|x\|^2 \triangleq x_1^2 + \dots + x_n^2$

Para cualquier vector $x \in \mathbb{R}^n$ con componentes x_i , recordemos que la *norma* -1 es la suma de los valores absolutos de x_i , la *norma* -2 es la raíz cuadrada de la suma de los cuadrados de x_i , y la *norma* - ∞ es el máximo valor absoluto de x_i .

Generalmente.

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

Habría que recordar que todas las normas vectoriales son equivalentes, en el sentido que para cualquier vector x , existen constantes a_1, a_2 tales que $a_1 \|x\|_p \leq \|x\|_q \leq a_2 \|x\|_p$ para cualesquiera normas p, q . Esto significa que cualquier norma puede ser usada mostrar limitación de señales o convergencia.

Dada cualquier norma vectorial, la norma-matriz inducida está definida por

$$\|A\| \equiv \sup \left\{ \frac{\|Ax\|}{\|x\|} \mid x \neq 0 \right\}$$

Recordando que $\|A\|_1$, es igual al máximo de las sumas por columna de los valores absolutos.

$\|A\|_\infty$ es igual al máximo de las sumas por renglón de los valores absolutos y $\|A\|_2$ es el máximo Valor Singular de A .

$$\|A\|_1 = \max_{j=1,2,\dots,n} \sum_{i=1}^m |a_{ij}|$$

$$\|A\|_\infty = \max_{i=1,2,\dots,m} \sum_{j=1}^n |a_{ij}|$$

La norma matricial inducida satisface:

$$\|Ax\| \leq \|A\| \|x\|.$$

Para cualesquiera dos matrices A, B tenemos:

$$\|AB\| \leq \|A\| \|B\|.$$

El producto interno Frobenius es $\langle A, B \rangle_F = A^T B$ para cualquier matriz B dimensionalmente compatible. La norma matricial de Frobenius definida por:

$$\|A\|_F^2 = \text{traza}(A^T A) = \sum a_{ij}^2$$

De acuerdo a la desigualdad de Schwarz, tenemos:

$$|\langle A, B \rangle_F| \leq \|A\|_F \|B\|_F.$$

Con a_{ij} los elementos de A . La norma Frobenius no es una norma inducida para alguna norma vectorial, pero es compatible con la *norma - 2* en el sentido que:

$$\|Ax\|_2 \leq \|A\|_F \|x\|_2$$

A menos de que se diga lo contrario, todas las normas serán Frobenius y todas las normas vectoriales serán *norma - 2*.

Una matriz Q es definida positiva si $x^T Q x > 0$ para toda $x \neq 0$, y semidefinida positiva si $x^T Q x \geq 0$ para toda $x \neq 0$, definida negativa si $x^T Q x < 0$ para toda $x \neq 0$, semidefinida negativa si $x^T Q x \leq 0$ para toda $x \neq 0$. Estos son, denotados respectivamente como $Q > 0$, $Q \geq 0$, $Q < 0$ y $Q \leq 0$. Si Q es simétrica, entonces es definida positiva si todos los valores propios son positivos, semidefinida positiva si todos los valores propios son no-negativos, definida negativa si todos los valores propios son negativos y semidefinida negativa si todos los valores propios son no-positivos.

Una norma para el vector x también está definida por $x^T Q x$ donde Q es cualquier matriz simétrica definida positiva. Estas *norma* – Q satisfacen la desigualdad de Sylvester

$$\sigma_{\min}(Q) \|x\|^2 \leq x^T Q x \leq \sigma_{\max}(Q) \|x\|^2$$

Con $\sigma_{\min}, \sigma_{\max}$ el mínimo y máximo valor singular, respectivamente. Esto también significa que

$$-x^T Q x \leq \sigma_{\min}(Q) \|x\|^2$$

Si una matriz simétrica A es definida positiva, entonces escribimos $A > 0$. Si A_1 y A_2 son matrices simétricas definidas positivas, podemos escribir $A_1 > A_2$ si $A_1 - A_2 > 0$.

2.b Normas de funciones en espacios \mathcal{L}^p

Espacios de Banach

Un espacio de Banach es por definición un espacio vectorial normado completo. Esto quiere decir que un espacio de Banach es un espacio vectorial V sobre el campo de los números reales o el de los complejos con una norma $\|\cdot\|$ tal que toda sucesión de Cauchy (con respecto a la métrica $d(x, y) = \|x - y\|$) en V tiene un límite en V .

Si X e Y son dos espacios de Banach, entonces podemos formar su suma directa $X + Y$, la cual tiene una estructura de espacio vectorial topológica natural pero no una norma canónica. Sin embargo es un también un espacio de Banach para algunas normas equivalentes, por ejemplo:

$$\|x \oplus y\| = \left(\|x\|^p + \|y\|^p \right)^{\frac{1}{p}}, 1 \leq p \leq \infty$$

Si M es un subespacio lineal cerrado de un espacio de Banach X , entonces el espacio cociente X/M es un espacio de Banach también.

Cada producto interno define una norma asociada. El espacio de producto interno es llamado un espacio de Hilbert si esta norma asociada esta completa.

Todo espacio de Hilbert es un espacio de Banach. El recíproco también se mantiene bajo ciertas condiciones.

El espacio de todas las funciones continuas $f : [a, b] \rightarrow K$ definidas sobre un intervalo cerrado $[a, b]$ tiene la estructura de espacio de Banach si definimos la norma según

$$\|f\| = \sup \{|f(x)| : x \in [a, b]\}$$

Este espacio es completo con esta norma, y el espacio de Banach resultante se denota por $C[a, b]$.

Relación con espacios de Hilbert

Como se menciona anteriormente, todo espacio de Hilbert es un espacio de Banach porque, por definición, un espacio de Hilbert es completo con respecto a la norma asociada a su producto interior. No todos los espacios de Banach son espacios de Hilbert. Una condición necesaria y suficiente para que un espacio de Banach sea también un espacio de Hilbert es la identidad del paralelogramo:

$$\|u + v\|^2 + \|u - v\|^2 = 2(\|u\|^2 + \|v\|^2)$$

para toda u y v en V , y donde $\|\cdot\|$ es la norma sobre V .

El espacio de Lebesgue L^p es siempre un espacio de Banach pero sólo es un espacio de Hilbert cuando $p = 2$.

Si la norma de un espacio de Banach satisface esta identidad, el producto interno asociado que forma dentro de un espacio de Hilbert está dado por la identidad de polarización.

Si V es un espacio real de Banach, entonces la identidad de polarización es:

$$\langle u, v \rangle = \frac{1}{4} (\|u + v\|^2 - \|u - v\|^2)$$

y en el caso que V sea un espacio de Banach complejo la identidad de polarización está dada por

$$\langle u, v \rangle = \frac{1}{4} (\|u + v\|^2 - \|u - v\|^2 - i(\|u + iv\|^2 - \|u - iv\|^2))$$

Espacios L^p

Los espacios L^p son los espacios vectoriales normados más importantes en el contexto de la teoría de la medida y de la integral de Lebesgue.

Consideremos (X, Σ, μ) un espacio de medida. Se define el espacio vectorial:

\mathcal{L}^p , para $p \in [1, \infty)$ Como el espacio de todas las funciones medibles f que cumplen:

$$\int |f|^p d\mu < \infty$$

Asimismo, se define el espacio \mathcal{L}^∞ como el espacio de las funciones medibles f que verifican:

$$\inf \left\{ a \in \mathbb{R} : \mu(\{x \in X : |f(x)| \geq a\}) = 0 \right\} < \infty$$

Es decir, aquellas funciones medibles acotadas. Una norma natural para definir en estos espacios sería:

$$\|f\|_p = \left(\int |f|^p d\mu \right)^{\frac{1}{p}}$$

$$\text{Si } p < \infty, \text{ y } \|f\|_\infty = \inf \left\{ a \in \mathbb{R} : \mu(\{x \in X : |f(x)| \geq a\}) = 0 \right\}$$

Sin embargo, una aplicación así definida no resulta norma, ya que no se cumple $\|f\|_p = 0 \Rightarrow f = 0$, pues cualquier función que sea igual a la función nula, salvo en un conjunto de medida nula, tendrá norma cero.

Así, se define la siguiente relación de equivalencia R sobre \mathcal{L}^p : $fRg \Leftrightarrow f = g$. Se prueba que efectivamente es una relación de equivalencia, y se defina $L^p = \mathcal{L}^p / R$, es decir., el espacio vectorial cuyos elementos son las clases de equivalencia de la relación R . Considerando entonces sobre L^p las normas anteriormente definidas (donde f es cualquier representante de la clase de equivalencia),

se prueba que $\|\cdot\|_p$ resulta ser norma y que su valor no depende del representante de la clase de equivalencia escogido. Usualmente no se hace distinción entre función y clase de equivalencia en este contexto.

1. L^p es Banach
2. L^2 es un Espacio de Hilbert, dotado del producto interno $\langle f, g \rangle = \int f g d\mu$.
3. Si $p \in (1, \infty)$, L^p es reflexivo.
4. Si denotamos por E al espacio de las funciones simples, se cumple que $E \cap L^p$ es denso en L^p .
5. Si $p \in (1, \infty)$, el dual topológico de L^p es L^q donde q es tal que $\frac{1}{p} + \frac{1}{q} = 1$.
6. Si el espacio de medida es σ -finito, entonces el dual de L^1 se identifica con L^∞ .
7. El espacio de las funciones infinitamente derivables en un abierto $\Omega \subseteq \mathbb{R}^n$ a soporte compacto y que están en L^p con $1 \leq p < \infty$, es denso en L^p , es decir $c_l_{L^p(\Omega)}(C_0^\infty(\Omega) \cap L^p(\Omega)) = L^p(\Omega)$.

Los espacios L^p a veces son llamados *Espacios de Lebesgue*, forman una clase importante de ejemplos de *Espacios de Banach* en Analisis funcional y de espacios vectoriales topológicos. Los espacio de Lebesgue tienen múltiples aplicaciones en la Física e Ingeniería.

Operadores lineales

Si V y W son espacios de Banach sobre el mismo campo K , el conjunto de todos los mapeos K -lineales continuos $A: V \rightarrow W$ se denota por $L(V, W)$. Es de notar que en espacios dimensionales infinitos no todas los mapeos lineales son continuos automáticamente. En general un mapeo lineal sobre un espacio normado es continuo si y sólo si este esta acotado sobre una bola unitaria. Así, en el espacio vectorial $L(V, W)$ puede definirse la norma de un operador A , como

$$\|A\| = \sup \{ \|Ax\|_W \mid x \in V, \|x\|_V \leq 1 \}$$

Con respecto a esta norma, $L(V, W)$ es un espacio Banach. Cuando $V=W$, el espacio $L(V)=L(V,V)$ forma una Algebra de Banach unitaria; la operación multiplicación está dada por la composición lineal de mapeos.

Espacios ℓ^p

La *norma* $-L^p$ puede ser extendida a vectores teniendo un número infinito de componentes, produciendo el espacio ℓ^p que tiene como casos especiales:

- ℓ^1 es el espacio de series convergentes absolutamente,
- ℓ^2 es el espacio de sucesiones cuadráticas sumables, el cual es un espacio de Hilbert y
- ℓ^∞ es el espacio de sucesiones acotadas.

El espacio de sucesiones tiene una estructura natural de espacio vectorial por la aplicación de la adición y la multiplicación escalar coordenada por coordenada. Explícitamente, para $x = (x_1, x_2, \dots, x_n, x_{n+1}, \dots)$ una secuencia infinita de números reales (o complejos) define el vector

Suma como

$$(x_1, x_2, \dots, x_n, x_{n+1}, \dots) + (y_1, y_2, \dots, y_n, y_{n+1}, \dots) = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n, x_{n+1} + y_{n+1}, \dots)$$

Siendo la acción escalar dada por:

$$\lambda(x_1, x_2, \dots, x_n, x_{n+1}, \dots) = (\lambda x_1, \lambda x_2, \dots, \lambda x_n, \lambda x_{n+1}, \dots)$$

Y se define la *norma* $-p$ $\|x\|_p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p + |x_{n+1}|^p + \dots)^{\frac{1}{p}}$

El espacio ℓ^p se define entonces como el conjunto de todas las sucesiones infinitas de números reales o complejos tales que la *norma* $-p$ es finita.

También se define la *norma* $-\infty$ como:

$$\|x\|_\infty = \sup(|x_1|, |x_2|, \dots, |x_n|, |x_{n+1}|, \dots)$$

y el correspondiente espacio ℓ^∞ de todas la sucesiones acotadas. Resulta que

$$\|x\|_\infty = \lim_{p \rightarrow \infty} \|x\|_p$$

si el lado derecho es finito o el lado izquierdo es infinito. Por lo tanto se considera en espacios ℓ^p para $1 \leq p \leq \infty$.

La *norma* $-p$ así definida sobre ℓ^p es en realidad una norma, y ℓ^p junto con esta norma es un espacio de Banach. El espacio ℓ^2 es el único espacio ℓ^p que es un espacio de Hilbert ya que cualquier norma que es inducida por un producto interno debe satisfacer la identidad del paralelogramo

$$\|x + y\|_p^2 + \|x - y\|_p^2 = 2\|x\|_p^2 + 2\|y\|_p^2$$

El espacio C_0 está definido como el espacio de todas las sucesiones que convergen a cero, con una norma idéntica a $\|x\|_\infty$. Es un subespacio cerrado de ℓ^∞ , por lo tanto un espacio Banach.

3. SISTEMAS LINEALES INVARIANTES EN EL TIEMPO LIT

3.a. Sistemas Dinámicos

En el estudio de sistemas adaptativos, se restringe la atención a sistemas dinámicos que son gobernados por ecuaciones diferenciales ordinarias conteniendo un número finito de parámetros. Si x_1, x_2, \dots, x_n representa la n-coordenadas en un espacio Euclideo n-espacio R^n y t el tiempo, el funcionamiento de un típico sistema dinámico dimensional-finito es descrito por las ecuaciones diferenciales

$$\dot{x}_i(t) = f_i(x_1(t), x_2(t), \dots, x_n(t), t) \quad i = 1, 2, \dots, n$$

O de manera equivalente por el vector de ecuación diferencial

$$\dot{x} = f(x, t) \quad \dots\dots\dots (2.1)$$

Donde x y f son vectores columna con componentes x_i y f_i ($i = 1, 2, \dots, n$) respectivamente. Las variables x_i son referidas como variables de estado y x es referido como el vector estado. La notación $x(t)$ indica que los componentes x_i de x son funciones de t . Nos referiremos al espacio dimensional $(n + 1)$ de las cantidades x_1, x_2, \dots, x_n, t como el espacio de movimiento M . Si estas son continuas, entonces el punto $(x(t), t)$ se mueve a lo largo de un segmento curvo en un espacio de movimiento cuando t varía sobre un intervalo $[t_1, t_2]$. La proyección del movimiento sobre el espacio de estado es llamada curva de estado, trayectoria del movimiento, o solución de la ecuación (2.1). Si el tiempo inicial es t_0 , y el valor inicial de $x(t)$ es $x(t_0) = x_0$, se denotará la solución de la Ecuación (2.1) como $x(t; x_0, t_0)$ donde $x(t_0; x_0, t_0) = x_0$. En los casos donde la condición inicial es evidente para el contexto, la solución es simplemente denotada como $x(t)$.

De la existencia del teorema de Caratheodory se sabe que para toda (x, t) perteneciente al dominio B en M , (i) $f(x, t)$ es continua en x para toda t fija, (ii) medible en t para toda x fija, y (iii) $\|f(x, t)\| \leq \mu(t)$ donde $\mu(t)$ es integrable en $|t - t_0| \leq \alpha$ entonces para algún $\alpha_1 > 0$ existe una solución $x(t; x_0, t_0)$ para $|t - t_0| \leq \alpha_1$. Se denotarán la clase de funciones que satisfacen las condiciones (i)-(iii) por F . En este trabajo el interés estará en soluciones de ecuaciones diferenciales que existan para todo $t \geq t_0$.

Una solución constante $x(t_0; x_0, t_0) \equiv x_0$ es un estado de equilibrio o un punto singular de la Ecuación (2.1). Si x_0 es la única solución constante en la vecindad de x_0 , esto es llamado un estado de equilibrio aislado. Si el lado derecho de la ecuación (2.1) no depende de t , la ecuación es llamada autónoma. Si $f(x, t + T) = f(x, t)$ para alguna constante T , la ecuación se dice que es periódica, Si $f(x, t)$ es lineal en x la ecuación (2.1) se dice que es una ecuación lineal y el correspondiente sistema dinámico que esto representa es referido como un sistema lineal.

3.a Definición de Sistemas Dinámicos Lineales Invariantes en el tiempo

Un sistema dinámico de control viene descrito por un sistema de la forma

$$\begin{aligned} \dot{x} &= f(t, x, u), \quad t \in [t_0, \infty), \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m, \\ y &= \eta(t, x, u), \quad y \in \mathbb{R}^p \end{aligned}$$

donde $f(t, x, u)$ satisface condiciones suficientes para la existencia y unicidad de soluciones, por ejemplo las condiciones de Caratheodory. Aquí, u es la variable de control y aparece como parámetro en la ecuación diferencial. La variable observable es y ; cuando $y = x$, se dice que el estado es completamente observable. En general $p < n$.

Para que un sistema dinámico sea lineal debe cumplirse el siguiente teorema:

Se tiene un sistema que partiendo de un estado inicial $x_1(t_0)$, con una entrada $u_1(\tau)$, $t_0 \leq \tau \leq t$, responde con una salida $y_1(t)$, y a partir de un estado inicial $x_2(t_0)$ con una entrada $u_2(\tau)$, $t_0 \leq \tau \leq t$, responde con otra salida $y_2(t)$. Se dice que dicho sistema es lineal si para a y b reales, partiendo del estado inicial $x_3(t_0) = ax_1(t_0) + bx_2(t_0)$ con una entrada $u_3(\tau) = au_1(\tau) + bu_2(\tau)$, $t_0 \leq \tau \leq t$, la salida es $y_3(t) = ay_1(t) + by_2(t)$.

Esta propiedad de linealidad en sistemas diferenciales se traduce en que las funciones f y η

son lineales con respecto a x y a u .

$$f(t, \alpha x_1 + \beta x_2, \alpha u_1 + \beta u_2) = \alpha f(t, x_1, u_1) + \beta f(t, x_2, u_2)$$

$$\eta(t, \alpha x_1 + \beta x_2, \alpha u_1 + \beta u_2) = \alpha \eta(t, x_1, u_1) + \beta \eta(t, x_2, u_2),$$

donde f y η son funciones vectoriales, por lo que la propiedad de linealidad se verifica si y sólo si las ecuaciones del modelo de estado se pueden expresar de forma matricial como:

$$\left. \begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \right\} \dots\dots\dots \text{modelo de estado matricial}$$

donde:

$x(t)$ es el vector de estado, de dimensión n .

$u(t)$ es el vector de entradas, de dimensión m .

$y(t)$ es el vector de salida, de dimensión p .

$A(t)$ $n \times n$ es la matriz del sistema.

$B(t)$ $n \times m$ es la matriz de entradas.

$C(t)$ $p \times n$ es la matriz de salida.

$D(t)$ $p \times m$ es la matriz de entradas para las salidas (en la mayoría de los sistemas se anula)

Los sistemas dinámicos invariantes en el tiempo son aquellos donde las matrices A,B,C, D no dependen explícitamente del tiempo. Se cumple el siguiente teorema:

Un sistema, con un estado inicial dado por $x_0 = x(t_0)$, sometido a una entrada $u_1(\tau), t_0 \leq \tau \leq t$, y que produce como salida la señal $y_1(t)$, es invariante en el tiempo si $\forall T$, partiendo del mismo estado x_0 pero en el instante $t_0 + T$, excitado con una entrada $u_2(\tau + T) = u_1(\tau), t_0 \leq \tau \leq t$, responde con una salida que es $y_2(t + T) = y_1(t)$

3.b. La función de transferencia

Si el sistema cumple con las condiciones de linealidad e invarianza se puede establecer la función de transferencia la cual rige la relación entrada-salida la cual es única sea cual fuere el modelo de estado del sistema.

La expresión de un sistema LIT es:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

La función de transferencia se define como el cociente de las transformadas de Laplace de la entrada entre la salida, con condiciones iniciales nulas. Tomando transformadas de Laplace en las ecuaciones anteriores, tomando en cuenta que las condiciones iniciales son nulas, obtenemos

$$\begin{aligned}sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s)\end{aligned}$$

Despejando $X(s)$ de la primera y sustituyendo en la segunda ecuación, obtenemos

$$\begin{aligned}X(s) &= [sI - A]^{-1} BU(s) \\ Y(s) &= C[sI - A]^{-1} BU(s) + DU(s) = [C[sI - A]^{-1} B + D]U(s)\end{aligned}$$

de donde la función de transferencia es,

$$G(s) = C[sI - A]^{-1} B + D.$$

Polos del Sistema

Siendo $[sI - A]^{-1} = \frac{1}{\det[sI - A]} \text{Adj}[sI - A]^T$, se concluye que el polinomio característico del sistema

es: $p(s) = \det[sI - A]$ por lo que los polos del sistema coinciden con los valores propios de la matriz A . Se ve también que como el polinomio característico sólo depende de A , también ha de ser así con la estabilidad del sistema. La posición de los ceros del sistema viene dada por las matrices A , B , C y D .

3.c. Obtención del Modelo de Estado

Conforme a las ecuaciones de estado y de salida del sistema lineal invariante:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

se pueden deducir las siguientes condiciones que habrá que tener en cuenta cuando se elige un modelo de estado:

- En la ecuación de estado sólo pueden estar relacionadas las variables de estado sus primeras derivadas y las entradas.
- En la ecuación de salida solo pueden estar relacionadas las variables de estado, las entradas y las salidas.

- Las variables de estado no pueden presentar discontinuidades, aunque la entrada al sistema si pueda tenerlas, pues en tal caso la derivada de la variable de estado no estaría definida.
- Se admiten discontinuidades en las entradas (como por ejemplo una entrada de escalón), por lo cual nunca pueden elegirse las entradas como variables de estado del sistema.

Representación en variables de estado de sistemas monovariantes

La expresión genérica de un sistema monovariante:

$$(s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0)y = (b_ms^m + \dots + b_1s + b_0)u \dots\dots\dots(a)$$

puede reescribirse de la siguiente forma como cociente de dos polinomios en el operador derivada s :

$$y = \frac{b_ms^m + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}u \dots\dots\dots(b)$$

El procedimiento consiste en elegir como primera variable de estado x_1 :

$$x_1 = \frac{1}{s^n + a_{n-1}s^{n-1} + \dots + a_0}u$$

lo que quiere decir que x_1 es una solución de la ecuación diferencial del primer término igualada a U . Se eligen las restantes variables por sucesiva derivación:

$$\begin{aligned} x_2 &= \dot{x}_1 \\ x_3 &= \dot{x}_2 \\ &\dots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= s^n x_1 = -a_{n-1}s^{n-1}x_1 - a_{n-2}s^{n-2}x_1 - \dots - a_1sx_1 - a_0x_1 + u \\ &= -a_{n-1}x_n - a_{n-2}x_{n-1} - \dots - a_1x_2 - a_0x_1 + u \end{aligned}$$

Así, el sistema en forma matricial resulta:

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & -a_3 & \dots & -a_{n-1} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u$$

La ecuación de salida queda:

$$y = (b_ms^m + \dots + b_1s + b_0)x_1$$

pasando esta expresión a notación matricial se obtiene:

$$y = (b_0 \quad b_1 \quad b_2 \quad \cdots \quad b_m \quad 0 \quad \cdots \quad 0)x$$

Variables de Jordan

Partiendo de la ecuación (b), la representación del sistema en variables de Jordan admite la siguiente forma:

$$y = \left(b_n + \frac{\rho_1}{s - \lambda_1} + \frac{\rho_2}{s - \lambda_2} + \cdots + \frac{\rho_n}{s - \lambda_n} \right) u$$

Se le asigna una variable a cada uno de estos operadores:

$$x_1 = \frac{1}{s - \lambda_1} u \quad \dot{x}_1 = \lambda_1 x_1 + u$$

$$x_2 = \frac{1}{s - \lambda_2} u \quad \dot{x}_2 = \lambda_2 x_2 + u$$

$$x_3 = \frac{1}{s - \lambda_3} u \quad \dot{x}_3 = \lambda_3 x_3 + u$$

...

$$x_n = \frac{1}{s - \lambda_n} u \quad \dot{x}_n = \lambda_n x_n + u$$

Con lo que la ecuación viene representada por:

$$\dot{x} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} x + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u$$

La ecuación de salida se obtiene como:

$$y = (\rho_1 \quad \rho_2 \quad \cdots \quad \rho_n)x + b_n u$$

3.d. Solución de la Ecuación de Estado de Sistemas Lineales

Matriz de Transición. Solución a la ecuación homogénea

El objetivo es solucionar la ecuación de estado:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

En la ecuación homogénea se supone que la entrada $u(t)$ es nula, por lo que la ecuación a resolver es de la forma:

$$\dot{x}(t) = A(t)x(t)$$

Para encontrar la solución de la ecuación homogénea se puede utilizar el método de integración por aproximaciones sucesivas de Peano-Baker, que dice que dada una ecuación de la forma

$$\dot{x}(t) = f(x(t), u(t), t)$$

con condiciones iniciales $x(t_0) = x_0$ se puede obtener la solución de dicha ecuación construyendo una secuencia de funciones del tipo:

$$\varphi = x_0$$

$$\varphi_k(t) = \varphi_0 + \int_{t_0}^t f(\varphi_{k-1}(\tau), \dots(\tau), \tau) d\tau$$

en la que la solución de la ecuación diferencial se obtiene como:

$$x(t) = \lim_{k \rightarrow \infty} \varphi_k(t)$$

$$\varphi_1(t) = \varphi_0 + \int_{t_0}^t A(\tau)x_0 d\tau = x_0 + \int_{t_0}^t A(\tau)d\tau x_0$$

$$\varphi_2(t) = \varphi_0 + \int_{t_0}^t A(\tau) \left(x_0 + \int_{t_0}^{\tau} A(\tau_1)d\tau_1 x_0 \right) d\tau =$$

$$= x_0 + \int_{t_0}^t A(\tau)d\tau x_0 + \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1)d\tau_1 d\tau x_0 =$$

$$= \left(I + \int_{t_0}^t A(\tau)d\tau + \int_{t_0}^t A(\tau) \int_{t_0}^{\tau} A(\tau_1)d\tau_1 d\tau \right) x_0$$

a medida que progresa la construcción de los sucesivos términos de la serie, se comprueba que en todos ellos va apareciendo el término x_0 que se puede extraer por la derecha; después de la extracción la expresión tiende a ser:

$$x(t) = \Phi(t, t_0) x_0$$

donde $\Phi(t, t_0)$ es la llamada matriz de transición, mediante la que se construye la solución de la ecuación diferencial

$$\Phi(t, t_0) = e^{\int_{t_0}^t A(\tau) d\tau}$$

Cuando la matriz A es invariante la matriz conmuta con su integral de forma que se puede aplicar la expresión general:

$$\Phi(t, t_0) = e^{\int_{t_0}^t A d\tau} = e^{A \int_{t_0}^t d\tau} = e^{A(t-t_0)}$$

Cálculo de la matriz de transición para el caso LIT

Cuando la matriz A es invariante se puede calcular la matriz de transición por el método de Jordan, esta matriz tiene la forma:

$$\Phi(t, t_0) = e^{A(t-t_0)}$$

El método de Jordan consiste en realizar una transformación:

$$x = T\tilde{x} \quad \Rightarrow \quad \tilde{x} = T^{-1}x$$

Tal que la matriz del sistema sea también transformada según:

$$\tilde{A} = T^{-1}AT$$

Así $\Phi(t, t_0) = T\tilde{\Phi}(t, t_0)T^{-1}$

De esta forma, según el método de Jordan, el cálculo de la exponencial de una matriz se realiza en los siguientes pasos:

- Cálculo de la matriz diagonal en bloques de Jordan, mediante el cálculo previo de las matrices T y T^{-1} . Para el cálculo de la matriz T esta se compone, por columnas de los vectores propios asociados a los valores propios de la matriz A .
- Cálculo de la exponencial de la matriz diagonal $\tilde{\Phi}(t, t_0)$ formada por los valores propios de la matriz A .
- Multiplicación de esta matriz por las matrices de transformación para devolver el sistema a su expresión original $\Phi(t, t_0)$.

Un ejemplo

Obtener el modelo de estado y la evolución de las variables de estado del sistema representado mediante la siguiente ecuación diferencial:

$$\ddot{y} + 6\dot{y} + 11y + 6y = 0$$

En primer lugar se han de elegir las variables de estado; en este caso; dado que la ecuación es de tercer orden, serán necesarias tres, siendo una posible elección:

$$x_1 = y \quad x_2 = \dot{y} \quad x_3 = \ddot{y}$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = -6x_3 - 11x_2 - 6x_1$$

con lo que el modelo de estado expresado en forma matricial queda:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

La reducción a la forma canónica de Jordan es

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ -3 & -2 & -1 \\ 9 & 4 & 1 \end{pmatrix} \begin{pmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ -3 & -2 & -1 \\ 9 & 4 & 1 \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ -3 & -2 & -1 \\ 9 & 4 & 1 \end{pmatrix} \begin{pmatrix} -3 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 3/2 & 1/2 \\ -3 & -4 & -1 \\ 3 & 5/2 & 1/2 \end{pmatrix}$$

$$\exp(tA) = \begin{pmatrix} 1 & 1 & 1 \\ -3 & -2 & -1 \\ 9 & 4 & 1 \end{pmatrix} \begin{pmatrix} e^{-3t} & 0 & 0 \\ 0 & e^{-2t} & 0 \\ 0 & 0 & e^{-t} \end{pmatrix} \begin{pmatrix} 1 & 3/2 & 1/2 \\ -3 & -4 & -1 \\ 3 & 5/2 & 1/2 \end{pmatrix}$$

$$\begin{pmatrix} 2 + e^{-3t} - 3e^{-2t} + 3e^{-t} & \frac{1}{2} e^{-3t} (3 - 8e^t + 5e^{2t}) & \frac{1}{2} e^{-3t} (-1 + e^t)^2 \\ -3e^{-3t} (1 - 2e^t + e^{2t} + 2e^{3t}) & -\frac{1}{2} e^{-3t} (-1 + e^t)^2 (9 + 2e^t) & -\frac{1}{2} e^{-3t} (3 - 4e^t + e^{2t}) \\ 14 + 9e^{-3t} - 12e^{-2t} + 3e^{-t} & \frac{1}{2} e^{-3t} (27 - 32e^t + 5e^{2t}) & \frac{1}{2} e^{-3t} (9 - 8e^t + e^{2t} - 2e^{3t}) \end{pmatrix}$$

3.e. Controlabilidad y Observabilidad en Sistemas Lineales Invariantes

Sistema controlable

Se dice que un punto del espacio de estado de un sistema, x_1 , es controlable desde el estado x_0 en $[t_0, t_1]$ si existe una entrada u definida en el intervalo $[t_0, t_1]$ tal que transfiera el estado del sistema desde x_0 en t_0 hasta x_1 en t_1 .

Un sistema se dice controlable en $[t_0, t_1]$ si todos los puntos de su espacio de estado son controlables en $[t_0, t_1]$

Un sistema se dice controlable si todos los puntos de su espacio de estado son controlables

Controlabilidad en Sistemas LIT

En el caso de sistemas lineales invariantes, la controlabilidad se puede estudiar mediante el siguiente teorema:

Dado el sistema de dimensión n con ecuación de estado:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

es controlable si y sólo si la matriz de controlabilidad Q , definida de la siguiente forma:

$$Q = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$$

es de rango máximo, es decir n

El conjunto de puntos del espacio de estado alcanzables partiendo del estado inicial nulo forman un subespacio vectorial, al que se le denomina *subespacio controlable* generado por los vectores columna linealmente independientes de Q .

Controlabilidad de la salida

Se dice que un punto del espacio de salida de un sistema, y_1 , es controlable si para todo punto origen y para todo instante inicial t_0 , existe una entrada u definida en el intervalo $[t_0, t_1]$ con t_1 finito, que lleve la salida al valor y_1 en t_1 .

Los teoremas de controlabilidad se pueden generalizar como:

Dado el sistema

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t)$$

la salida es controlable en $[t_0, t_1]$ si y sólo si la matriz definida por:

$$W_c(t_1, t_0) = \int_{t_0}^{t_1} c(t_1)\phi(t_1, \tau)B(\tau)B^T(\tau)\phi^T(t_1, \tau)C^T(\tau)d\tau$$

es no singular.

Y para sistemas lineales invariantes

Dado el sistema

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t)$$

la salida $y(t)$ es controlable si y sólo si la matriz definida por

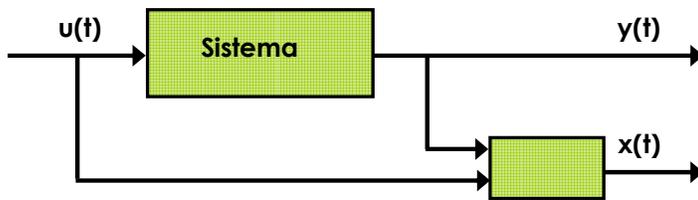
$$Q_c = \begin{bmatrix} CB & CAB & CA^2B & \dots & CA^{n-1}B \end{bmatrix}$$

es de rango p (notar que $Q_c = CQ$)

Sistema observable

La idea de observabilidad se relaciona con la posibilidad de conocer el valor del estado de un sistema a partir del conocimiento de la evolución de la entrada y de la salida que genera.

Una vez conocido el estado en un instante inicial, se puede determinar el estado en cualquier otro instante posterior utilizando la solución de la ecuación de estado. La observabilidad se presenta conceptualmente como una idea complementaria a la controlabilidad; si la controlabilidad estudia la relación entrada-estado, la observabilidad ve la relación estado-salida.



OBSERVABILIDAD

Observabilidad en un instante t_0

Se dice que un punto x_0 del espacio de estado es observable en $[t_0, t_1]$ si siendo éste el estado inicial en el instante t_0 , $x_0 = x(t_0)$, el conocimiento de la entrada $u(t)$ en el intervalo $[t_0, t_1]$ y de la salida $y(t)$ en el mismo intervalo permite determinar que el estado inicial del sistema en el instante t_0 es x_0 .

Observabilidad para un punto

Se dice que un punto del espacio de estado, x_0 , es observable si siendo éste el estado inicial para un instante inicial t_0 cualquiera, existe un intervalo de tiempo finito $[t_0, t_1]$ tal que el conocimiento de la entrada $u(t)$ en el intervalo $[t_0, t_1]$ y de la salida $y(t)$ en dicho intervalo permite determinar que el estado inicial es x_0 .

Sistema observable

Un sistema es observable si todos los puntos del espacio de estado son observables

Si fuera conocido $x(t_0)$ y $u(t_0)$ se podría obtener $x(t)$ con la expresión de la solución completa

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau$$

y teniendo en cuenta la ecuación de la salida:

$$y(t) = C(t)x(t) + D(t)u(t);$$

sustituyendo en esta ecuación la solución de la ecuación de estado:

$$y(t) = C(t)\Phi(t, t_0)x(t_0) + \int_{t_0}^t C(t)\Phi(t, \tau)B(\tau)u(\tau)d\tau + D(t)u(t);$$

agrupando los términos que dependen de la entrada y la salida, que se suponen conocidas se obtiene:

$$\dot{y}(t) = y(t) - \int_{t_0}^t C(t)\Phi(t, \tau)B(\tau)u(\tau)d\tau + D(t)u(t) = C(t)\Phi(t, t_0)x(t_0)$$

donde $\dot{y}(t)$ representa la componente de salida debida sólo a la existencia de condiciones iniciales no nulas, es decir, la aportación a la salida dada por la evolución libre del sistema, o lo que es lo mismo la salida del sistema ante entrada nula.

De esta forma se observa que la posibilidad de conocer el estado a partir de la salida y la entrada depende de $C(t)$ y $\Phi(t, t_0)$ que a su vez depende únicamente de $A(t)$.

En el caso particular de que la matriz $C(t)$ es cuadrada e invertible, se puede despejar $x(t_0)$ de la ecuación anterior:

$$x(t_0) = \Phi^{-1}(t, t_0)C^{-1}\dot{y}(t)$$

o lo que es lo mismo:

$$\Phi(t, t_0)x(t_0) = x(t) = C^{-1}\dot{y}(t)$$

En este caso la salida representa un cambio de base de las variables de estado mediante la matriz C y por lo tanto basta conocer la salida en un solo instante para conocer el estado del sistema.

Dado un sistema definido por las ecuaciones:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

$$y(t) = C(t)x(t) + D(t)u(t)$$

es observable en $[t_0, t_1]$ si y sólo si la matriz $V(t_1, t_0)$ definida como:

$$V(t_1, t_0) = \int_{t_0}^{t_1} \Phi^T(\tau, t_0)C^T(\tau)C(\tau)\Phi(\tau, t_0)d\tau$$

es no singular.

Estados no-observables

El hecho de que exista un valor del estado inicial no nulo, tal que al evolucionar el sistema a partir de él sin entrada la salida sea nula, hace que el sistema no sea observable, porque si se tiene que un sistema partiendo de un estado inicial x_1 da una salida $y_1(t)$, si existe algún punto x_2 tal que al evolucionar a partir de él la salida es nula, para $x_3 = x_1 + x_2$ la salida por linealidad, volverá a ser $y_1(t)$, por lo que no se podrá distinguir el estado inicial y no se tendrá ningún punto del sistema observable.

Observabilidad para Sistemas Lineales Invariantes

Dado un sistema de dimensión n definido por las ecuaciones:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

es observable si y sólo si la matriz de observabilidad P definida por:

$$P = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{n-1} \end{bmatrix} \text{ es de rango máximo, es decir } n$$

3.f. Estabilidad en sistemas LIT

La estabilidad de sistemas lineales esta bien estudiada, y puede ser considerada el precursor de casi todos los resultados en la estabilidad de sistemas no-lineales. En esta sección se suman algunos de los resultados bien conocidos pertenecientes a los sistemas lineales.

Consideremos un sistema lineal descrito por las ecuaciones diferenciales siguientes:

$$\dot{x}(t) = A(t)x(t) \quad x(t_0) = x_0 \dots\dots\dots (3.1)$$

donde $x : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ y $A : \mathbb{R}^+ \rightarrow \mathbb{R}^{n \times n}$. Los elementos de $A(t)$ se asume que están acotados y ser tramos continuos para toda $t \in \mathbb{R}^+$. Entonces la ecuación (3.1) tiene una solución única que puede ser expresada como:

$$x(t) = \Phi(t, t_0)x_0,$$

para cualquier condición inicial x_0 donde $\Phi(t, t_0)$ es la matriz de transición de la Ecuación (3.1) que satisface la ecuación diferencial

$$\Phi(t, t_0) = A(t)\Phi(t, t_0) \quad \Phi(t_0, t_0) = I.$$

Cada columna de $\Phi(t, t_0)$ representa una solución linealmente independiente de la ecuación diferencial (3.1) y cualquier solución de la posterior puede ser expresada como una combinación lineal de las columnas de $\Phi(t, t_0)$. El desarrollo de las soluciones de la Ecuación (3.1) pueden ser agrupadas como sigue:

Teorema 3.f.1

- (i) *Todas las soluciones de la Ecuación 3.1 están acotadas si y sólo si $\|\Phi(t, t_0)\| \leq k$ donde k es una constante positiva.*
- (ii) *Todas las soluciones de la Ecuación 3.1 tienden a cero cuando $t \rightarrow \infty$, si y sólo si $\|\Phi(t, t_0)\| \rightarrow 0$ cuando $t \rightarrow \infty$.*

Para sistemas generales de tiempo variante, la determinación de la forma analítica de la matriz de transición es una tarea difícil y por lo tanto, las condiciones del Teorema 3.f.1 son difíciles de verificar. Para el caso especial cuando el sistema es de tiempo invariante, las condiciones analíticas del Teorema 3.f.1 pueden transformarse en condiciones algebraicas.

Cuando el sistema es lineal y de tiempo invariante, descrito por la ecuación diferencial

$$\dot{x}(t) = Ax(t) \dots\dots\dots (3.2)$$

donde A es una matriz constante, la matriz de transición $\Phi(t, t_0)$ esta dada por

$$\Phi(t, t_0) = e^{A(t-t_0)},$$

Utilizando la transformación de similaridad $A = T \Lambda T^{-1}$ donde Λ es la forma canónica de Jordan de A , la matriz de transición y, por lo tanto las condiciones del teorema 3.f.1 puede ser evaluada usando la identidad

$$e^{(At)} = T e^{(\Lambda t)} T^{-1}$$

Teorema 3.f.2

- 1.- *Todas las soluciones de la Ecuación (3.2) son acotadas si y sólo si todos los valores propios de A tienen partes reales no-positivas y aquellos con partes reales cero son ceros simples del polinomio mínimo de A .*
- 2.- *Todas las soluciones de la Ecuación (3.2) tienden a cero cuando $t \rightarrow \infty$ si y solo si, todos los valores propios de A tienen partes reales negativas.*

Nos referiremos a matrices que satisfacen la condición 1 como matrices estables y aquellas que satisfacen la condición 2 como matrices asintóticamente estables.

Los valores propios de A puede ser determinados como las raíces de la ecuación característica

$$g(\lambda) \triangleq \det[\lambda I - A] = 0.$$

Si las raíces de $g(\lambda)$ permanecen en \mathbb{C}^- , $g(\lambda)$ decimos que es un polinomio de Hurwitz y las soluciones de la Ecuación (3.2) tienden a cero. Si las raíces satisfacen la condición 1 en el Teorema 3.f.2, $g(\lambda)$ se dice que es un polinomio estable. Estas condiciones pueden revisarse usando el criterio Routh-Hurwitz obtenido hace cien años. Esto muestra la importancia de los resultados de Routh y Hurwitz en la teoría de estabilidad de sistemas lineales invariantes. Ya que en general la matriz de transición $\Phi(t, t_0)$ de sistemas lineales de tiempo variante no puede ser expresada directamente en términos de la matriz $A(t)$, no hay condiciones algebraicas similares conocidas para sistemas de tiempo-variante.

4. ESTABILIDAD EN SISTEMAS NO-LINEALES

"Y basados en la convicción de que los sistemas adaptativos sólo pueden ser diseñados con confianza cuando sus propiedades de estabilidad global son bien entendidas".

Kumpati S. Narendra

4.a. Análisis por linealización

En el estudio de los sistemas dinámicos, la primera propiedad que es necesario conocer del sistema es cuando es estable, bajo que condiciones y cerca de que puntos de operación. Por otra parte para hacer posible la descripción de un sistema hay que simplificarlo, tomando solamente los factores más significativos. Además las ecuaciones raramente se pueden resolver en forma cerrada, sino que se suelen resolver de forma aproximada. Es importante el estudio de si las variaciones pequeñas en las condiciones iniciales hacen que cambien mucho el comportamiento del sistema.

Consideremos el sistema

$$\begin{cases} \dot{x} = f_1(x, y) \\ \dot{y} = f_2(x, y) \end{cases} \dots\dots\dots(4.a.1)$$

Los puntos de equilibrio del sistema son las soluciones del sistema $f_1(x, y) = f_2(x, y) = 0$. Los sistemas lineales pueden tener a lo sumo un punto singular (salvo los casos degenerados). Puesto que siempre

se puede hacer una traslación del sistema de coordenadas del plano fase, podemos considerar que el punto singular está en el origen.

Cuando el sistema es no-lineal se pueden linealizar las ecuaciones (4.a.1) en torno al origen para poder aproximar el sistema por uno lineal y clasificar los posibles puntos singulares. Supongamos que el sistema lineal viene dado por las siguientes ecuaciones.

$$\begin{cases} \dot{x} = ax + by \\ \dot{y} = cx + dy \end{cases}$$

de manera matricial
$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

La ecuación característica del sistema linealizado es $|\lambda I - A| = 0$, donde $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ es

$$D(\lambda) = \begin{vmatrix} \lambda - a & -b \\ -c & \lambda - d \end{vmatrix} = \lambda^2 - (a+b)\lambda + ad - bc = 0$$

llamando $p = -(a+d)$, $q = ad - bc$, se tiene $D(\lambda) = \lambda^2 + p\lambda + q = 0$

Las raíces de esta ecuación son los valores propios:

$$\lambda_1, \lambda_2 = \frac{-p \pm \sqrt{p^2 - 4q}}{2}$$

El tipo de punto singular del sistema lineal, depende de cómo son los valores propios del sistema. La solución de $\dot{x} = Ax$ para un estado inicial x_0 es:

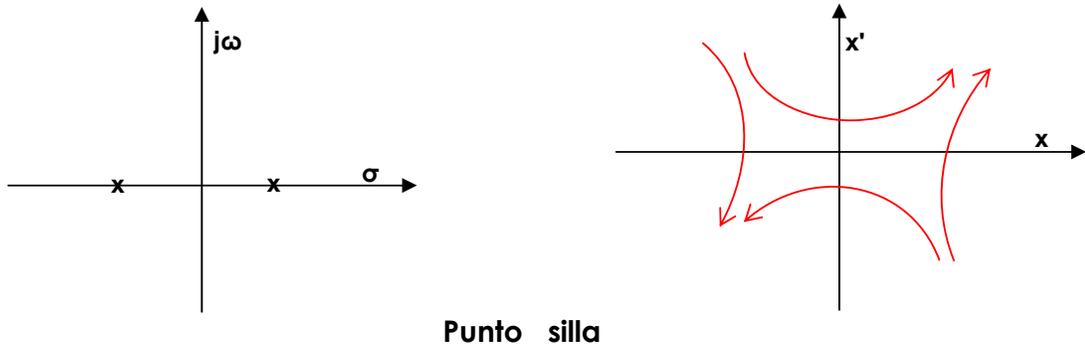
$$x(t) = P \exp(Jt) P^{-1} x_0$$

donde J es la forma canónica de Jordan de A y P es una matriz no singular de paso, tal que $A = P^{-1}AP$. Según los valores propios de A , la forma de Jordan puede tomar una de las siguientes formas:

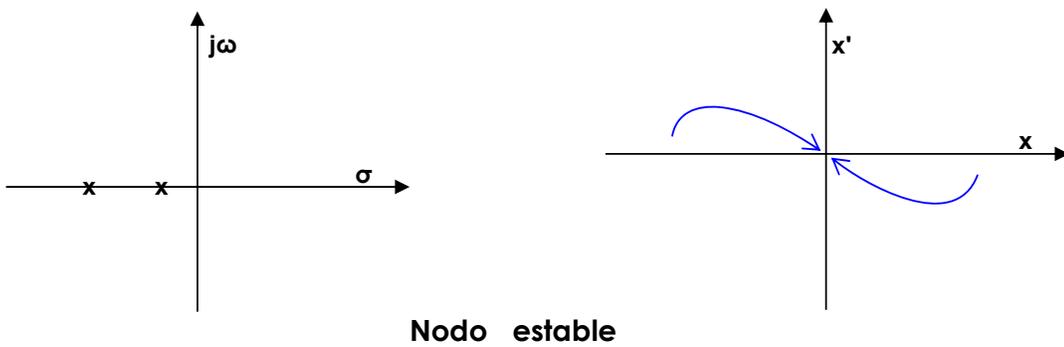
$$\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \begin{pmatrix} \lambda & k \\ 0 & \lambda \end{pmatrix}, \text{ y } \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix}$$

donde k puede ser 0 o 1. La primera forma corresponde al caso de valores propios reales y distintos, la segunda forma corresponde al caso de valores propios reales e iguales y la tercera forma corresponde al caso de valores propios complejos $\lambda_{1,2} = \alpha \pm \beta j$.

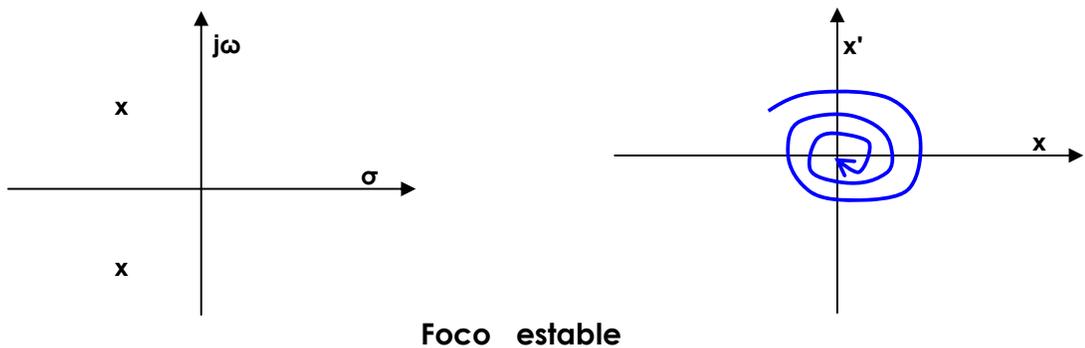
1. Si $q < 0$, los valores propios son reales y de signo distinto. El punto singular se denomina punto silla.



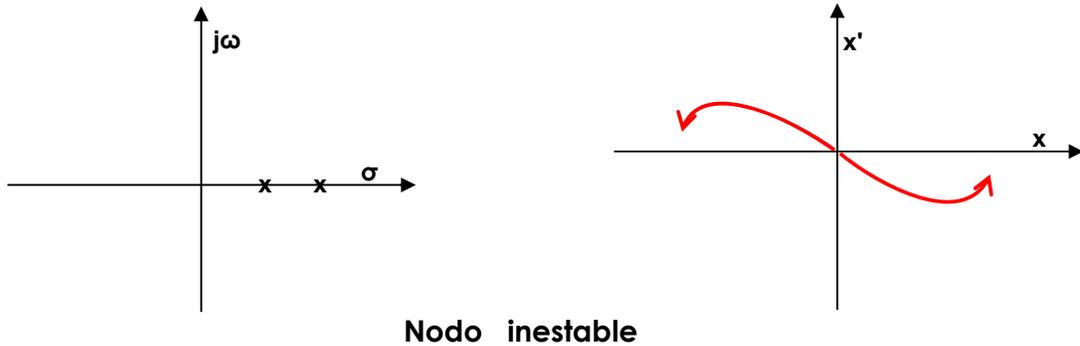
2. Si los valores propios λ son reales y negativos las soluciones son exponenciales decrecientes. Las trayectorias en el plano de fases convergen hacia el origen y el punto singular se denomina nodo estable.



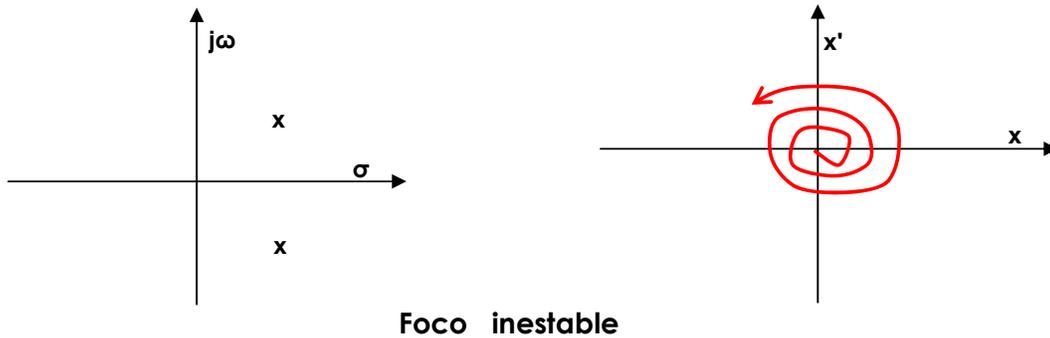
3. Si los valores propios λ son complejos conjugados con parte real negativa, las trayectorias son sinusoides compuestas con términos exponenciales decrecientes en forma de espirales que se van aproximando al origen. El punto singular se denomina foco estable.



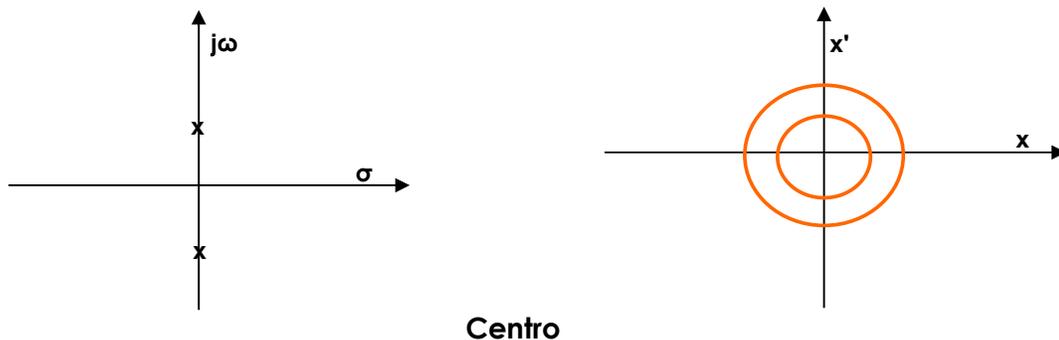
4. Si los valores propios son reales y positivos los modos son exponenciales crecientes y las trayectorias se alejan del origen. El punto singular se denomina nodo inestable.



5. Si los valores propios son complejos conjugados con parte real positiva, los modos son sinusoides compuestas con términos exponenciales crecientes y las trayectorias en el plano de fases son espirales que se van alejando del origen. El punto singular se denomina foco inestable.



6. Si los valores propios son imaginarios puros, los modos son sinusoides y las trayectorias en el plano fase son circunferencias con centro en el origen. El punto singular se denomina centro.



7. Si uno de los valores propios es cero, el plano de fase es degenerado. La matriz A tiene un subespacio nulo (kernel) no nulo y cualquier vector del subespacio nulo será un punto de equilibrio, es decir, el sistema tiene un subespacio de equilibrio de dimensión uno, en vez de un punto de equilibrio.

8. Si ambos valores propios son cero la matriz A es la matriz cero y el subespacio nulo es de dimensión dos, es decir todos los puntos del plano son puntos de equilibrio. En este caso las trayectorias están formadas por un punto.

4.b. Análisis de la estabilidad por el método de Lyapunov

4.b.1. Definiciones y teoremas en el Método Directo

El método de Lyapunov está pensado para sistemas modelados en variables de estado. Un punto de equilibrio o punto crítico es un punto en el que \dot{x} es cero en ausencia de entradas y perturbaciones. Por tanto si el sistema es puesto en ese estado permanece indefinidamente en él, y es un punto de equilibrio. Un punto crítico es estable cuando las trayectorias que nacen en puntos próximos al punto de equilibrio, permanecen uniformemente próximas a dicho punto en los tiempos posteriores.

Si se produce una pequeña perturbación, el sistema puede volver al punto de partida y se dice que es estable. El sistema puede alejarse de ese punto y el punto se llama inestable. Si el sistema no se acerca ni se aleja, el punto se llama neutralmente estable.

En el método de Lyapunov se consideran sistemas del tipo

$$\dot{x} = f(x, t)$$

donde x es el vector de estado de dimensión n y f es una función lineal o no-lineal que depende de x y de t . Tienen especial interés los sistemas denominados autónomos o sistemas que no varían en el tiempo.

$$\dot{x} = f(x)$$

Para todos estos tipos de sistemas el punto de equilibrio se considera el origen, ya que haciendo el cambio de variables $\bar{x}_i = x_i - c_i$ se consigue trasladar el punto de equilibrio al origen. Asimismo, se supone que no hay otra solución constante en el entorno del origen, por tanto, es un punto de equilibrio aislado.

La Estabilidad de Lyapunov considera la estabilidad como una propiedad interna del sistema, y trata con el efecto de perturbaciones en las condiciones iniciales. Primero debemos asegurarnos la existencia de soluciones para todas las condiciones iniciales (x_0, t_0) donde $x_0 \in \mathbb{R}^n$ y $t_0 \in \mathbb{R}^+$.

Sea un sistema descrito por la ecuación diferencial no-lineal

$$\dot{x} = f(x, t) \quad f(0, t) = 0, \forall t \geq t_0 \quad \dots\dots\dots (4.b.1)$$

donde $x(t_0) = x_0$ y $f : \mathbb{R}^+ \rightarrow \mathbb{R}^n$ es tal que la solución $x(t; x_0, t_0)$ existe para toda $t \geq t_0$. Ya que $f(0, t) \equiv 0$, esto implica que el origen es un estado de equilibrio. Las siguientes definiciones pertinentes para algunas de nociones básicas en estabilidad de Lyapunov en tal estado de equilibrio.

Definición L.1 “Estado de equilibrio estable”

El estado de equilibrio $x=0$ de la ecuación (4.b.1) se dice que es estable si para cada $\varepsilon > 0$ y $t_0 \geq 0$, existe un $\delta(\varepsilon, t_0) > 0$ tal que $\|x_0\| < \delta$ implica que $\|x(t; x_0, t_0)\| < \varepsilon, \forall t \geq t_0$ (Ver figura 4.1.a)

Esencialmente la Definición L.1 establece que resultan pequeñas perturbaciones en pequeñas desviaciones del estado de equilibrio o mas precisamente que podemos mantener la trayectoria cercana al origen por empezar suficientemente cercana a este.

Definición L.2 “Estado de equilibrio atractor”

El estado de equilibrio $x=0$ de la Ecuación (4.b.1) se dice que es atractor si para algún $\rho > 0$ y cada η y $t_0 > 0$ existe un número $T(\eta, x_0, t_0)$ tal que $\|x_0\| < \rho$ implica que $\|x(t; x_0, t_0)\| < \eta$ para toda $t \geq t_0 + T$.

Atractividad implica que todas las trayectorias que empiezan en una vecindad del origen eventualmente se aproximan al origen. Si el vector inicial x_0 tiene una norma menos que ρ , entonces para todo tiempo t mas grande que $t_0 + T$, la trayectoria esta a menos de una distancia η lejos del origen. Uno puede estar tentado a concluir que un sistema atractor debe ser estable también. Pero atractividad y estabilidad muestran ser conceptos diferentes. Por ejemplo un sistema autónomo puede ser atractor pero inestable.

Definición L.3 “Asintóticamente estable”

El estado de equilibrio $x=0$ de la ecuación diferencial (4.b.1) se dice ser asintóticamente estable si es estable y atractor (ver figura c).

En este caso la trayectoria tiende a el origen cuando $t \rightarrow \infty$ y al mismo tiempo permanece cerca del origen si empieza suficientemente cercano a este.

Las propiedades de estabilidad de muchos sistemas no dependen del tiempo inicial t_0 . Sistemas autónomos y algunos periódicos son ejemplos típicos. Esto motiva las definiciones de estabilidad uniforme y estabilidad asintótica uniforme.

Definición L.4 “Uniformemente estable”

El estado de equilibrio $x = 0$ de la ecuación diferencial (4.b.1) se dice ser uniformemente estable si en la Definición L.1, δ es independiente del tiempo inicial t_0 (ver figura d).

Definición L.5 “Uniformemente asintóticamente Estable ”

El estado de equilibrio $x = 0$ de la ecuación (4.b.1) es uniformemente asintóticamente estable (U.A.S.), si es estable uniformemente para algún $\varepsilon_1 > 0$ y para cada $\varepsilon_2 > 0$ existe $T(\varepsilon_1, \varepsilon_2) > 0$ tal que si $\|x_0\| < \varepsilon_1$, entonces $\|x(t; x_0, t_0)\| < \varepsilon_2$ para toda $t \geq t_0 + T$ (ver figura e).

La importancia de la estabilidad asintótica y uniforme en el análisis de los sistemas dinámicos se volverá clara en las siguientes secciones.

Definición L.6 “Exponencialmente estable”

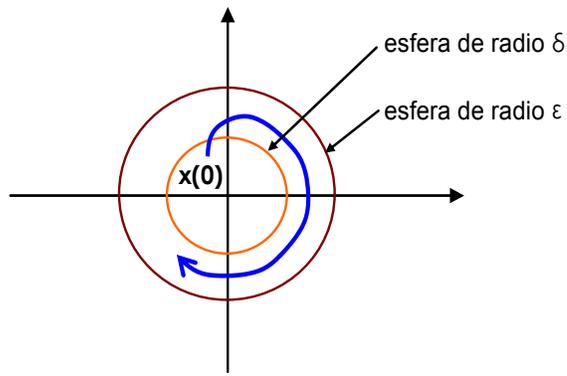
El estado de equilibrio $x = 0$ de la ecuación (4.b.1) es exponencialmente estable si existen constantes $a > 0$ y $b > 0$ tales que $\|x(t; x_0, t_0)\| \leq a \cdot e^{(-b(t-t_0))} \|x_0\|$, $\forall t \geq t_0$ para toda t_0 y en una vecindad B del origen.

Definición L.7 “Estabilidad en el sentido de Lyapunov”

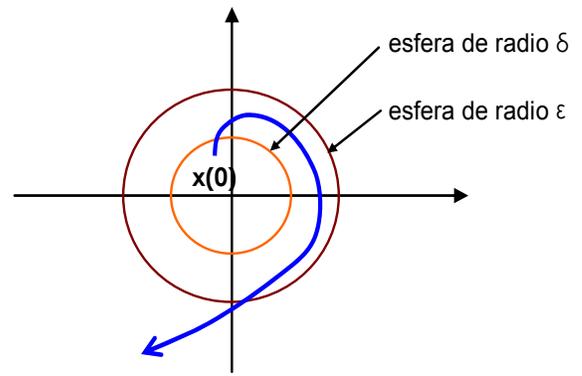
El punto de equilibrio $x = 0$ se denomina estable si para todo $\varepsilon > 0$, existe $\delta > 0$ tal que si $\|x(0)\| < \delta$ entonces $\|x(t)\| < \varepsilon$ para todo $t > 0$. En caso contrario el punto de equilibrio es inestable (ver figuras a y b).

Definición L.8 “Estabilidad asintótica”

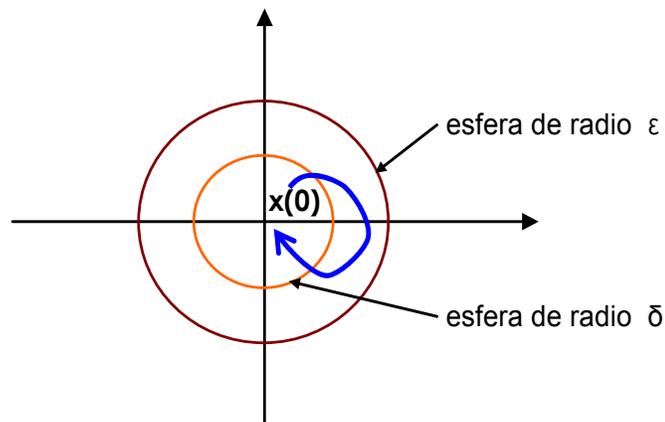
El punto de equilibrio 0 es asintóticamente estable si es estable y además existe $\delta > 0$ tal que si $\|x(0)\| < \delta$, entonces $\|x(t)\| \rightarrow 0$ cuando $t \rightarrow \infty$



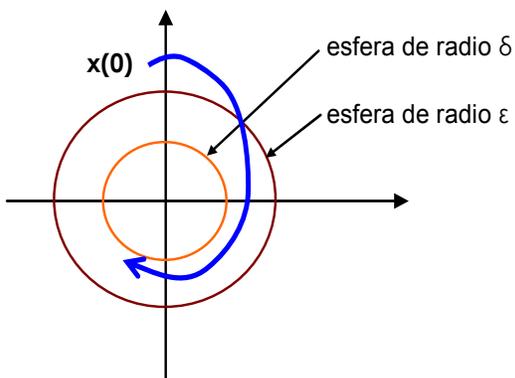
a) Punto de equilibrio estable



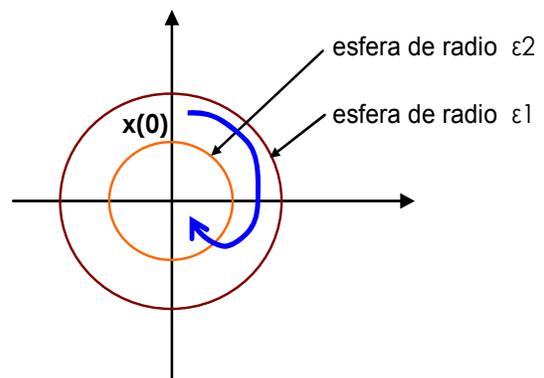
b) Punto de equilibrio inestable



c) Punto de equilibrio Estable Asintóticamente



d) Punto de equilibrio Estable Uniformemente



e) Punto de equilibrio Estable Asintóticamente y Uniformemente

figura 4.1

Una función $L(x)$ es definida negativa (positiva) en un entorno $S_\varepsilon(0)$ del origen si $L(x) < 0$ ($L(x) > 0$), $\forall x \in S_\varepsilon(0)$ con $x \neq 0$, y $L(0) = 0$. La función $L(x)$ es semidefinida negativa (positiva) en un entorno $S_\varepsilon(0)$ del origen si $L(x) \leq 0$ ($L(x) \geq 0$), $\forall x \in S_\varepsilon(0)$ con $x \neq 0$, y $L(0) = 0$.

Estabilidad asintótica implica que la solución tiende al equilibrio cuando $t \rightarrow \infty$ y estabilidad asintótica uniforme implica que la convergencia de la solución es independiente del tiempo inicial. Si $\lim_{\varepsilon \rightarrow \infty} \delta(\varepsilon) = \infty$ en las definiciones dadas, entonces se dice que la estabilidad se mantiene en lo amplio o que el estado de equilibrio es globalmente estable. Si la Definición L.5 se mantiene para cada $\varepsilon_1 > 0$, en la ecuación (4.b.1) se dice que el estado de equilibrio es global, asintóticamente y uniformemente estable (U.A.G.S.). Estabilidad exponencial de un sistema implica siempre estabilidad asintótica uniforme. Lo inverso en general no es cierto. En el caso de sistemas lineales, sin embargo, estabilidad asintótica uniforme es equivalente a exponencialmente estable. También se mantienen todas las propiedades de globalmente estable para sistemas lineales. Si el sistema es autónomo todas las propiedades de estabilidad son uniformes. Si $B = \mathbb{R}^n$ en la Definición L.6 entonces el estado de equilibrio es exponencialmente y globalmente estable.

El método directo de Lyapunov es capaz de determinar la estabilidad de un punto equilibrio sin la determinación explícita de la solución de la Ecuación (4.b.1). El método ha probado ser efectivo en el análisis de estabilidad de ecuaciones diferenciales no-lineales cuyas soluciones son generalmente muy difíciles de determinar. Esto involucra encontrar una adecuada función escalar $L(x, t)$ y

examinar su derivada respecto del tiempo $\dot{L}(x_0, t) \stackrel{\Delta}{=} \frac{d}{dt} L(x(t, x_0), t) \Big|_{t=0}$ a lo largo de las trayectorias

del sistema. El razonamiento detrás de este método es que en un sistema puramente disipativo, la energía almacenada en el sistema es siempre positiva y cuya derivada de tiempo es no-positiva. En teoría de estabilidad, los teoremas de Lyapunov para estabilidad están establecidos separadamente en sistemas autónomos y no-autónomos y son establecidas restricciones sobre las funciones $L(x, t)$ y $\dot{L}(x, t)$ para asegurar las distintas clases de estabilidad definidas anteriormente.

Teorema 4.b.(I) "Estabilidad uniforme y global; sistema autónomo"

Sea un sistema autónomo el descrito por la ecuación

$$\dot{x} = f(x), \dots\dots\dots (4.b.2)$$

donde $f(0) = 0$. El punto de equilibrio $x = 0$ es asintótica uniforme y globalmente estable si existe una función escalar $L(x)$ con la derivada parcial continua con respecto a x , tal que $L(0) = 0$, y (i) $L(x) > 0$ para toda $x \neq 0$, (ii) $\dot{L}(x) < 0$ para toda $x \neq 0$, y (iii) $L(x) \rightarrow \infty$ cuando $\|x\| \rightarrow \infty$.

Teorema 4.b.(II) “Estabilidad Asintótica Uniforme y Global ”

Sea un sistema el descrito por la ecuación

$$\dot{x} = f(x, t) \quad f(0, t) = 0, \forall t \geq t_0 \dots\dots\dots (4.b.1)$$

El estado de equilibrio de la ecuación (4.b.1) es estable asintóticamente, uniformemente y globalmente si existe una función escalar $L(x, t)$ con primeras derivadas parciales continuas con respecto a x y a t tal que $L(0, t) = 0$ y si las siguientes condiciones son satisfechas:

- (i) $L(x, t)$ **es definida positiva**, esto es, existe una función escalar no decreciente continua α tal que $\alpha(0) = 0$ y $L(x, t) \geq \alpha(\|x\|) > 0$, para toda t y toda $x \neq 0$;
- (ii) $L(x, t)$ **es decreciente**, esto es, existe una función escalar no decreciente continua β tal que $\beta(0) = 0$ y $\beta(\|x\|) \geq L(x, t)$ para toda t ;
- (iii) $\dot{L}(x, t)$ **es definida negativa**, esto es,

$$\dot{L}(x, t) \stackrel{\Delta}{=} \frac{\partial L}{\partial t} + (\nabla L)^T f(x, t) \leq -\gamma(\|x\|) < 0$$

Donde γ es una función escalar no decreciente continua con $\gamma(0) = 0$, ∇L es el gradiente de L con respecto a x , y la derivada de tiempo de L es evaluada a lo largo de la trayectoria de la ecuación diferencial (4.b.1).

- (iv) $L(x, t)$ **es no acotada radialmente**, esto es, $\alpha(\|x\|) \rightarrow \infty$ cuando $\|x\| \rightarrow \infty$.

Demostración

Una idea general de la demostración esta dada en dos estados:

1. Estabilidad Uniforme

De (i) y de (ii) tenemos, $\alpha(\|x\|) \leq L(x, t) \leq \beta(\|x\|)$

Para cualquier $\varepsilon > 0$, existe una $\delta > 0$ tal que $\beta(\delta) < \alpha(\varepsilon)$. Sean las condiciones iniciales escogidas de tal manera que $\|x_0\| < \delta$. Entonces

$$\begin{aligned}\alpha(\varepsilon) &> \beta(\delta) \geq L(x_0, t_0) \\ &\geq L(x(t; x_0, t_0), t)\end{aligned}$$

ya que $\dot{L}(x, t) \leq 0$ por la condición (iii), por lo tanto

$$\alpha(\varepsilon) \geq \alpha(\|x(t; x_0, t_0)\|).$$

Ya que $\alpha(\cdot)$ es no decreciente, esto implica que $\|x(t; x_0, t_0)\| < \varepsilon \quad \forall t \geq t_0$, y por lo tanto, el origen de la ecuación (4.b.1) es **Estable Uniformemente**. (ver gráfico 1)

2. Estabilidad Asintóticamente Uniforme

Para establecer la estabilidad asintótica uniforme, se necesita mostrar que dado cualquier ε_2 , existe un $T(\varepsilon_1, \varepsilon_2) > 0$ tal que

$$\|x_0\| < \varepsilon_1 \Rightarrow \|x(t)\| < \varepsilon_2 \quad \forall t \geq t_0 + T$$

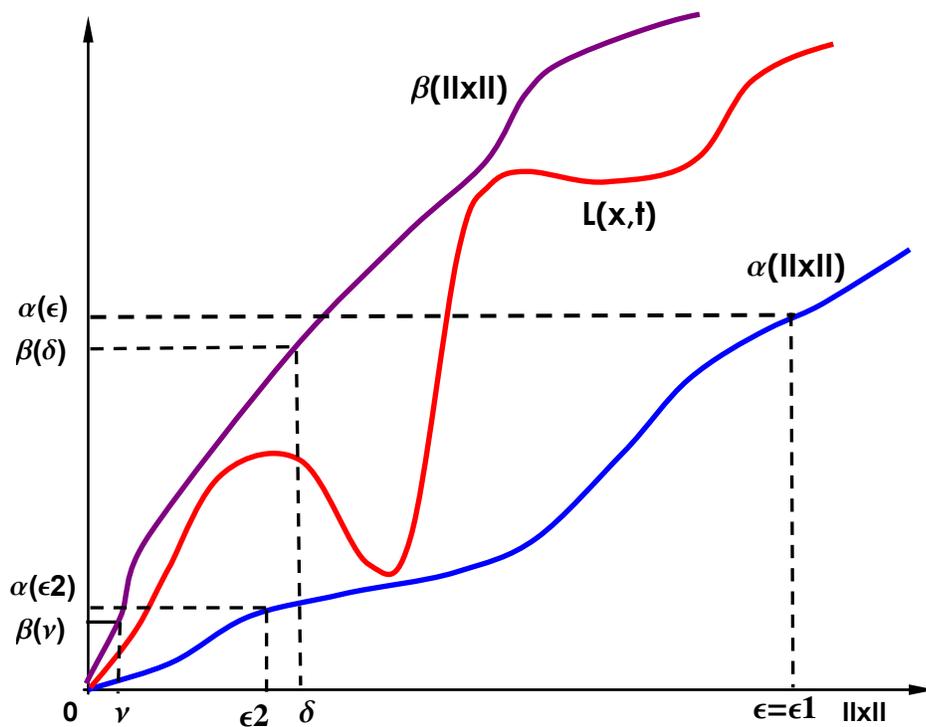
Sea δ y ε_1 tales que $\beta(\delta) < \alpha(\varepsilon_1)$ y, por lo tanto, por (1), si $\|x_0\| < \delta$, entonces $\|x(t; x_0, t_0)\| < \varepsilon_1$ para toda $t \geq t_0$. Si escogemos $\varepsilon_2 < \delta$ y $\nu > 0$ tal que $\beta(\nu) < \alpha(\varepsilon_2)$, entonces si $\|x(t; x_0, t_0)\| < \nu$ para algún tiempo $t = t_0 + T$, entonces $\|x(t; x_0, t_0)\| < \varepsilon_2$ para toda $t \geq t_0 + T$. La existencia de tal T se muestra para contradicción. Sea $\|x(t)\|$ estar en un intervalo compacto $[\nu, \varepsilon_1]$ para toda $t \geq t_0$. La función continua $\gamma(\|x\|)$ de la condición (iii) del Teorema 4.b.1 asume un valor mínimo $\gamma_0 > 0$ en este

intervalo compacto. Definiendo $T = \frac{\beta(\delta)}{\gamma_0}$, entonces existe un tiempo $t_2 = t_0 + T$ en el cual

$$\begin{aligned}0 < \alpha(\nu) &\leq L(x(t_2; x_0, t_0), t_2) \\ &\leq L(x_0, t_0) - T \cdot \gamma_0\end{aligned}$$

$\leq \beta(\delta) - \beta(\delta) = 0$, lo cual es una contradicción. Por lo tanto el origen de la ecuación (4.b.1) es UAS estable asintóticamente y uniformemente. Ya que por la condición (iv), $\alpha(\cdot)$ es no acotada radialmente, una ε puede ser encontrada tal que $\beta(\delta) < \alpha(\varepsilon)$ para cualquier δ . En suma, δ puede hacerse arbitrariamente grande. Por lo tanto, $x = 0$ en la ecuación (4.b.1) es Estable Asintóticamente, Uniformemente y Globalmente. (ver grafico 1)

Si $\dot{L}(x, t) \leq 0$ y la condición (i) es satisfecha, el origen de la ecuación (4.b.1) es Estable; si en adición a esto la condición (ii) es satisfecha, se desprende Estabilidad Uniforme y $L(x, t)$ es referida como **Función de Lyapunov**.



Estabilidad Uniforme y Estabilidad Asintótica Uniforme
Grafico 1

Comentario

En las definiciones de estabilidad anteriores se ha asumido tácitamente que las soluciones de las ecuaciones diferenciales para unas condiciones iniciales dadas pueden ser extendidas para todo $t \geq t_0$. Una condición suficiente para esto es que $f(x,t)$ satisfaga una condición de Lipschitz, esto es,

$$\|f(x_1,t) - f(x_2,t)\| \leq k \|x_1 - x_2\|$$

Para toda $(x_1,t), (x_2,t)$ en M . Sin embargo en muchos casos esto no puede ser verificado. En vez de esto asumimos que $f \in F$ así que la solución existe en un intervalo $[t_0 - \alpha_1, t_0 + \alpha_1]$ alrededor de t_0 y se usa la existencia de una función de Lyapunov para asegurar que $x(t; x_0, t_0)$ pertenece a un conjunto compacto. Esto asegura que la solución puede ser extendida para toda $t \geq t_0$.

Sin perder generalidad, no hay una regla explícita para encontrar la función de Lyapunov que satisfaga las condiciones del Teorema 4.b.(I) excepto en el caso de sistemas LTI . Si el origen de la Ecuación (4.b.1) es UAS . un importante Teorema de Massera nos asegura la existencia de la función de Lyapunov.

Hay teoremas que aseguran la existencia de funciones de Lyapunov para sistemas lineales. Esto, por consiguiente, puede ser modificado adecuadamente para aplicar a clases específicas de sistemas no-lineales.

Con condiciones $L(x,t) > 0, \dot{L}(x,t) \leq 0$

La condición (ii) en el Teorema 4.b.a, requiere que $\dot{L}(x,t)$ sea evaluada a lo largo de cualquier trayectoria que sea negativa-definida, esto es particularmente riguroso de satisfacer. En la práctica es a menudo considerablemente mas simple construir una función de Lyapunov $L(x,t)$ con una derivada de tiempo semidefinida-negativa que una que sea definida-negativa. Mas relevante para nuestros propósitos es el hecho que la condicion (ii) puede no ser nunca satisfecha para todo el espacio de estado para sistemas adaptativos. Por lo tanto, lo mejor que uno puede esperar es hacer $\dot{L}(x,t)$ semidefinida-negativa. Uno de los problemas fundamentales en el análisis de estabilidad de sistemas adaptativos consecuentemente es la determinación del funcionamiento asintótico de las soluciones de ecuaciones diferenciales gobernado el sistema por un conocimiento de la función de Lyapunov $L(x,t)$ cuya derivada de tiempo es semidefinida-negativa.

El problema descrito ha sido discutido con gran amplitud en la literatura de estabilidad, y para ciertas clases de sistemas LaSalle ha demostrado que la estabilidad asintótica uniforme puede ser concluida.

La Estabilidad Asintótica es una propiedad fuerte que es muy difícil de alcanzar en la práctica en sistemas dinámicos. Sin embargo, se tienen algunas definiciones suaves que son más útiles.

Teorema 4.b.(III) “Uniformemente SISL ”

Un punto de equilibrio x_e es estable en el sentido de Lyapunov (SISL) en t_0 . Si para cada $\varepsilon > 0$ existe un $\delta(\varepsilon, t_0) > 0$ tal que $\|x(t_0) - x_e\| < \delta(\varepsilon, t_0)$ implica que $\|x(t) - x_e\| < \varepsilon$ para $t \geq t_0$. La estabilidad se dice que es uniforme (es decir, uniformemente SISL). Si $\delta(\cdot)$ es independientes de t_0 . Esto es, el sistema es SISL para todo t_0 .

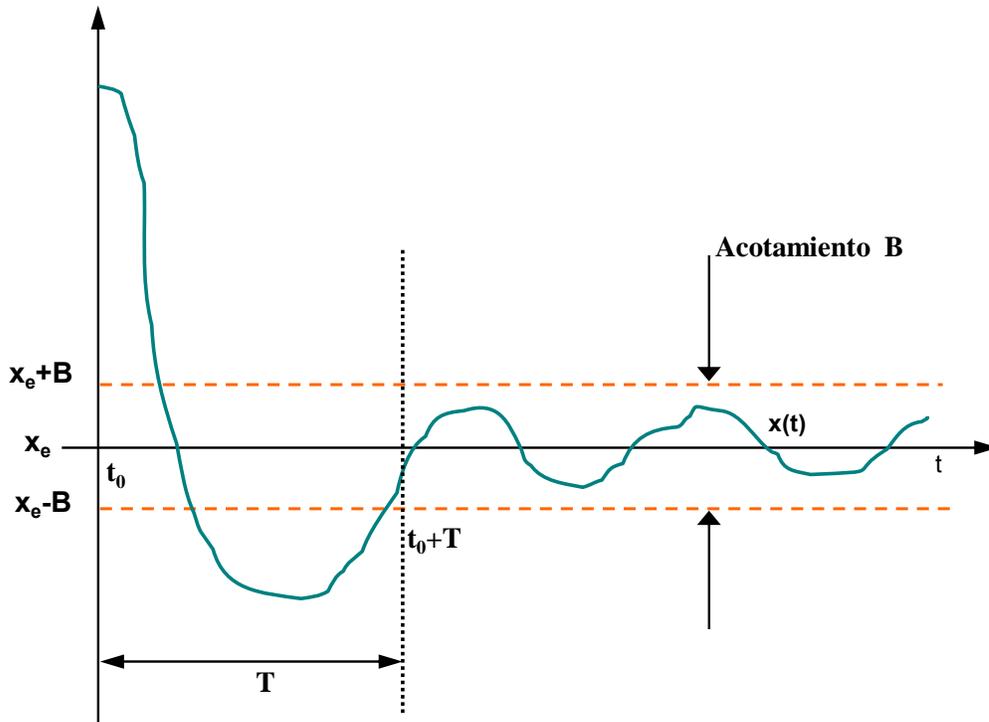
Hay que notar que SISL requiere la habilidad de guardar el estado $x(t)$ cercano arbitrariamente a x_e por empezar suficientemente cercano a este. Este aún es un requerimiento muy fuerte para el análisis de de estabilidad de sistemas prácticos en presencia de perturbaciones. Sin embargo lo vamos a resolver en la siguiente definición.

Esta definición esta relacionada con el Grafico 2 siguiente.

Teorema 4.b.(IV) “ Acotado Uniformemente y Últimamente ”

El punto de equilibrio x_e es acotado uniformemente y últimamente (UUB) si existe un conjunto compacto $S \subset \mathbb{R}^n$ tal que para cada $x(t_0) \in S$ existe un rango acotado B y un tiempo $T(B, x(t_0))$ tal que $\|x(t) - x_e\| < B$ para todo $t \geq t_0 + T$.

El intento aquí es capturar la noción que para todos los estados iniciales en el conjunto compacto S , la trayectoria del sistema eventualmente alcanza después de un tiempo T una vecindad acotada de x_e .



ACOTAMIENTO UNIFORME Y ULTIMO

Gráfico 2

La diferencia entre Acotamiento Uniforme y Ultimo UUB y *SISL* es que el acotamiento B no puede hacerse arbitrariamente pequeño por empezar cercano a x_e . En sistemas prácticos, en acotamiento B depende de perturbaciones y otros factores. Sin embargo un buen diseño de controlador debe garantizar que B sea bastante pequeño. El término uniforme indica que B no depende de t_0 . El término uniforme indica que el acotamiento se mantiene después de un cierto lapso T . Si $S = \mathbb{R}^n$, el sistema se dice que es globalmente *UUB*, es decir *GUUB*.

Hay que notar que si la función $f(\cdot)$ en (4.b.1) no depende explícitamente del tiempo t la estabilidad o acotamiento siempre se mantiene uniforme. La no-uniformidad sólo es problema con problemas que dependen del tiempo.

Los sistemas industriales son generalmente no-lineales, un problema que es exacerbado por la presencia de no-linealidades del actuador tales como fricción, zonas muertas y backlash. Las estrictas tolerancias modernas en la fabricación de semiconductores de integración a gran escala, el maquinado de alta precisión de partes para sistemas aeroespaciales y otros campos ponen severos

requerimientos en posicionamiento, velocidad y precisión de sistemas industriales mecánicos. El diseño de sistemas de control para sistemas no-lineales no es fácil. Teorema de Estabilidad de Lyapunov.

La función de Lyapunov es una función de la energía generalizada. El siguiente resultado importante muestra que puede ser usada para analizar estabilidad del sistema (2.1.9). Ahora veremos como usar este resultado para diseñar sistemas de control.

Teorema 4.b.(V) “ Estable en el Sentido de Lyapunov ”

Si existe una función de Lyapunov para el sistema (4.b.1), entonces el punto de equilibrio es *SISL* .

El siguiente teorema proporciona un fuerte resultado si la derivada de Lyapunov es negativa definida.

Teorema 4.b.(VI) “ Estabilidad de Asintótica ”

Si existe una función de Lyapunov para el sistema (4.b.1) con el fortalecimiento de la condición $\dot{L}(x) < 0$ para toda $x(t)$ diferente de cero, entonces el punto de equilibrio es *AS* .

Si la derivada de la función de Lyapunov es definida negativa entonces el sistema es *GAS* . Si la derivada de la función de Lyapunov es semidefinida negativa entonces el sistema es sólo *SISL* ; esto es, el estado interno es acotado pero no va a cero con el tiempo.

4.b.2. Ejemplos del Método Directo

Para tener una matriz de una función de Lyapunov, en las formas cuadráticas se puede aplicar el criterio de Sylvester, que dice que la condición necesaria y suficiente para que la forma cuadrática

$$L(x) = (x_1, x_2, \dots, x_n) \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$$

(donde la matriz es real y simétrica), sea definida positiva es que sus menores principales sean positivos:

$$a_{11} > 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \dots, \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} > 0 .$$

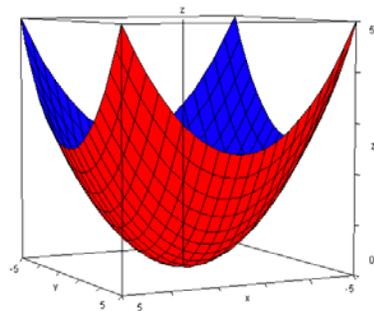
$L(x)$ es definida negativa si

$$a_{11} < 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \dots, (-1)^n \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} > 0$$

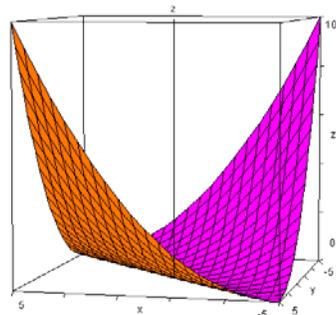
$L(x)$ es semidefinida positiva si la matriz es singular y todos sus menores principales son no negativos.

$L(x)$ es semidefinida negativa si $(-1)^k \cdot (\text{det. principal de orden } k) \geq 0$

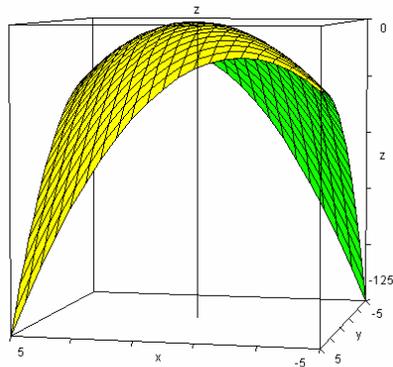
Ejemplos:



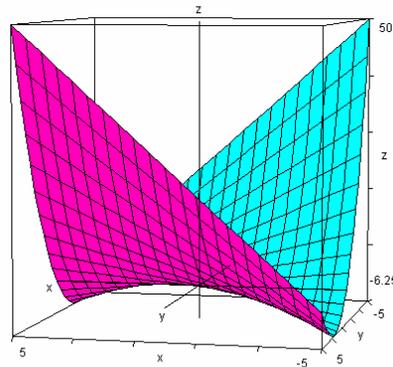
$L(x) = x_1^2 + x_2^2$ es definida positiva ($\forall x \neq 0, L(x) > 0, L(0) = 0$)



$L(x) = (x_1 + x_2)^2$ es semidefinida positiva (con $x_1 = -x_2 \neq 0 \rightarrow L(x) = 0$)



$L(x) = -x_1^2 - (x_1 + x_2)^2$ es definida negativa ($\forall x \neq 0, L(x) < 0, L(0) = 0$)



$L(x) = x_1x_2 + x_2^2$ es indefinida

Ejemplo 1

Estudiar la estabilidad del sistema

$$\dot{x}_1 = x_2 - x_1(x_1^2 + x_2^2)$$

$$\dot{x}_2 = -x_1 - x_2(x_1^2 + x_2^2)$$

Se encuentran los puntos de equilibrio $\dot{x} = 0$

$$\begin{aligned} 0 &= x_2 - x_1(x_1^2 + x_2^2) & 0 &= x_2^2 - x_1x_2(x_1^2 + x_2^2) \\ 0 &= -x_1 - x_2(x_1^2 + x_2^2) & 0 &= +x_1^2 + x_1x_2(x_1^2 + x_2^2) \end{aligned} \Rightarrow$$

Por lo cual, $0 = x_1^2 + x_2^2$ y sustituyendo en el primer sistema queda

$$x_1 = 0$$

$$x_2 = 0$$

Es decir el único punto de equilibrio es el origen

Considerando la función de Lyapunov $L(x) = x_1^2 + x_2^2$, que es definida positiva

Ahora se verificará si su derivada considerando las ecuaciones del sistema es definida negativa

$$\begin{aligned}\dot{L}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1(x_2 - x_1(x_1^2 + x_2^2)) + 2x_2(-x_1 - x_2(x_1^2 + x_2^2))\end{aligned}$$

Es decir, $\dot{L}(x) = -2(x_1^2 + x_2^2)^2$ es definida negativa. Por lo tanto, el sistema es asintóticamente

estable. Como se verifica que $L(x) = x_1^2 + x_2^2 \rightarrow \infty$, cuando $\|x\|_2 = \sqrt{x_1^2 + x_2^2} \rightarrow \infty$, entonces el sistema es global y asintóticamente estable.

Ejemplo 2

Estudiar la estabilidad del sistema

$$\dot{x}_1 = x_1x_2^2 - x_1$$

$$\dot{x}_2 = -x_1^2x_2$$

Considerando la función de Lyapunov $L(x) = \frac{1}{2}(x_1^2 + x_2^2)$, que es definida positiva. Ahora se verificará si su derivada es definida negativa:

$$\begin{aligned}\dot{L}(x) &= x_1(x_1x_2^2 - x_1) + x_2(-x_1^2x_2) \\ \dot{L}(x) &= -x_1^2\end{aligned}$$

Es sólo semidefinida negativa ya que puede ser cero cuando $x_2 \neq 0$. Por lo tanto el sistema es sólo SISL, esto es, el estado interno es acotado pero no va a cero con el tiempo.

Ejemplo 3

Estudiar la estabilidad del sistema

$$\dot{x}_1 = x_2 + x_1(x_1^2 + x_2^2)$$

$$\dot{x}_2 = -x_1 + x_2(x_1^2 + x_2^2)$$

Si se linealiza el sistema en el origen, se obtiene

$$\begin{aligned}\frac{\partial \dot{x}_1}{\partial x_1} &= 3x_1^2 + x_2^2 \Big|_{(0,0)} = 0 & \frac{\partial \dot{x}_1}{\partial x_2} &= 1 + 2x_1x_2 \Big|_{(0,0)} = 1 \\ \frac{\partial \dot{x}_2}{\partial x_1} &= -1 + 2x_1x_2 \Big|_{(0,0)} = -1 & \frac{\partial \dot{x}_2}{\partial x_2} &= x_1^2 + 3x_2^2 \Big|_{(0,0)} = 0\end{aligned}$$

es decir $\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ y por tanto el origen es un centro. Si tomamos como función de

Lyapunov $L(x) = x_1^2 + x_2^2$, como su derivada es:

$$\begin{aligned} L(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1(x_2 + x_1(x_1^2 + x_2^2)) + 2x_2(-x_1 + x_2(x_1^2 + x_2^2)) \\ &= 2x_1^4 + 4x_1^2x_2^2 + 2x_2^4 \end{aligned}$$

se obtiene que $L(x)$ y $\dot{L}(x)$ son definidas positivas lo cual nos dice el teorema que el sistema es inestable.

El criterio de Lyapunov es aplicable a sistemas no-lineales, pero su verificación, en general, no es nada fácil. Puede ser que no se encuentre una función de Lyapunov, pero esto no indica que dicha función no exista.

4.b.3. Extensión de Lyapunov. Lema de Barbalat

En algunas situaciones, los teoremas anteriores proporcionan muy poca información y sin embargo llegan a pedir mucho de los sistemas industriales y de defensa que tengan perturbaciones y deficiencias de actuadores. Algunas extensiones de la teoría de Lyapunov están dirigidas a resultados prácticos que puedan ser usados en los actuales sistemas de control industrial.

El primer resultado está basado en el Lema de Barbalat.

Dado un subconjunto $S \subset \mathbb{R}^n$, una función $f(x): S \rightarrow \mathbb{R}^m$ es continua en S si para cada $x_0 \in S$ y para cada $\varepsilon > 0$ existe un $\delta(\varepsilon, x_0) > 0$ tal que $\|x - x_0\| < \delta(\varepsilon, x_0)$ implica que $\|f(x) - f(x_0)\| < \varepsilon$.

Si $\delta(\cdot)$ es independiente de x_0 entonces se dice que $f(x)$ es uniformemente continua. Continuidad uniforme a menudo es difícil de probar. Sin embargo, si $f(x)$ es continua y su derivada $\frac{\partial f}{\partial x}$ es acotada entonces $f(x)$ es continua uniformemente. Esta prueba es sólo suficiente pero es fácil de usar.

Teorema 4.b.(VII) Lema de Barbalat, Extensión de Lyapunov

Sea $L(x)$ una función de Lyapunov. Si $\dot{L}(x)$ es continua uniformemente, entonces $\dot{L}(x) \rightarrow 0$ cuando $t \rightarrow \infty$.

Este teorema permite obtener mas información acerca del funcionamiento del sistema que la proporcionada por el Teorema 4.b.(V). Se dice que si $\dot{L}(x)$ tiene una propiedad extra, entonces uno

puede concluir que $\dot{L}(x) \rightarrow 0$. Esto puede decirnos a menudo que ciertos estados en el sistema no son simplemente acotados ya que en realidad van a cero. Para verificar la continuidad uniforme de $\dot{L}(x)$, sólo es necesario revisar que $\ddot{L}(x)$ está acotada.

Ejemplo 1 (Estabilidad Asintótica de los componentes del estado usando el lema de Barbalat)

Consideremos el sistema

$$\begin{aligned} \dot{x}_1 &= x_1 x_2^2 - x_1 \\ \dot{x}_2 &= -x_1^2 x_2 \end{aligned}$$

Con una función de Lyapunov candidata $L(x) = \frac{1}{2}(x_1^2 + x_2^2)$, la cual tiene

$$\dot{L}(x) = -x_1^2$$

Ya que esta es semidefinida negativa, podemos concluir que el sistema es *SISL*. Usando el Lema de Barbalat, podemos extraer mas información acerca de la estabilidad de esta función de Lyapunov. En efecto

$$\ddot{L}(x) = -2x_1\dot{x}_1 = -2x_1^2 x_2^2 + 2x_1^2$$

El análisis de Lyapunov muestra *SISL*. Lo cual muestra que $\|x\|$ está acotada. Esto a su vez muestra que $\dot{L}(x)$ está acotada, de aquí que $\dot{L}(x)$ es continua uniformemente. Por lo tanto $\dot{L}(x)$ en realidad va a cero. Esto finalmente demuestra que $x_1(t) \rightarrow 0$. Así la extensión de Barbalat ha permitido probar que ciertos componentes del estado son Estables Asintóticamente *AS*.

Los siguientes resultados proporcionan una forma más leve de estabilidad que *SISL*, que es mas útil para el diseño de controladores en sistemas prácticos, los cuales normalmente tienen la forma

$$\dot{x} = f(x) + d(t) \dots\dots\dots(4.b.3)$$

Con $d(t)$ una perturbación desconocida pero acotada. El teorema dice cuando el acotamiento uniforme y último está garantizado.

Teorema 4.b.(VIII) "Acotamiento Uniforme y Ultimo por Extensión de Lyapunov"

Se supone que para el sistema (4.b.3) existe una función $L(x)$ con derivadas parciales continuas tales que para x en un conjunto compacto $S \subset R^n$.

$$L(x) \text{ es definida positiva.}$$

$$\dot{L}(x) \text{ es definida negativa para } \|x\| > R$$

Para algún $R > 0$ tal que la bola de radio R es contenida en S . Entonces el sistema es Acotada Uniformemente y Últimamente *UUB* y mas aún, $\|x(t)\|$ es acotada en una vecindad de la bola de radio R .

Ejemplo 2 (*UUB* Extensión de Lyapunov).

Considerar el sistema

$$\begin{aligned} \dot{x}_1 &= x_1 x_2^2 - x_1 (x_1^2 + x_2^2 - 2) \\ \dot{x}_2 &= -x_1^2 x_2 - x_2 (x_1^2 + x_2^2 - 2) \end{aligned}$$

Y seleccionando una candidata a función de Lyapunov $L(x) = \frac{1}{2}(x_1^2 + x_2^2)$. Esta tiene la derivada

$$\dot{L}(x) = -(x_1^2 + x_2^2)(x_1^2 + x_2^2 - 2)$$

La cual es negativa siempre que

$$\|x\|^2 = x_1^2 + x_2^2 > 2$$

Las técnicas normales de Lyapunov fallan aquí ya que $\dot{L}(x)$ ni siquiera es semidefinida negativa. Sin embargo, la extensión *UUB* muestra que el sistema es *UUB* con el estado restringido a la vecindad de una bola con radio $\|x\| = 2$.

4.c. Análisis Estabilidad Total y BIBO (entradas acotadas-salidas acotadas)

Es de gran interés la estabilidad global de los sistemas adaptativos y la principal aproximación usada para asegurar esto es el método directo de Lyapunov. En muchas situaciones, sin embargo, el método de Lyapunov no puede ser usado exclusivamente y uno tiene que usar otros conceptos que han sido tratados extensivamente en la literatura de estabilidad. Esto incluye la estabilidad BIBO la cual es mas apropiada cuando están presentes perturbaciones externas continuas y Estabilidad Total cuando las perturbaciones pueden ser consideradas ser dependientes del estado.

Cuando las perturbaciones externas e internas están presentes en el sistema, las ecuaciones para analizar tienen la siguiente forma no-lineal:

$$\dot{x}(t) = f(x,t) \quad f(0,t) = 0 \quad \forall t \geq t_0 \quad (S) \quad \dots\dots\dots (4.c.1)(a)$$

$$\dot{x}(t) = f(x,t) + g(x,t) \quad (S_p) \quad \dots\dots\dots (4.c.1)(b)$$

donde (S) representa un sistema no perturbado con $x = 0$ como el estado de equilibrio, y (S_p) es un sistema perturbado $g(x,t)$ el cual puede deberse a no-linealidades, variaciones de parámetros ó perturbaciones estructurales.

Definición 4.c.1

El estado de equilibrio $x=0$ de (S) es **totalmente estable** si para cada $\varepsilon > 0$, existen dos números positivos $\delta_1(\varepsilon)$ y $\delta_2(\varepsilon)$ tales que $\|x(t_0)\| < \delta_1$ y $\|g(x,t)\| < \delta_2$ implican que cada solución $x(t; x_0, t_0)$ de (S_p) , satisface la condición $\|x(t; x_0, t_0)\| < \varepsilon$

El siguiente teorema indica la relación entre estabilidad total y estabilidad asintótica uniforme en el sentido de Lyapunov.

Teorema 4.c.a

Si el estado de equilibrio de (S) es estable asintóticamente y uniformemente, entonces este es totalmente estable.

Del Teorema 4.c.a, se desprende para perturbaciones pequeñas suficientemente y condiciones iniciales que son, $\|u\| \leq \delta_2$ y $\|x_0\| \leq \delta_1$ donde δ_1, δ_2 son pequeñas suficientemente, las soluciones de la ecuación diferencial no-lineal

$$\dot{x} = f(x,t) + u(t) \quad x(t_0) = x_0 \quad \dots\dots\dots (4.c.2)$$

Serán acotadas uniformemente para toda $t \geq t_0$. Sin embargo, en la práctica, es de gran interés determinar la naturaleza de las soluciones, cuando algún margen es especificados sobre $\|u\|$.

La estabilidad BIBO (entradas acotadas – salidas acotadas) es considerablemente mas difícil de establecer en sistemas no-lineales de la forma de la Ecuación (4.c.2) que en sistemas lineales, donde tal estabilidad continua si el sistema no-forzado es *UASG* (o estable exponencialmente). Sin embargo, los análisis de sistemas adaptativos en la presencia de perturbaciones dirige invariablemente los cuestionamientos relacionados a Estabilidad *BIBO*.

Teorema 4.c.b

Considerar la ecuación diferencial

$$\dot{x} = f(x) \quad x: R^+ \rightarrow R^n \quad \dots\dots\dots (4.c.3)$$

Sea Ω una vecindad del origen acotada y sea Ω^c su complemento. Se supone que $L(x)$ es una función escalar con derivadas parciales continuas en Ω^c y satisface 1) $L(x) > 0, \forall x \in \Omega^c$, 2) $\dot{L}(x) \leq 0, \forall x \in \Omega^c$ y 3) $L(x) \rightarrow \infty$ cuando $\|x\| \rightarrow \infty$. Entonces cada solución de la Ecuación (4.c.3) esta acotada para toda $t \geq 0$.

5. CONTROL DE SISTEMAS ADAPTATIVOS

5.a. Problema de Identificación: caso escalar

El método directo de Lyapunov es la herramienta principal usada para la derivación de las leyes adaptativas estables. Este método puede ser usado para diseñar sistemáticamente una amplia clase de sistemas adaptativos estables. En los casos considerados la estabilidad global es asegurada por la existencia de una función de Lyapunov para el sistema entero, elegir las funciones necesarias es crucial para una adecuada ley adaptativa.

Consideremos el problema de Identificación de la dinámica de una planta con una entrada acotada u y una salida x_p descrita por una ecuación diferencial de primer orden:

$$\dot{x}_p(t) = a_p x_p(t) + k_p u(t)$$

donde los parámetros a_p y k_p son constantes pero son desconocidas. El estado de equilibrio de la planta no-forzada se asume que es estable asintóticamente así que $a_p < 0$. El problema de identificación entonces se reduce a la determinación de a_p y k_p del par observado $u(t)$ y $x_p(t)$ de entrada-salida.

Para realizar esto describimos dos Modelos de Estimación que tienen diferente estructura. Aunque el análisis de estabilidad de los dos es casi idéntico, solo el segundo modelo puede ser fácilmente extendido a sistemas de alto orden. La entrada u y la salida \hat{x}_p de los dos modelos están relacionadas por ecuaciones diferenciales (ver Graficos 5.1 y 5.2).

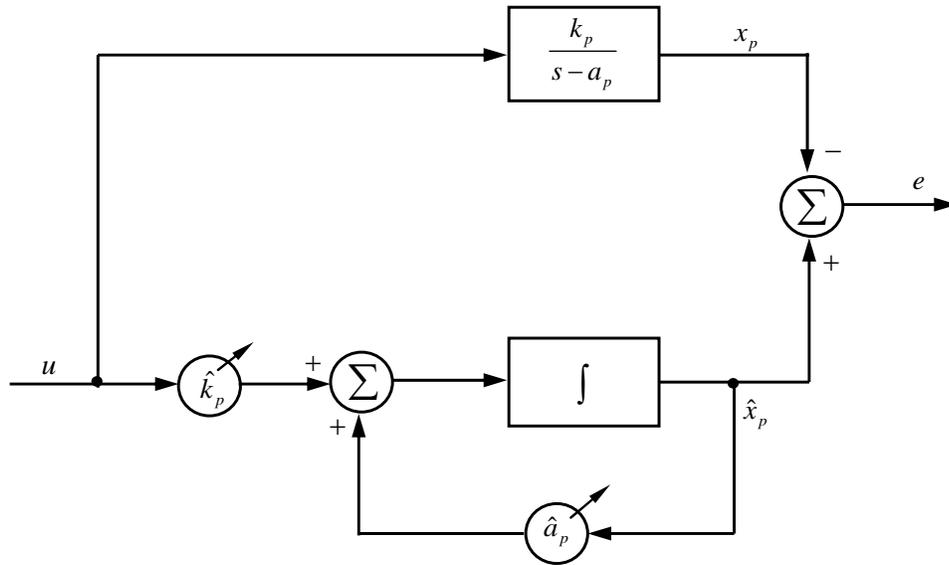
$$\dot{\hat{x}}_p = \hat{a}_p(t) \hat{x}_p + \hat{k}_p(t) u(t) \dots\dots\dots \text{Modelo 1 (5.0.a)}$$

$$\text{y } \dot{\hat{x}}_p = a_m \hat{x}_p + (\hat{a}_p(t) - a_m) x_p(t) + \hat{k}_p(t) u(t) \quad a_m < 0 \dots\dots\dots \text{Modelo 2 (5.0.b)}$$

$\hat{a}_p(t)$ y $\hat{k}_p(t)$ en los dos modelos representan respectivamente los estimados de a_p y k_p en el tiempo t . El objetivo es ajustar $\hat{a}_p(t)$ y $\hat{k}_p(t)$ de tal manera que

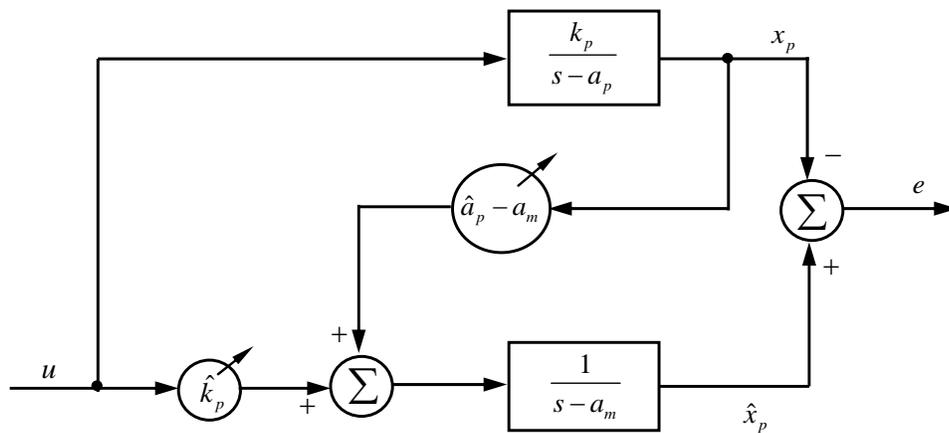
$$\lim_{t \rightarrow \infty} \hat{a}_p(t) = a_p \quad \text{y} \quad \lim_{t \rightarrow \infty} \hat{k}_p(t) = k_p$$

Aún cuando todas las señales en el sistema entero permanezcan acotadas. Además, esto tiene que ser realizado usando las derivadas de tiempo de las señales observadas ya que el ruido está presente invariablemente.



MODELO 1

Grafico 5.1



MODELO 2

Grafico 5.2

Definiendo el error entre la salida correcta $x_p(t)$ de la planta y la salida estimada $\hat{x}_p(t)$ como $e(t)$ y los parámetros del error entre $\hat{a}_p(t)$ y a_p y $\hat{k}_p(t)$ y k_p como $\phi(t)$ y $\psi(t)$ respectivamente, así tenemos las ecuaciones diferenciales del error:

$$\dot{e}(t) = a_p e(t) + \phi(t) \hat{x}_p(t) + \psi(t) u(t) \dots\dots\dots \text{Modelo 1} \quad (5.1.a)$$

$$\dot{e}(t) = a_m e(t) + \phi(t) x_p(t) + \psi(t) u(t) \dots\dots\dots \text{Modelo 2} \quad (5.1.b)$$

Con

$$e(t) \triangleq \hat{x}_p(t) - x_p(t) \quad \phi(t) \triangleq \hat{a}_p(t) - a_p \quad \psi(t) \triangleq \hat{k}_p(t) - k_p$$

En términos de las ecuaciones de error de los modelos el objetivo puede ser establecido como la determinación de una regla para el ajuste de $\hat{a}_p(t)$ y $\hat{k}_p(t)$ haciendo que los errores $\phi(t)$ y $\psi(t)$ tiendan a cero cuando $t \rightarrow \infty$. Es claro que si $u(t)$ y $x_p(t)$ están acotadas uniformemente y $\phi(t)$ y $\psi(t)$ tienden a cero, el error $e(t)$ también tiende a cero asintóticamente. De aquí que la convergencia del error de salida a cero necesariamente se obtendrá si el parámetro exacto es alcanzado. Ya que el parámetro del error nunca es conocido explícitamente, el éxito del procedimiento de identificación debe ser inferido en la práctica del desempeño del error de salida medido.

5.b. Leyes adaptativas

La ley adaptativa para el Modelo 2. La ecuación (5.1.b) representa el primer paso importante en la descripción matemática del proceso de identificación usando el Modelo 2. Indica como los errores de parámetro afectan el error de salida en el sistema. Ya que el identificador es un diferenciador libre, intentamos obtener leyes de identificación adaptativas de la forma:

$$\begin{aligned} \dot{\hat{a}}_p(t) &= f_1[x_p(t), u(t), \hat{x}_p(t), \hat{a}_p(t)] \\ \dot{\hat{k}}_p(t) &= f_2[x_p(t), u(t), \hat{x}_p(t), \hat{k}_p(t)] \end{aligned} \dots\dots\dots (5.2)$$

Así que el ajuste de los parámetros $\hat{a}_p(t)$ y $\hat{k}_p(t)$ están basados en todas las señales que pueden ser medidas directamente. Las funciones f_1 y f_2 han de ser escogidas de tal manera que las ecuaciones 5.1.b y 5.2 constituyan un sistema de tercer orden cuyos puntos de equilibrio estén dados por $\hat{a}_p = a_p$, $\hat{k}_p = k_p$ y $e = 0$ sea estable globalmente. Unas de esas reglas son:

$$\begin{aligned} \dot{\hat{a}}_p(t) &= -e(t)x_p(t) \\ \dot{\hat{k}}_p(t) &= -e(t)u(t) \end{aligned} \dots\dots\dots (5.3)$$

Ya que a_p y k_p son constantes, las leyes adaptativas anteriores pueden ser expresadas en términos de los errores de parámetro $\phi(t)$ y $\psi(t)$ como:

$$\begin{aligned} \dot{\phi}(t) &= -e(t)x_p(t) \\ \dot{\psi}(t) &= -e(t)u(t) \end{aligned} \dots\dots\dots (5.4)$$

En este caso el estado de equilibrio $e = \phi = \psi = 0$ puede mostrarse que es estable globalmente. La justificación para la elección de la leyes adaptativas (5.3) está basada en la candidata para función de Lyapunov

$$V(e, \phi, \psi) = \frac{1}{2} [e^2(t) + \phi^2(t) + \psi^2(t)] \dots\dots\dots (5.5)$$

$V(e, \phi, \psi)$ es una forma cuadrática de $e(t)$, $\phi(t)$ y $\psi(t)$ y es definida-positiva. Evaluando su derivada a lo largo de cualquier trayectoria de las ecuaciones diferenciales (5.1.b) y (5.4) se produce

$$\dot{V}(e, \phi, \psi) = a_m e^2(t) + \phi(t)e(t)x_p(t) + \psi(t)e(t)u(t) + \phi(t)\dot{\phi}(t) + \psi(t)\dot{\psi}(t) \dots \dots \dots \quad (5.6)$$

$\dot{\phi}(t)$ y $\dot{\psi}(t)$ en la ecuación (5.6) deben ser tales que cancelen los términos que contienen $\phi(t)$ y $\psi(t)$ así \dot{V} se vuelve semidefinida-negativa. Esto sugiere ley adaptativa en la ecuación (5.4) de la cual obtenemos

$$\dot{V}(e, \phi, \psi) = a_m e^2(t) \leq 0 \dots \dots \dots (5.7)$$

Por lo tanto, $V(e, \phi, \psi)$ es una función de Lyapunov y el origen es estable uniformemente y globalmente. Se desprende que $e(t)$, $\phi(t)$ y $\psi(t)$ están acotadas para todo $t \geq t_0$.

Se debe notar que ya que los errores de parámetro $\phi(t)$ y $\psi(t)$ en cualquier tiempo t no son conocidos (ya que a_p y k_p son desconocidos) la ley adaptativa no puede ser implantada como esta dada en la ecuación (5.4). Por lo tanto, mientras el análisis del sistema adaptativo puede ser llevado a cabo convenientemente usando las ecuaciones de error (5.1.b) y (5.4), la implantación práctica de la ley adaptativa tiene que ser llevada a cabo en términos de los parámetros $\hat{a}_p(t)$ y $\hat{k}_p(t)$ como está dado en la ecuación (5.3).

Ya que
$$-\int_{t_0}^{\infty} \dot{V}(e(\tau), \phi(\tau), \psi(\tau)) d\tau = V(t_0) - V(\infty) < \infty$$

tenemos que

$$0 \leq \int_{t_0}^{\infty} e^2(\tau) d\tau < \infty$$

O que $e \in \mathcal{L}^2$. Ya que $\dot{e}(t)$ como está dado por la ecuación (5.1.b), está acotada, se desprende del siguiente Lema

Si $f : \mathbb{R}^+ \rightarrow \mathbb{R}$ es continua uniformemente para $t \geq 0$, y si el límite de la integral $\lim_{t \rightarrow \infty} \int_0^t |f(\tau)| d\tau$ existe y es finito, entonces $\lim_{t \rightarrow \infty} f(t) = 0$.

Por lo tanto, en este caso $\lim_{t \rightarrow \infty} e(t) = 0$.

En tiempos iniciales del control adaptativo se podía concluir de las ecuaciones (5.5) y (5.7) que $\lim_{t \rightarrow \infty} e(t) = 0$. Sin embargo, como se ve en el análisis anterior, que sólo podemos concluir que $e \in \mathcal{L}^2$.

Ya que una función puede ser una cuadrática integrable sin tender a cero, es claro que condiciones adicionales son necesarias para que $e(t)$ tienda a cero asintóticamente.

5.c. Ganancias adaptativas

En la discusión anterior, las leyes adaptativas en la ecuación (5.4) aseguran la existencia de la función de Lyapunov de la forma $V = \frac{1}{2}[e^2 + \phi^2 + \psi^2]$. Si, por otra parte $V(e, \phi, \psi)$ es elegida para que tenga la forma

$$V = \frac{1}{2} \left[e^2 + \frac{1}{\gamma_1} \phi^2 + \frac{1}{\gamma_2} \psi^2 \right]$$

donde γ_1 y γ_2 son constantes positivas, las leyes adaptativas deben tener la forma:

$$\dot{\phi} = -\gamma_1 e x_p$$

$$\dot{\psi} = -\gamma_2 e u$$

Para hacer $V(e, \phi, \psi)$ una función de Lyapunov, γ_1 y γ_2 son referidas como *ganancias adaptativas*.

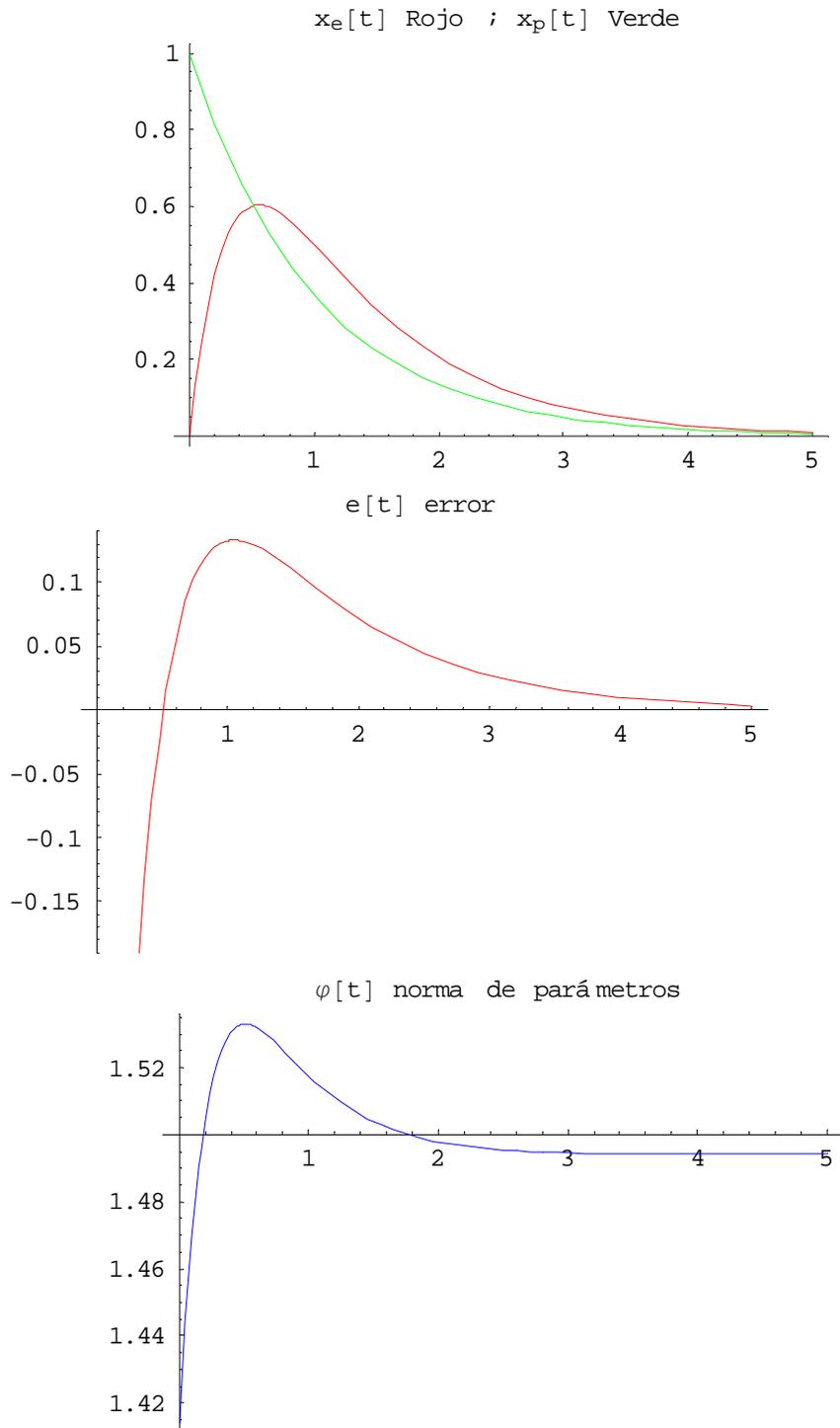
Ya que el sistema entero es estable globalmente para todos los valores positivos de γ_1 y γ_2 , las últimas pueden ser escogidas para optimizar el desempeño del sistema entero en algún sentido.

5.d. Simulación de identificación de una planta lineal

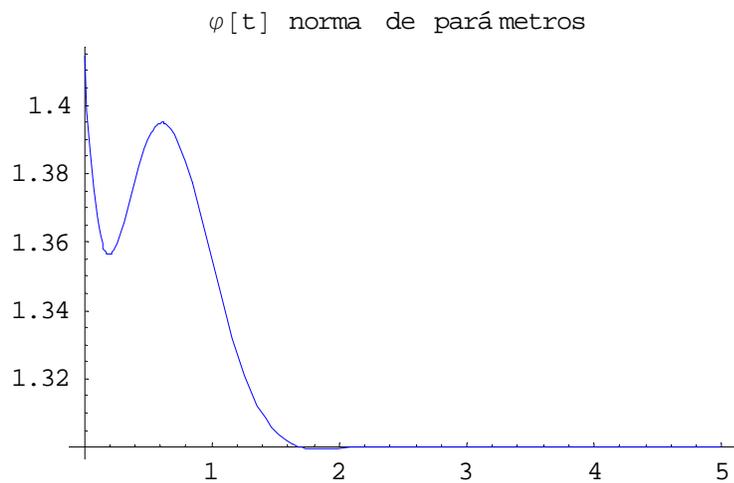
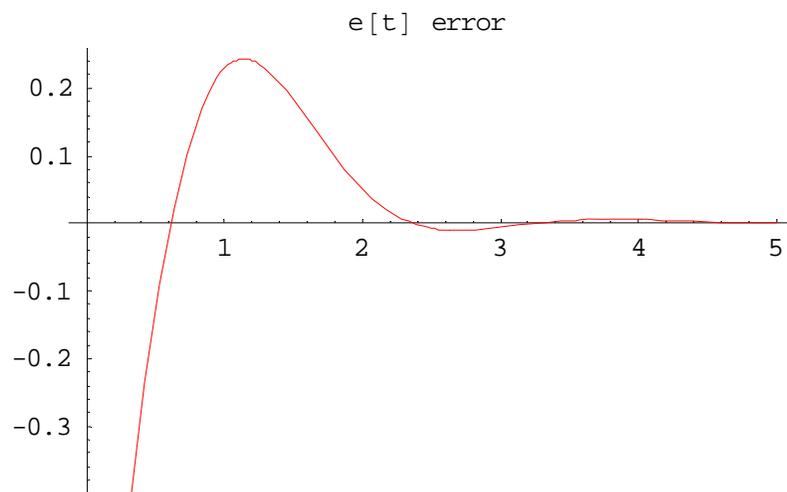
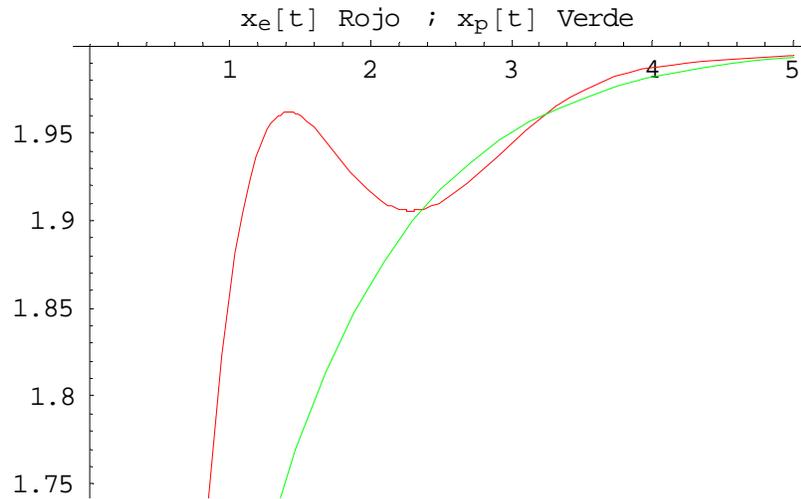
Ver programas de Matemática Identificación_1, Identificación_2, Identificación_3

La identificación de la planta con $a_p = -1$ y $k_p = 1$, usando el procedimiento discutido se muestra en el programa Identificación_1. El parámetro a_m en la ecuación (5.0.b) es igual a -3 y la simulación es llevada a cabo cuando: **(i)** $u(t) \equiv 0$, **(ii)** $u(t) \equiv 2$ y **(iii)** $u(t) = 2 \cos t + 3 \cos 2t$. Las condiciones iniciales se eligieron como se indica en la figura. En cada caso, el error de salida $e(t)$ y la norma de los errores de parámetro $\|\varphi(t)\| = \sqrt{\phi^2(t) + \psi^2(t)}$ son mostrados como funciones de tiempo.

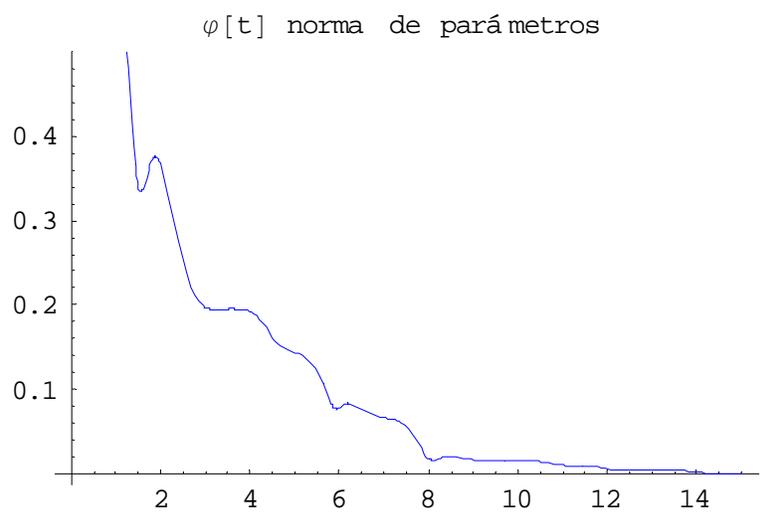
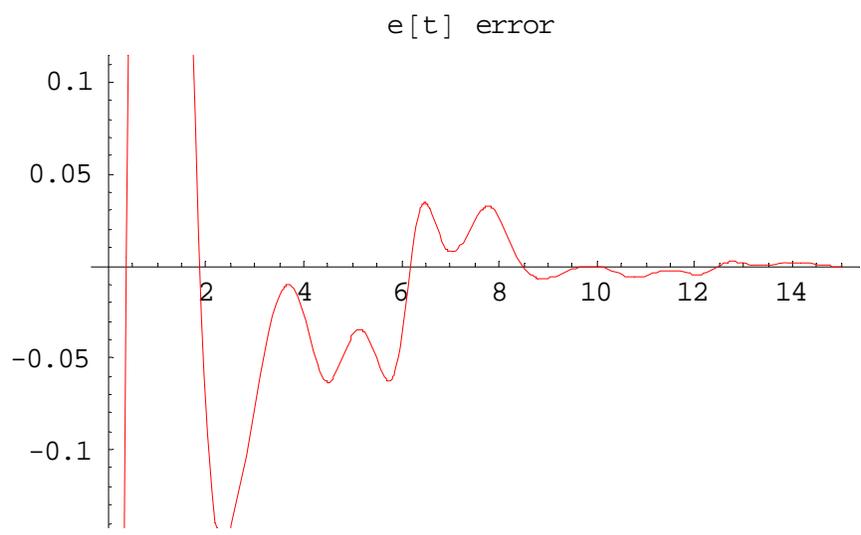
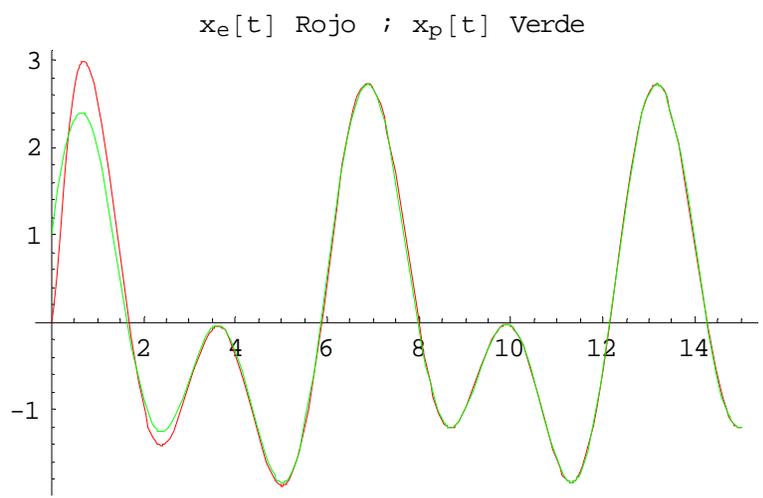
(i) Cuando $u(t) \equiv 0$ se ve que el error de salida converge a cero asintóticamente mientras que los errores de parámetro convergen a una constante con valor no-cero (programa Identificación_1).



- (ii) Cuando $u(t) \equiv 2$ se observa un desempeño similar de los errores. El valor asintótico de $\|\varphi(t)\|$ es más pequeño en este caso (programa Identificación_2).



- (iii) Cuando $u(t) = 2 \cos t + 3 \cos 2t$, $e(t)$, $\phi(t)$ y $\psi(t)$ tienden a cero asintóticamente (programa Identificación_3).



5.e. Identificación de plantas no-lineales

En la ecuación diferencial del error (5.1.b), el lado derecho es una función lineal de los errores de parámetro ϕ y ψ . Además, estos errores están asociados con las señales x y u que pueden ser medidas. Este hecho nos permite construir leyes de identificación estables de la forma de las ecuaciones (5.4). De aquí que el mismo procedimiento puede ser extendido a la identificación de casos especiales de plantas no-lineales como se muestra más abajo.

Sea la planta que va a ser identificada de la forma

$$\dot{x}_p = a_p x_p + \alpha f(x_p) + k_p g(u) \dots\dots\dots (5.8)$$

donde a_p, α y k_p son parámetros escalares constantes que son desconocidos, $f(\cdot)$ y $g(\cdot)$ son funciones no-lineales suaves conocidas tales que $x_p = 0$ es el único estado de equilibrio de la ecuación (5.8) cuando $g(u) \equiv 0$, y la ecuación (5.8) tiene soluciones acotadas para una entrada acotada u . Para estimar los parámetros desconocidos, construimos un estimador de la forma:

$$\dot{\hat{x}}_p = a_m \hat{x}_p + (\hat{a}_p(t) - a_p) x_p + \hat{\alpha}(t) f(x_p) + \hat{k}_p(t) g(u)$$

Así que el error e satisface la ecuación diferencial

$$\dot{e} = a_m e + (\hat{a}_p(t) - a_p(t)) x_p + (\hat{\alpha}(t) - \alpha) f(x_p) + (\hat{k}_p(t) - k_p) g(u)$$

y tiene la misma forma como la ecuación (5.1.b). Asimismo se desprenden las leyes adaptativas de identificación de la forma:

$$\begin{aligned} \dot{\hat{a}}_p &= -e x_p \\ \dot{\hat{\alpha}} &= -e f(x_p) \\ \dot{\hat{k}}_p &= -e g(u) \end{aligned}$$

pueden ser usadas para asegurar la estabilidad global de todo el sistema con el error de salida tendiendo a cero asintóticamente.

Simulación de Identificación de una Planta No-lineal

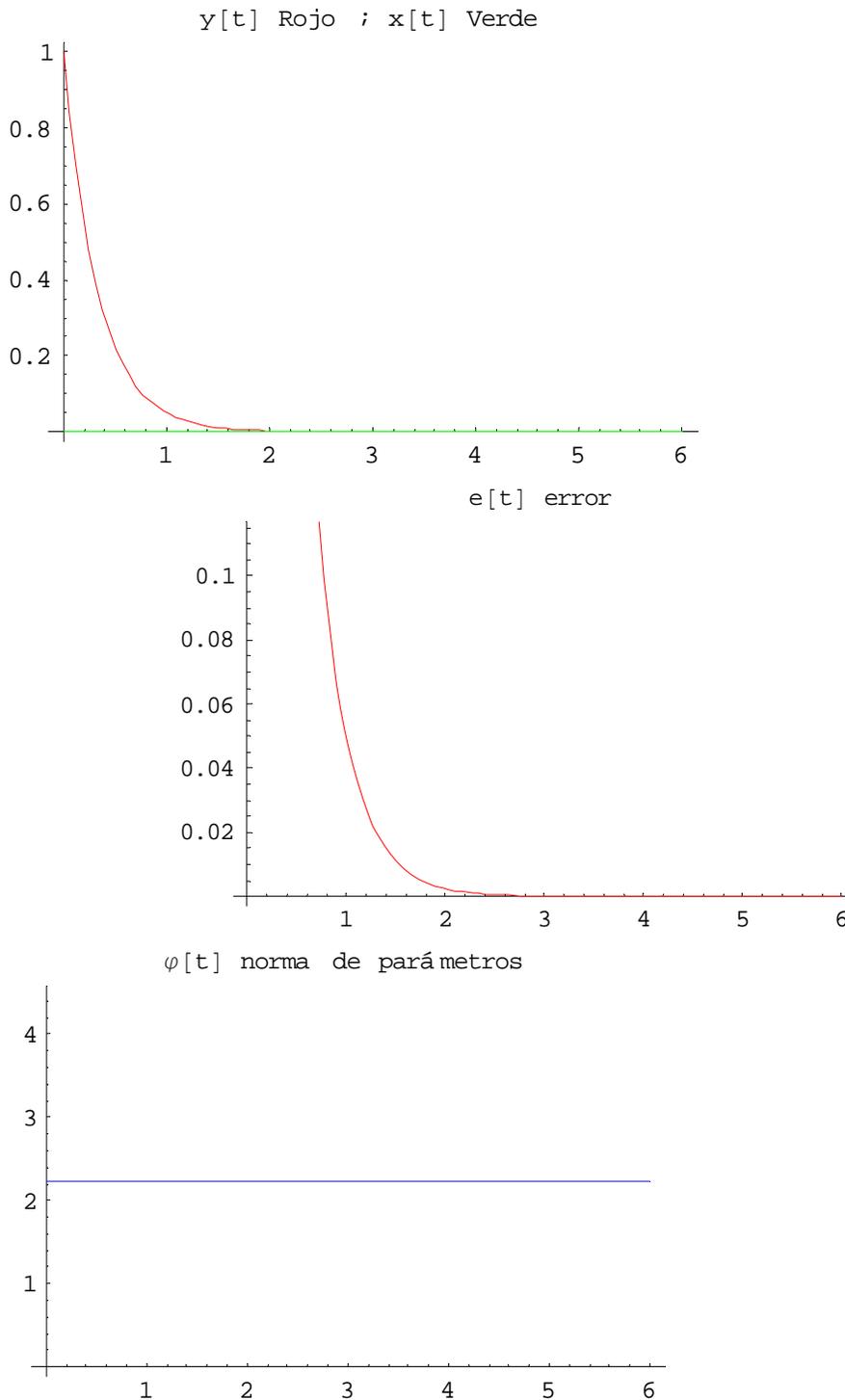
Ver programas de Matemática: Identificación_N1, Identificación_N2 e Identificación_N3

Los resultados obtenidos de la identificación de una planta no-lineal descrita por la ecuación diferencial

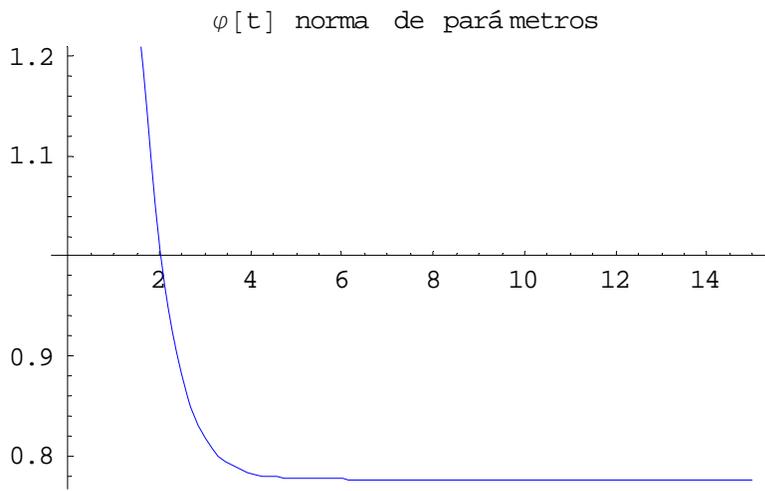
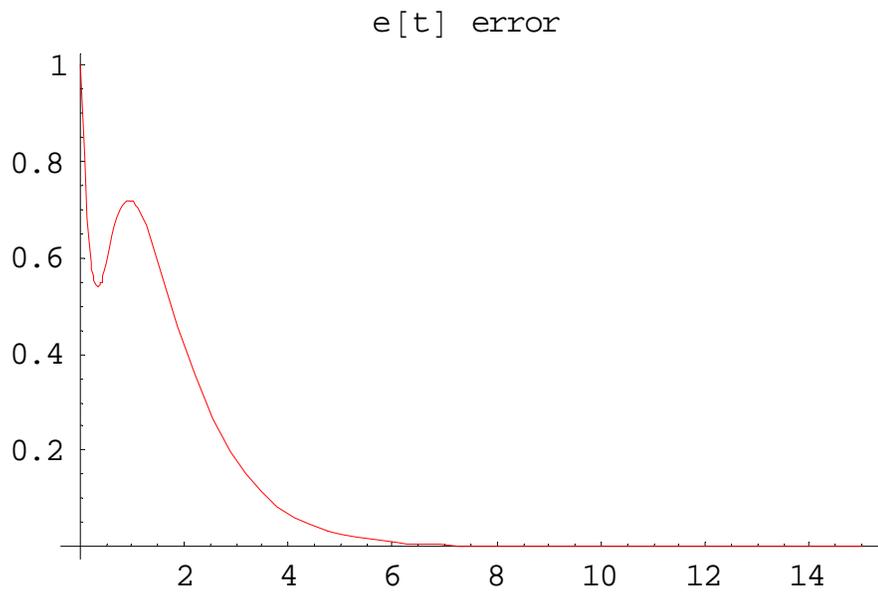
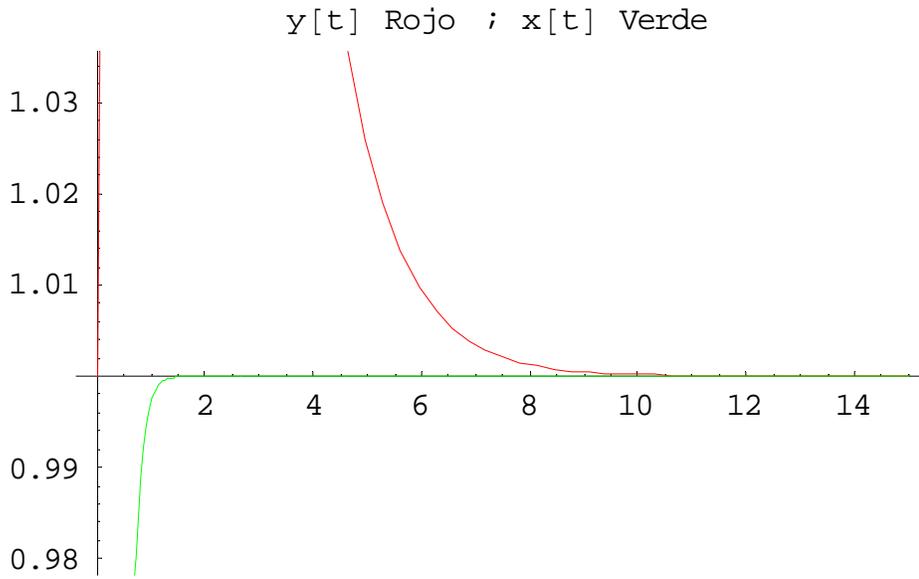
$$\dot{x}_p = -x_p - 2x_p^3 + u$$

Ahora se muestra en el programa Identificación_N1 con $a_m = -3$, con condiciones iniciales como se muestra en el programa y las mismas entradas como en el programa Identificación_1. El error de salida se ve que tiende a cero para $u(t) \equiv 0$ y $u(t) \equiv 3$ como se observa en los programas

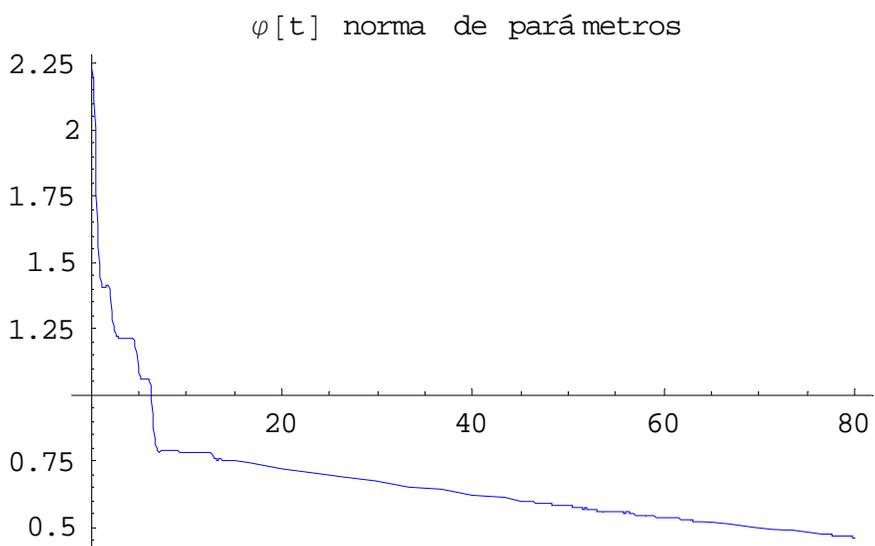
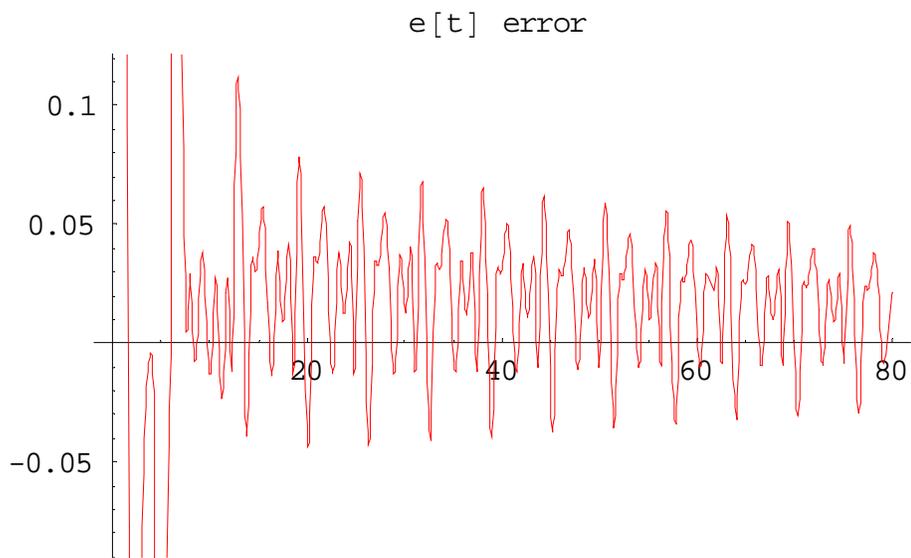
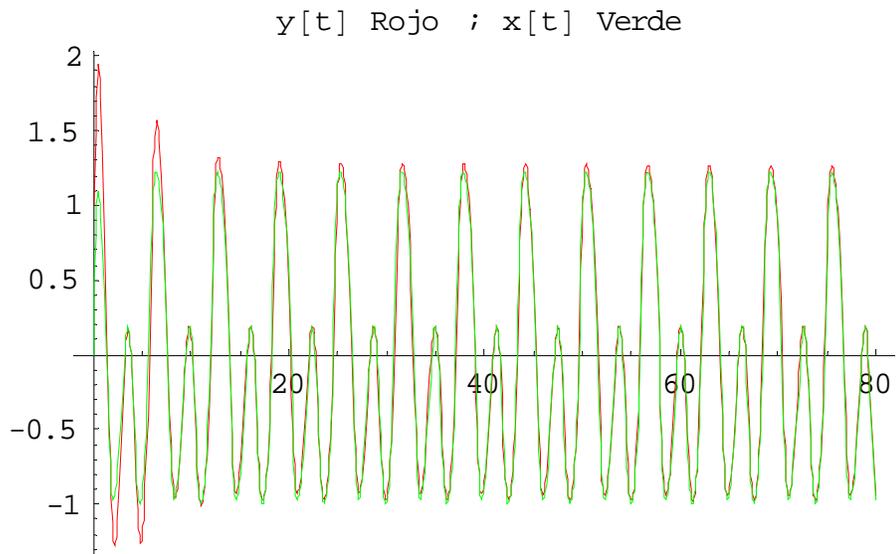
Identificación_N1 e Identificación_N2 pero los errores de parámetro no. Para $u(t) = 2\cos t + 3\cos 2t$, la salida así como los errores de parámetro tienden a cero. La velocidad de convergencia del error de parámetro en el programa Identificación_N3 es mucho mas lenta que en el programa Identificación_3.



Identificación No-lineal programa **Identificación_N1**



Identificación No-lineal programa **Identificación_N2**



Identificación No-lineal programa **Identificación_N3**

5.f. Control directo; el caso escalar

El problema más simple de control adaptativo de un sistema dinámico puede ser establecido en términos de una planta de la forma descrita para el caso escalar anteriormente expuesto.

Una planta con un par entrada-salida $\{u(\cdot), x_p(\cdot)\}$ está descrito por la ecuación diferencial escalar

$$\dot{x}_p(t) = a_p(t)x_p(t) + k_p(t)u(t)$$

donde $a_p(t)$ y $k_p(t)$ son parámetros de la planta. Un modelo de referencia es descrito por la ecuación diferencial de primer orden

$$\dot{x}_m(t) = a_m x_m(t) + k_m r(t) \dots\dots\dots (5.9)$$

donde $a_m < 0$, a_m, k_m son constantes conocidas y r es un tramo-continuo de una función de tiempo acotada. Se asume que a_m, k_m y r han sido escogidas para que $x_m(t)$ represente la salida deseada de la planta en el tiempo t . El objetivo es determinar una entrada de control u acotada, permaneciendo todas las señales del sistema acotadas y

$$\lim_{t \rightarrow \infty} |x_p(t) - x_m(t)| = 0 \dots\dots\dots (5.10)$$

Si $a_p(t)$ y $k_p(t)$ pueden ser directamente ajustados y sus valores son conocidos esto puede ser alcanzado trivialmente por la elección

$$a_p(t) \equiv a_m \quad k_p(t) \equiv k_m \quad \text{y} \quad u(t) = r(t), \forall t \geq t_0$$

Para hacer el problema más realista, se asume que los valores iniciales de $a_p(t)$ y $k_p(t)$ son desconocidos pero sus derivadas de tiempo pueden ser ajustadas usando la medición de señales en el sistema. Cuando se estableció de esta manera, el problema de control se ve idéntico a un problema de identificación usando el Modelo 1. En este caso el error entre las salidas de la planta y el modelo de referencia satisfacen la ecuación diferencial

$$\dot{e}(t) = a_m e(t) + (a_p(t) - a_m)x_p(t) + (k_p(t) - k_m)r(t) \dots\dots\dots(5.11)$$

Si las leyes adaptativas

$$\begin{aligned} \dot{a}_p(t) &= -e(t)x_p(t) \\ \dot{k}_p(t) &= -e(t)r(t) \end{aligned} \dots\dots\dots (5.12)$$

son usadas. Las ecuaciones (5.11) y (5.12) son similares a las ecuaciones (5.1.b) y (5.3). Su estabilidad global puede ser mostrada de la misma manera que fue descrita anteriormente.

5.g. Control por retroalimentación

En la práctica, el ajuste directo de los parámetros de la planta como esta descrito anteriormente no puede ser posible y el desempeño de la planta puede estar alterado sólo por el uso de ganancias de alimentación al frente y retroalimentación. Si a_p y k_p son constantes conocidas, la planta esta dada por:

$$\dot{x}_p(t) = a_p x_p(t) + k_p u(t) \dots\dots\dots (5.13)$$

entonces una entrada de control de la forma

$$u(t) = \theta^* x_p(t) + k^* r(t) \dots\dots\dots(5.14)$$

puede ser elegida cuando

$$\theta^* \triangleq \frac{a_m - a_p}{k_p} \quad \text{y} \quad k^* \triangleq \frac{k_m}{k_p}$$

Usando la expresión de la ecuación (5.14) en la ecuación (5.13) se puede ver que la función de transferencia de la planta junto con el controlador será lo mismo que lo que el modelo de referencia a fin de que se el objetivo en la ecuación (5.10) se cumpla. Se está seguro de la existencia de tales θ^* y k^* proporcionando un $k_p \neq 0$, esto es, cuando la planta es controlable. Cuando a_p y k_p son desconocidas, la entrada de control se elige para tener la forma siguiente

$$u(t) = \theta(t)x_p(t) + k(t)r(t) \dots\dots\dots (5.15)$$

Donde $\theta(t)$ y $k(t)$ son los parámetros ajustables del controlador. La adaptación debe ser tal que $\theta(t)$ y $k(t)$ evolucionen a valores constantes θ^* y k^* respectivamente. La planta dada por la ecuación (5.13) junto con el controlador adaptativo en la ecuación (5.15) la cual referiremos en adelante como **el sistema** puede ser descrito por:

$$\dot{x}_p(t) = (a_p + k_p \theta(t))x_p(t) + k_p k(t)r(t) \dots\dots\dots (5.16)$$

Definiendo el error de salida e y los errores de parámetro ϕ, ψ como:

$$e(t) \triangleq x_p(t) - x_m(t), \quad \phi(t) \triangleq \theta(t) - \theta^* \quad \text{y} \quad \psi(t) \triangleq k(t) - k^*$$

Obtenemos la ecuación del error de las ecuaciones (3.17) y (3.24) como

$$\dot{e}(t) = a_m e(t) + k_p \phi(t)x_p(t) + k_p \psi(t)r(t) \dots\dots\dots (5.17)$$

El problema entonces es determinar como $\dot{\theta}(t)$ y $\dot{k}(t)$ o alternativamente $\dot{\phi}(t)$ y $\dot{\psi}(t)$ son ajustados usando toda la información disponible.

Caso cuando k_p es desconocido (el signo de k_p es conocido)

En este caso, la entrada tiene la forma de la ecuación (5.15). Como se muestra adelante, la presencia del término k_p en la ecuación del error (5.17) requiere una modificación de las leyes adaptativas para mantener estabilidad. En particular las leyes adaptativas son escogidas como:

$$\begin{aligned} \dot{\theta}(t) = \dot{\phi}(t) &= -\text{sgn}(k_p) e(t) x_p(t) \\ \dot{k}(t) = \dot{\psi}(t) &= -\text{sgn}(k_p) e(t) r(t) \end{aligned} \dots\dots\dots (5.18)$$

donde $\text{sgn}(x)$ está definida como

$$\text{sgn}(x) = \begin{cases} +1, & \text{si } x \geq 0 \\ -1, & \text{si } x < 0 \end{cases}$$

La candidata a función de Lyapunov

$$V(e, \phi, \psi) \triangleq \frac{1}{2} [e^2 + |k_p| (\phi^2 + \psi^2)] \dots\dots\dots (5.19)$$

tiene una derivada de tiempo $\dot{V}(e, \phi, \psi)$, la cual cuando es evaluada a lo largo de las soluciones de las ecuaciones (5.17) y (5.18) que conduce a

$$\begin{aligned} \dot{V}(e, \phi, \psi) &= e\dot{e} + |k_p| [\phi\dot{\phi} + \psi\dot{\psi}] \\ &= a_m e^2 + k_p \phi e x_p + k_p \psi e r - |k_p| [\text{sgn}(k_p) \phi e x_p + \text{sgn}(k_p) \psi e r] \\ &= a_m e^2 \leq 0 \end{aligned} \dots\dots\dots (5.20)$$

Se concluye de las ecuaciones (5.19) y (5.20) que el estado de equilibrio del sistema descrito por las ecuaciones (5.17) y (5.18) es estable uniformemente y globalmente. Ya que \dot{e} es acotado y $e \in \mathcal{L}^2$, se puede mostrar que

$$\lim_{t \rightarrow \infty} e(t) = 0.$$

Como en el problema de identificación, los resultados no están afectados por la introducción de ganancias adaptativas positivas γ_1 y γ_2 en las leyes adaptativas.

5.h. Control de plantas no-lineales

Se mostrará que el procedimiento adoptado anteriormente para obtener el control adaptativo estable de una planta lineal de la ecuación (5.13) puede también ser extendido a una clase de plantas no-lineales. Esto se indica brevemente a continuación

Sea la planta no-lineal que será controlada adaptativamente dada por:

$$\dot{x}_p = a_p x_p + \alpha f(x_p) + k_p u \dots\dots\dots (5.21)$$

donde a_p, k_p, α son desconocidas, x_p así como $f(x_p)$ pueden ser medidas, y $f(\cdot)$ es una función suave de ese argumento con $f(0) = 0$. Es deseado que x_p siga la salida x_m del lineal modelo de referencia dado en la ecuación (5.9). Escogemos una entrada de control de la forma

$$u = \theta(t)x_p + \hat{\alpha}(t)f(x_p) + k(t)r \dots\dots\dots (5.22)$$

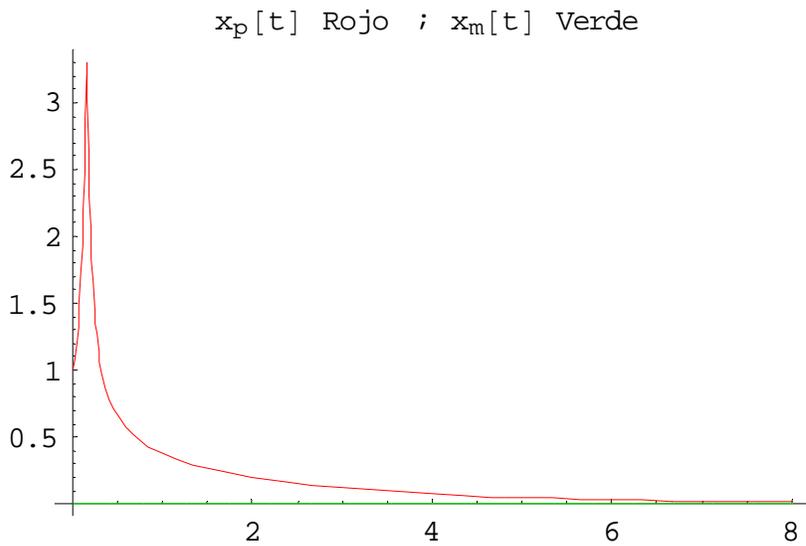
y las leyes adaptativas para ajuste $\theta(t), \hat{\alpha}(t)$ y $k(t)$ dadas por:

$$\begin{aligned} \dot{\theta} &= -\text{sgn}(k_p)ex_p \\ \dot{\hat{\alpha}} &= -\text{sgn}(k_p)ef(x_p) \dots\dots\dots (5.23) \\ k &= -\text{sgn}(k_p)er \end{aligned}$$

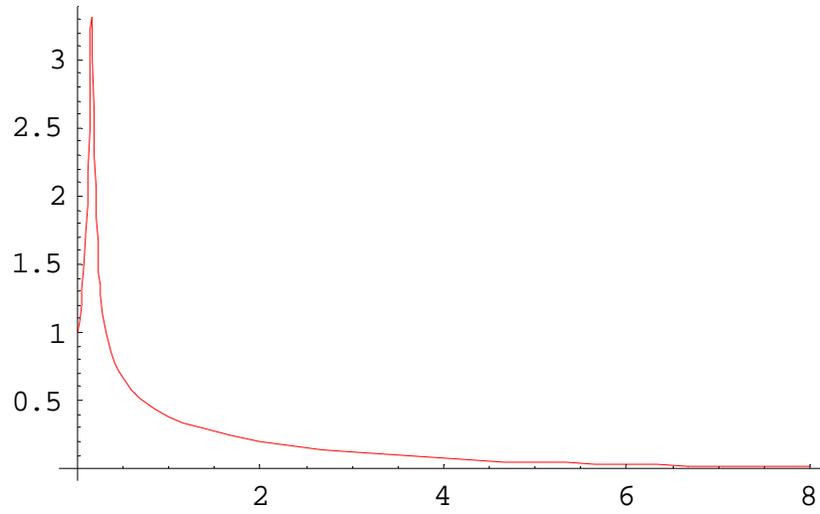
Usando los mismos argumentos de antes, se puede mostrar que el sistema entero dado por las ecuaciones (5.21)-(5.23) tiene soluciones acotadas y que $\lim_{t \rightarrow \infty} e(t) = 0$.

Simulación 3.4

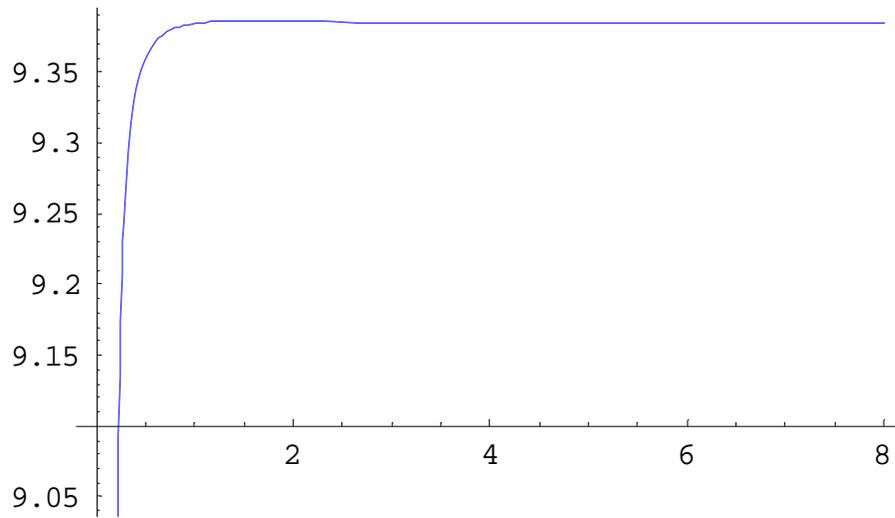
Los resultados son del control de un sistema no-lineal $\dot{x}_p = x_p + 3x_p^3 + u$ con un modelo de referencia descrito por $\dot{x}_m = -x_m + r$. La convergencia del error de salida y los errores de parámetro se muestran en los programas de Matemática Control_1, Control_2 y Control_3 para a) $r(t) \equiv 0$, b) $r(t) \equiv 5$ y c) $r(t) = 2 \cos t + 3 \cos 2t$.



$e[t]$ error

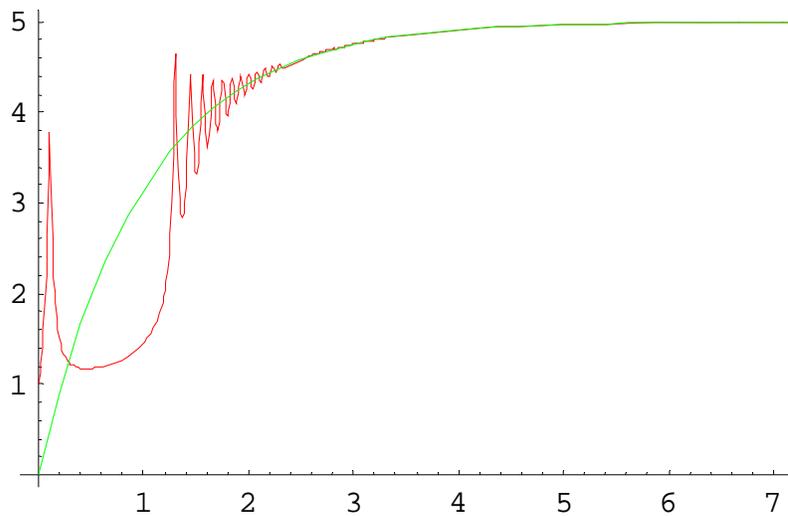


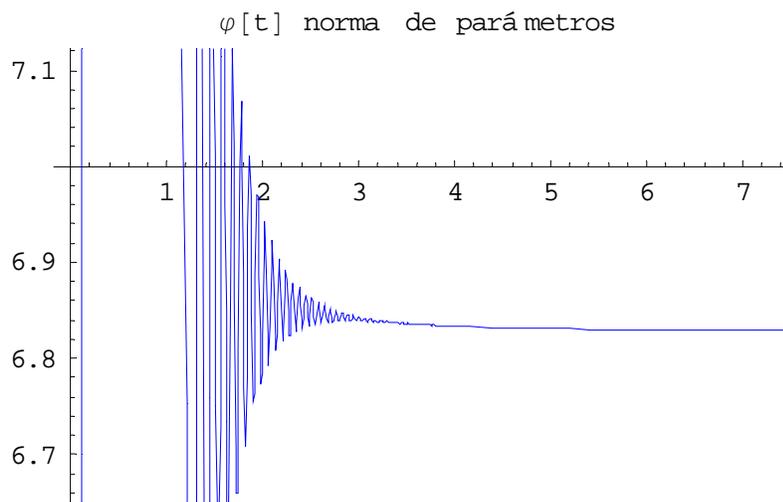
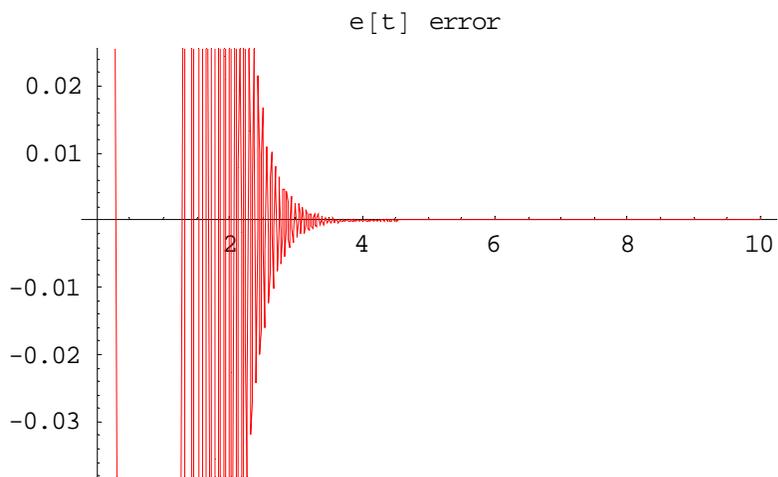
$\varphi[t]$ norma de parámetros



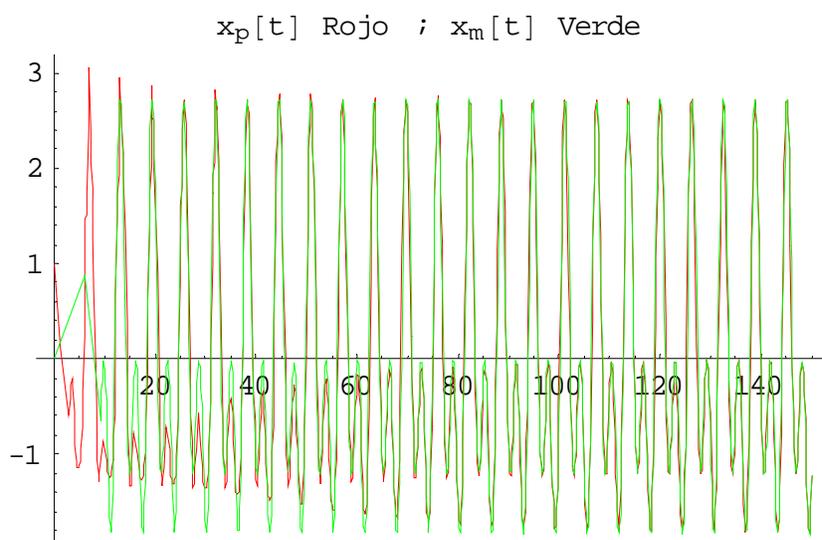
Problema de Control programa Control_1

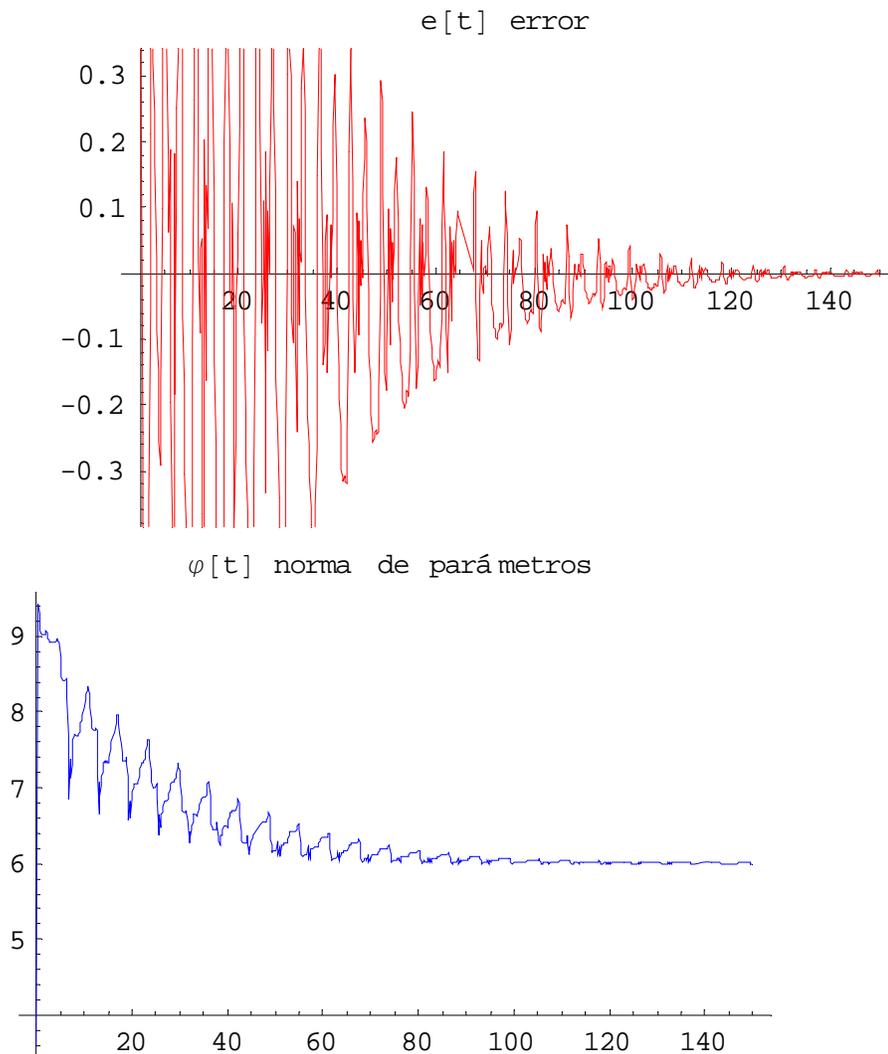
$x_p[t]$ Rojo ; $x_m[t]$ Verde





Problema de Control programa Control_2





Problema de Control programa Control_3

Comentarios 3.1

- i) En el problema de control, la planta, como está descrita en la ecuación (5.13) puede ser inestable.
- ii) En el problema de identificación, el objetivo principal es estimar los parámetros a_p y k_p de la planta. En contraste con esto, el objetivo en el problema de control es sólo que la salida de la planta siga la salida del modelo asintóticamente. Por lo tanto, la convergencia de los parámetros a los valores deseados, lo cual es central para el problema de identificación, no es crítico en el problema de control. En ambos casos, sin embargo, la excitación persistente de la entrada pertinente es necesaria para asegurar la convergencia de parámetros.
- iii) En el problema de identificación, los parámetros del modelo de identificación son ajustados. Cualquier inestabilidad que pueda resultar, puede manifestarse en sí misma sólo en este modelo. En contraste con esto, en el problema de control, los parámetros del controlador son ajustados en el lazo de retroalimentación de la planta. Por lo tanto, el proceso controlado puede llegar a ser inestable, lo cual para un punto de vista práctico, puede llegar a ser desastroso.

iv) La estructura del modelo de identificación esta a la discreción del diseñador y es determinado por la parametrización específica de la planta que es usada. La identificación de parámetros es llevada a cabo en el contexto de esta estructura predeterminada. Ya que la planta se asume que es estable y las entradas acotadas, la parametrización puede ser escogida de tal manera que la identificación es llevada a cabo usando señales acotadas conocidas. En contraste a esto sólo la entrada de referencia r puede ser asumida que será acotada en el problema de control y la limitación de todas las señales en el lazo adaptativo tiene que ser demostrada usando la elección de leyes adaptativas específicas.

6. REDES NEURONALES PARA CONTROL

6.a. Introducción

Las Redes Neuronales artificiales están modeladas de manera parecida a aquellas que intervienen en los procesos biológicos, a su procesamiento de la información y de manera especial al funcionamiento de la unidad básica del sistema nervioso, "la neurona".

La neurona recibe múltiples señales de otras neuronas a través de las dendritas, cada señal es multiplicada por un coeficiente de peso. Estas señales son sumadas en el cuerpo de la célula o soma y cuando la señal compuesta alcanza un umbral de valor, una señal conocida como la acción potencial es enviada a través del axón, el cual es el canal de salida de la neurona. Este es el principio base de una red neuronal artificial. Los atributos básicos de las Redes Neuronales se dividen en dos: su arquitectura y sus propiedades funcionales ó neurodinámicas.

Arquitectura: número de neuronas artificiales y su interconectividad.

Neurodinámicas: Define las propiedades de las redes neuronales como aprender, recordar, asociar y comparar constantemente nueva información con el conocimiento existente.

Una red neuronal descompone información compleja en elementos ó características fundamentales y esos elementos y sus relaciones entre cada uno son almacenados en los banco de memoria del cerebro.

Sabemos que entre las muchas utilidades las redes neuronales tienen la habilidad de aprender, clasificar, almacenar, recordar, cruzar referencias, interpolar, extrapolar, adaptar parámetros y realizar mantenimiento de la red. Durante el proceso de aprendizaje la red ajusta sus pesos, La sinapsis de pesos , en respuesta a un estímulo de entrada, así que la actual respuesta de salida converge a la respuesta de salida deseada. Cuando la salida actual responde de la misma manera que la deseada, la ha a completado la fase de aprendizaje, es decir adquirió el conocimiento. Si la actual respuesta difiere de la respuesta objetivo se genera un error de señal. El proceso de minimización del error requiere de un circuito especial y de un maestro o supervisor, de aquí el nombre de Aprendizaje Supervisado.

Con redes neuronales artificiales la cantidad de cálculos requeridos para minimizar el error depende del algoritmo usado. Las técnicas de optimización matemática son de uso extensivo en los modelos de redes neuronales. Algunos parámetros a observar son: el tiempo requerido por iteración, el número de iteraciones por entrada, patrón cuando el error alcanza un mínimo durante una sesión de entrenamiento. La red neuronal alcanza ya sea un mínimo local o uno global y si es local saber como se puede salir de el.

Matemáticamente hablando, una red neuronal representa un sistema dinámico que puede ser modelado como un conjunto de ecuaciones diferenciales acopladas. Para la teoría de retroalimentación, un sistema como este depende de los valores de los parámetros de la ecuación, pequeños cambios en los parámetros pueden dar por resultado: estabilidad, inestabilidad (oscilaciones) o aún inestabilidad catastrófica (caos).

Las redes neuronales están caracterizadas por:

- 1.- La computación de los programas en forma colectiva y cinérgica. Las operaciones están descentralizadas.
- 2.- Robustez. La operación es insensible a fallas diseminadas. La operación es insensible a entradas parciales o a entradas con imprecisiones.
- 3.- Aprendizaje. La red hace asociaciones automáticamente. El programa es creado por la red durante el aprendizaje.
- 4.- La red se adapta con o sin maestro. No interviene el programador.
- 5.- Un número grande de redes neuronales requiere sincronización

Las poderosas técnicas desarrolladas en el área de control adaptativo complementan la actual tecnología computacional y tienen un enorme potencial en el mundo de las aplicaciones, donde los sistemas han estado controlados en la presencia de incertidumbre. Aunque los sistemas adaptativos son muy no-lineales por su naturaleza, la mayoría de las teorías de tales sistemas están originadas en sistemas lineales.

En los años recientes, las redes neuronales multicapa y las recientes redes han emergido como componentes importantes, las cuales han probado ser muy exitosas en el reconocimiento de patrones y problemas de optimización. Para un punto de vista teórico, estas redes pueden ser consideradas como componentes que pueden ser usados de manera efectiva en sistemas no-lineales complejos. Típicamente, las entradas son donde los patrones de entrada son aplicados, la capa donde la salida es obtenida es la capa de salida y las capas entre las entradas y la capa de salida son las capas ocultas, estas se llaman así porque sus salidas no son directamente observables.

El control neuronal ha encontrado muchas aplicaciones en control. En general, una red neuronal puede ser entrenada para responder a variaciones en las entradas de tal manera que la salida se mantiene tan cercana como es posible a la salida deseada. Como en control tradicional se puede tener un sistema de control de lazo abierto ó de lazo cerrado.

La red de retroalimentación continuamente aprende, clasifica, genera nuevas asociaciones de entradas-salidas y actualiza la red hacia delante. Así esta red de retroalimentación dentro de ciertas fronteras de conocimiento, tiene la inteligencia para aprender y adaptarse continuamente.

Aplicaciones

Las redes neuronales artificiales y la lógica difusa no son sólo curiosidades científicas. ellas han sido aplicadas en varios productos; ejemplos hay muchos. En los trenes del metro se usan para determinar los inicios y paros adecuados y la velocidad de crucero de los trenes; en máquinas de lavado, determina la cantidad de agua y el número de enjuagues; en cámaras y videograbadoras ajusta el color, contraste, brillo, foco, etc. En automóviles con transmisión automática, determina el engranaje adecuado; en sistemas vehiculares inteligentes encuentra la mejor ruta automáticamente guía al automóvil; y en sistemas de comunicación, señales de proceso, rutas de canales y el sistema de controles. Otros ejemplos incluyen una variedad de sistemas. En Ingeniería financiera; en patrones de reconocimiento de voz y ópticos; en sistemas de seguridad; en robótica para guía y control de movimiento, reconocimiento de patrones ópticos y manipulación de objetos. Se da una particular atención en señales de proceso (ecualización, supresión de ondas reflejo); en proceso de datos (imagen y voz); en sistemas de comunicación (control del tráfico de flujo, desviaciones) y en control (controladores y robótica).

Las redes neuronales pueden ser diseñadas con circuito digital, con circuito análogo o con la combinación de ambos. El diseño con circuito digital es lo más reciente. Los circuitos digitales son más conocidos y más usados que los otros métodos, las entradas y salidas son binaria y eso hace que los cálculos más fáciles. Sin embargo las redes neuronales digitales tienen aplicaciones limitadas.

El control adaptable usando computadoras ha progresado mucho desde 1970. Importantes investigaciones en control inteligente continúan y muchas aplicaciones han encontrado su camino en el mercado usando lógica difusa y redes neuronales.

Un uso concerniente al control de motores industriales es la habilidad para predecir fallas del sistema. Las fallas del motor dependen de parámetros específicos del motor tales como corriente pasajera, características de envejecimiento, posición del motor, y otras que hacen que la predicción de fallas sea una tarea difícil. Redes neuronales son usadas para aprender las invariantes de la corriente del motor y de las características de instalación. De un 80% a 90% de precisión de la predicción se alcanza, comparada con el 30% con otras técnicas del estado del arte.

Muchas diferentes aplicaciones de redes neuronales se unen en robótica. Es típico, el más simple robot estacionario con el movimiento de un brazo y un juego de sensores (es decir, una videocámara) para objetos sensibles. Son enseñadas a ejecutar varias tareas a los robots. Las más simples tareas de movimiento, reconocimiento de objetos y manipulación de objetos. Más robots inteligentes son no-estacionarios, reconocen objetos, y ejecutan múltiples tareas como control de movimiento y balance, control de guía, evitar colisiones, control de brazos múltiples, identificación de objetos,

soporte y manipulación, reconocimiento de voz, generación de voz, razonamiento elemental y comunicación de datos.

Redes neuronales y lógica difusa han sido usadas en muchas aplicaciones vehiculares, incluyendo trenes, cambios en transmisiones automáticas y vehículos inteligentes. El ALVINN (Autonomous Land Vehicle in a Neural Network) es un proyecto para un vehículo de navegación autónoma. Mas trabajo en esta área de sistemas de control con lógica difusa es la propuesta para controlar la velocidad en vacío de un motor automotriz.

En máquinas de rolado de acero el espesor devastado es controlado con tolerancias muy pequeñas. El controlador tiene en cuenta algunas variables, muchas de ellas no-lineales. Algunas de ellas son el espesor a la entrada y a la salida, la fuerza de rolado, ubicación de los sensores de espesor y velocidad del devanado como movimiento entre rollos. Una aplicación propuesta usa una red neuronal con función básica radial (RBF).

6.b. Estructura de la Red Neuronal

Un modelo matemático de una Red Neuronal se muestra en la figura 6.a. La Red Neuronal tiene dos capas de pesos ajustables lo que la identifica como una Red Neuronal Estandar de dos capas para control de manipuladores

Los valores x_k son las entradas y y_i son las salidas de la Red Neuronal. La función $\sigma(\cdot)$ es una función de activación contenida en el la capa oculta de la red. Los pesos de la capa oculta de la red son v_{jk} y los de la capa de salida son w_{ij} . Los sesgos de la capa oculta son θ_{vj} y los sesgos de la capa de salida son θ_{wi} . El número de neuronas de la capa oculta esta dado por L .

La fórmula matemática que describe la Red Neuronal indicada en la figura 6.a esta dada como:

$$y_i = \sum_{j=1}^L \left[w_{ij} \sigma \left(\sum_{k=1}^n v_{jk} x_k + \theta_{vj} \right) + \theta_{wi} \right] \quad i = 1, 2, \dots, m \quad \dots \dots \dots (6.b.1)$$

Se puede aclarar esto por la definición de las matrices de pesos:

$$V^T = \begin{bmatrix} \theta_{v1} & v_{11} & \dots & v_{1n} \\ \theta_{v2} & v_{21} & \dots & v_{2n} \\ \vdots & \vdots & & \vdots \\ \theta_{vL} & v_{L1} & \dots & v_{Ln} \end{bmatrix}, \quad W^T = \begin{bmatrix} \theta_{w1} & w_{11} & \dots & w_{1L} \\ \theta_{w2} & w_{21} & \dots & w_{2L} \\ \vdots & \vdots & & \vdots \\ \theta_{wm} & w_{m1} & \dots & w_{mL} \end{bmatrix} \dots \dots \dots (6.b.2)$$

Las cuales contienen los sesgos en las primeras columnas. Entonces la Red Neuronal puede ser escrita como:

$$y = W^T \sigma(V^T x) \quad \dots \dots \dots (6.b.3)$$

Donde el vector de salidas es $y = [y_1 \quad y_2 \quad \dots \quad y_m]^T$ y el vector de entradas considerando el sesgo en la primer columna es:

$$x = [1 \quad x_1 \quad x_2 \quad \cdots \quad x_n]^T$$

El j -ésimo renglón de $V^T x$ está dado por:

$$\begin{bmatrix} \theta_{vj} & v_{j1} & v_{j2} & \cdots & v_{jn} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \theta_{vj} + \sum_{k=1}^n v_{jk} x_k = z_j$$

Además la función de activación $\sigma(z_j)$ como se presenta en la ecuación (6.b.3) esta definida como un vector función de la capa oculta. La cual esta representada en este caso por la función sigmoideal

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

donde si $z = [z_1, z_2, \dots]^T$ es un vector, se puede definir el vector $\sigma(z) = [\sigma(z_1), \sigma(z_2), \dots]^T$

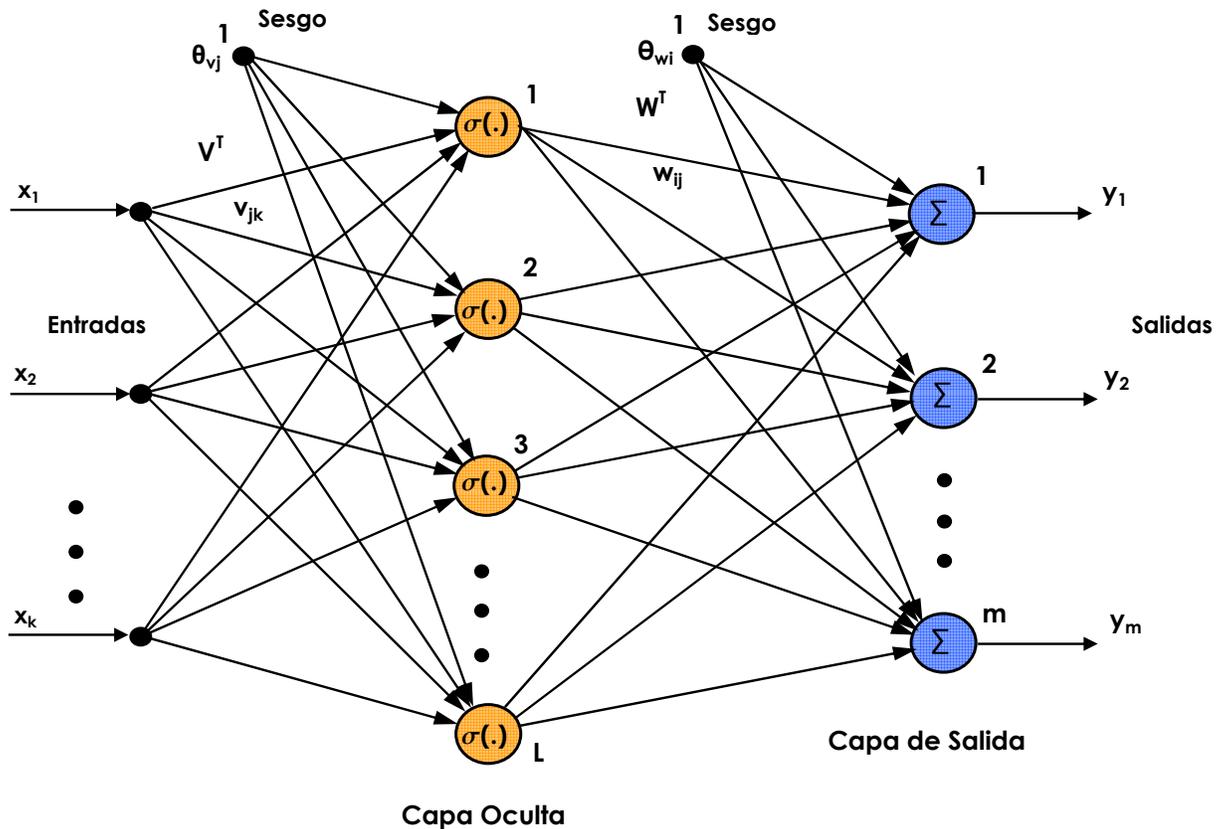
incluyendo el uno como el primer término en el vector $\sigma(V^T x)$ permitiendo incorporar los sesgos

θ_{wi} como la primer columna de W^T . Cualquier ajuste de W y de V incluirá también un ajuste de sesgos. Aunque para tener en cuenta sesgos distintos de cero, se considera $x_0 = 1$ y σ por la primer entrada que es uno.

6.c. El algoritmo de Retropropagación

El perceptrón multicapa MLP es utilizado en las Redes Neuronales para seguimiento y control. Para la actualización de pesos han sido usados diferentes algoritmos, los mas comunes son el Delta y el de Retropropagación.

El algoritmo de retropropagación ha sido ampliamente utilizado como un algoritmo de aprendizaje en redes neuronales multicapas con alimentación hacia delante, con una o más capas ocultas.



Modelo de Estructura de Red Neuronal de Dos Capas para control de manipuladores Figura 6.a

Basados en este algoritmo, la red aprende un mapeo de distribución asociativa entre las capas de entradas y salidas. Lo que hace que este algoritmo sea diferente de los otros es el proceso por el cual los pesos son calculados durante la fase de aprendizaje de la red.

Para actualizar los pesos se debe calcular el error. En la capa de salida el error es fácilmente medible que es la diferencia entre el valor actual de salida y el valor objetivo, sin embargo en las capas ocultas no hay una observación directa del error de aquí que otras técnicas deben utilizarse para calcular el error en las capas ocultas que logren minimizar el error de salida.

Durante una sesión de entrenamiento de la red, se presentan un par de patrones (X_k, T_k) donde X_k es el patrón de entradas y T_k es el patrón de objetivos deseado. El patrón X_k causa respuestas en las salidas de cada neurona en cada capa y así hasta la capa de salidas con las salidas actuales O_k . En la capa de salida la diferencia entre la salida actual y la salida objetivo producen los errores de señal. El error de señal depende de los valores de los pesos de las neuronas en cada capa. La velocidad y precisión del proceso de aprendizaje, esto es, el proceso de actualización de pesos, también depende de un valor conocido como *tasa de aprendizaje*.

La actualización de los pesos para la Capa de Salida W está dada por:

$$W_{t+1} = W_t + F \sigma(V_t^T x_d) E_t^T \dots\dots\dots (6.c.1(a))$$

Y la actualización de pesos para la Capa Oculta V está dada por:

$$V_{t+1} = V_t + Gx_d (\sigma_t^T W_t E_t)^T \dots\dots\dots (6.c.1(b))$$

Este es un algoritmo de ajuste de tiempo discreto donde el tiempo indexado es t . La salida deseada de la Red Neuronal en respuesta a la entrada de referencia $x_d \in \mathbf{R}^n$ está denotado como $y_d \in \mathbf{R}^m$. El error de salida de la Red Neuronal al tiempo t es $E_t = y_d - y_t$, donde $y_t \in \mathbf{R}^m$ es la salida actual de la Red al tiempo t . F, G son matrices de ponderación seleccionadas por el diseñador que determinan la velocidad de convergencia del algoritmo.

El término $\sigma'(\cdot)$ es la derivada de la función de activación la cual puede ser computada fácilmente.
$$\sigma' = \text{diag} [\sigma(V^T x_d)] [I - \text{diag} [\sigma(V^T x_d)]]$$

La versión del ajuste por retropropagación en tiempo continuo está dada como:

$$\begin{aligned} \bar{W} &= F \sigma(V^T x_d) E^T \\ \bar{V} &= G x_d (\sigma^T W E)^T \dots\dots\dots (6.c.2) \end{aligned}$$

Actualización de pesos para ejecución de seguimiento garantizada

Para que un algoritmo de ajuste de pesos garantice el seguimiento estable del sistema en lazo cerrado es requerido demostrar que el error de seguimiento $r(t)$ esta adecuadamente pequeño y que los pesos de las capas de las Redes Neuronales V, W permanecen acotados y así entonces el control $r(t)$ también esta acotado.

En este caso no hay la fase de aprendizaje de las redes fuera de línea, y los pesos de las redes son fácilmente inicializados sin los requerimientos de "pesos de estabilización inicial".

Retropropagación para el caso ideal

El siguiente resultado detalla el desempeño del lazo cerrado en ciertas condiciones ideales que demandan lo siguiente:

- 1) No hay error de reconstrucción funcional de la red
- 2) No hay perturbaciones no modeladas en las dinámicas del brazo robótico y
- 3) No hay términos de alto orden de la serie de Taylor.

Se puede demostrar que bajo estas condiciones el algoritmo trabaja.

Teorema 6.c.1

Sea la trayectoria deseada acotada y suponer que el término de perturbación $w_1(t)$ es igual a cero entonces el ajuste de pesos esta proporcionado por:

$$\dot{\hat{W}} = F \hat{\sigma} r^T \dots\dots\dots (6.c.3)$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}^T \hat{W}r)^T \dots\dots\dots (6.c.4)$$

Y cualesquiera matrices constantes positivas F, G . Entonces el error de seguimiento $r(t)$ va a cero con el tiempo t y los pesos estimados de V y W están acotados.

6.d. Aproximación a funciones continuas con la Red Neuronal

Redes Neuronales para la Aproximación de Funciones Continuas

Muchos resultados bien conocidos dicen que cualquier función suficientemente suave puede ser aproximada arbitrariamente cercana sobre un conjunto compacto usando una Red Neuronal de dos capas con pesos apropiados.

Los resultados para la aproximación de funciones continuas dice que:

Teorema 6.d.A

Dada cualquier función $f \in C(S)$ (el espacio de funciones continuas) con S un subconjunto compacto de R^n y cualquier $\varepsilon > 0$, existe una suma $G(x)$ de la forma

$$G(x) = \sum_{k=0}^L \alpha_k \sigma(m_k^T x + n_k) \dots\dots\dots (6.d.1)$$

Para algún $L, m_k \in R^n, n_k \in R$ tal que

$$|G(x) - f(x)| < \varepsilon \dots\dots\dots (6.d.2)$$

para toda $x \in S$.

Generalmente, si ε mas pequeño, entonces el número L de neuronas de la capa oculta es mas grande. Esto es, es requerida una Red Neuronal mas grande para mayor precisión.

La función $\sigma(\bullet)$ puede ser cualquier función sigmoideal continua, donde una función sigmoideal está definida como:

$$\sigma(x) \rightarrow \begin{cases} 1 \dots\dots \text{para } x \rightarrow +\infty \\ 0 \dots\dots \text{para } x \rightarrow -\infty \end{cases} \dots\dots\dots (6.d.3)$$

Este resultado muestra que cualquier función continua puede ser aproximada arbitrariamente bien usando una combinación lineal de funciones sigmoideales. Esta es conocida como **La Propiedad de Aproximación Universal** de las Redes Neuronales.

Una función general $f(x) \in C^m(S)$ puede ser escrita como:

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x) \dots \dots \dots (6.d.4)$$

con $N_1 = n$, $N_3 = m$, y $\varepsilon(x)$ un vector error de la reconstrucción funcional con Red Neuronal.

Si en N_2 existen pesos "ideales" W y V y son constantes, entonces $\varepsilon = 0$ para toda $x \in S$. se dice que $f(x)$ está en el rango funcional de la Red Neuronal. En general, dada una constante real $\varepsilon_N > 0$, decimos que $f(x)$ esta dentro del rango de la Red Neuronal ε_N si existe N_2 y pesos constantes tales que para toda $x \in \mathbb{R}^n$ existe una $f(x)$ como se expresa en la ecuación (6.d.4) que cumple que $\|\varepsilon\| < \varepsilon_N$.

Varios resultados conocidos para varias funciones de activación $\sigma(\bullet)$ basados en el teorema de Stone-Weierstrass dice que cualquier función suficientemente suave puede ser aproximada por una red adecuadamente grande.

El rango funcional de la Red Neuronal se dice que es denso en $C^m(S)$ si para cualquier $f \in C^m(S)$ y $\varepsilon_N > 0$, existe un N_2 finito, y una W y V tales que, de acuerdo con la ecuación (6.d.4) se mantiene que $\|\varepsilon\| < \varepsilon_N$, $N_1 = n$, $N_3 = m$. Los resultados comunes son los siguientes para el caso de σ las "funciones de aplastamiento" (un límite, medible, una función no decreciente para números reales en $[0,1]$), las cuales incluyen por ejemplo el escalón, la rampa y la sigmoideal.

Teorema 6.d.A

Sea $N_1 = n$, $N_3 = m$ y sea σ cualquier función de aplastamiento. Entonces el rango funcional de la Red Neuronal, ecuación (6.b.3), es denso en $C^m(S)$.

En este resultado la métrica que define la densidad es la norma del supremo. Más aún los sesgos de la última capa $\theta_{w\ell} w$ no son necesarios en este resultado. Los cuestionamientos de selección de σ , de escoger N_2 para un especial $S \subset \mathbb{R}^n$ y un ε_N son los tópicos actuales de investigación.

7. PERTURBACIONES

7.a. Tipos de perturbaciones

La mayoría de los actuadores usados en aplicaciones de control son dispositivos de control continuo. Si las relaciones de entrada-salida del mecanismo son ecuaciones algebraicas no-lineales, esto representa una **no-linealidad estática**. Por otro lado, si las relaciones entrada-salida son ecuaciones diferenciales no-lineales, esto representa una **no-linealidad dinámica**. Como ejemplos de no-

linealidades estáticas se encuentran: fricción, zonas muertas y saturación; como no-linealidades dinámicas se encuentran backlash e histeresis.

Una clase general de procesos industriales tiene la estructura de un sistema dinámico precedido por las no-linealidades del actuador. Problemas en el control de estos procesos son exacerbados cuando la precisión requerida es alta, como en mecanismos de microposicionamiento. Debido a la naturaleza no-analítica de las no-linealidades del actuador y el hecho que sus funciones no-lineales exactas son desconocidas, tales procesos representan un reto para el diseñador de control.

Modelado de las perturbaciones en el espacio de estados

Si se quiere conseguir que la respuesta del sistema sea la que deseamos a pesar de las influencias que las diferentes perturbaciones tienen sobre el sistema, resulta necesario comprender lo que son, así como describirlas y modelarlas adecuadamente.

Las perturbaciones pueden tener diversos orígenes:

- Variaciones en la carga a la que está sometido el sistema
- Errores producidos en las medidas de la salida del sistema debidos al ruido, derivadas de los sensores, errores en la transmisión de la señal, etc.
- Variaciones en la planta y en los actuadores.
- Simplificaciones y errores en el modelo matemático del sistema que hacen que la salida real no coincida con la modelada.

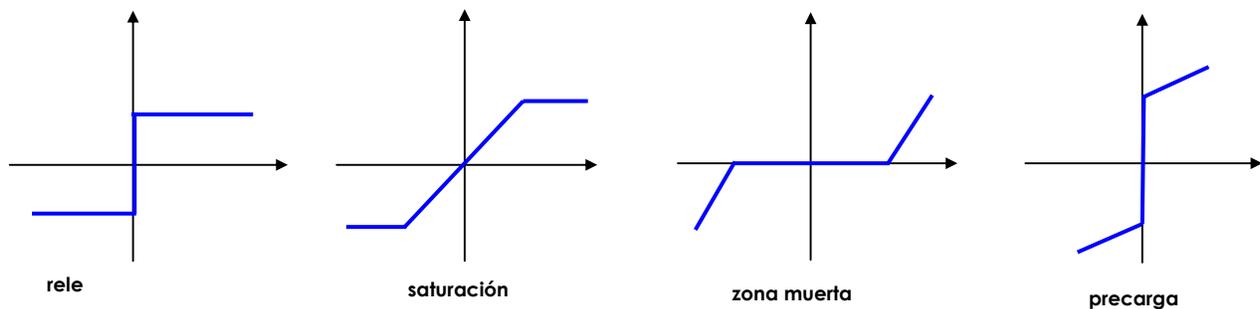
Las diferentes perturbaciones sobre el sistema w pueden conocerse en muchas ocasiones, por medio de una serie de experimentos y pruebas sobre el sistema, y en este caso dicha información puede utilizarse para construir un modelo matemático mejorado del sistema.

Se pueden diferenciar dos situaciones, si las perturbaciones se pueden predecir o si no se pueden predecir (ruido blanco). En el caso del modelado de sistemas no-lineales, las no-linealidades pueden ser *suaves o pronunciadas* y se pueden dividir en dos grandes tipos, en función de su naturaleza.

Funciones Analíticas Simples. Estas son desarrolladas en serie de Taylor y que, por lo tanto tienen todas sus derivadas. Entre estas funciones se encuentran las funciones potenciales, sinusoidales o productos de estas, las cuales presentan una variación suave.

Funciones lineales por partes o no-linealidades pronunciadas. No son funciones analíticas pero, sin embargo, dentro de cada zona el tratamiento es prácticamente lineal. Ejemplos de ellos son: fricción, relé, saturación, zona muerta, precarga y rectificación.

Funciones multivariables. Representan los efectos de la evolución temporal de las entradas como son: relé con histéresis, la histéresis magnética y el backlash.



Graficas correspondientes a no-linealidades pronunciadas

Grafico 7.1

Efectos de las No-linealidades

Entre los efectos debidos a la presencia de no-linealidades en el sistema se encuentran:

- **Aparición de nuevas frecuencias sobre el sistema.**

Al trabajar con sistemas no-lineales se puede ver que se modifica la frecuencia, es decir aparecen nuevos armónicos y desarrollando la salida en serie de Fourier se tendrán nuevos

coeficientes no nulos: $y(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} [A_n \cos(n\omega t) + B_n \sin(n\omega t)]$.

- **Pérdida de la precisión estática.**

Se pierde la precisión estática y puede haber más de un punto de equilibrio. El punto puede depender de las condiciones iniciales. Por ejemplo la fricción puede producir una banda de fricción.

- **Tiempos de respuesta de tipo distinto a exponenciales reales o complejas.**

Pueden aparecer respuestas de tipo distinto a exponenciales reales o complejas como son las oscilaciones exponencial decreciente y las oscilaciones lineal decreciente.

- **Aparición de Ciclos Límite.**

Los ciclos límite son oscilaciones no-lineales con frecuencia fija y cuya amplitud no depende de las condiciones iniciales.

7.b. Fricción en servomecanismos

La fricción es la resistencia natural relativa al movimiento entre dos cuerpos en contacto y es esencial para la operación común de sistemas mecánicos (ruedas, embragues, etc.) Pero en la mayoría de los procesos industriales esto también representa un problema ya que es difícil modelar y problemático tratarla en el diseño de sistemas de control. Fabricantes de componentes de sistemas de control de precisión se esfuerzan en minimizar la fricción y esto representa un incremento importante en los costos. Sin embargo, no obstante los esfuerzos en reducir la fricción el problema permanece.

En los posibles efectos de fricción no deseables se incluye *hangoff* y los *ciclos límite*. El *hangoff* evita el error en estado estable para el conveniente cero con un paso mandado en la entrada (esto

puede interpretarse como un ciclo límite de corriente directa). El ciclo límite es el comportamiento en el cual el error en estado estable oscila alrededor de cero.

La fricción es un fenómeno no-lineal complicado en el cual se produce una fuerza que tiende a oponerse al movimiento en un sistema mecánico. El movimiento entre los dos cuerpos en contacto causa la disipación de energía en el sistema. El mecanismo físico de la disipación depende de los materiales de las superficies rugosas, sus acabados, la lubricación aplicada y otros factores, muchos de ellos aún no son totalmente entendidos.

Lo concerniente a la ingeniería de control no es reducir la fricción sino tratar con esta ya que no puede ser eliminada. Para compensar la fricción es necesario entender y tener un modelo del proceso de fricción. Muchos investigadores han estudiado el modelado de la fricción.

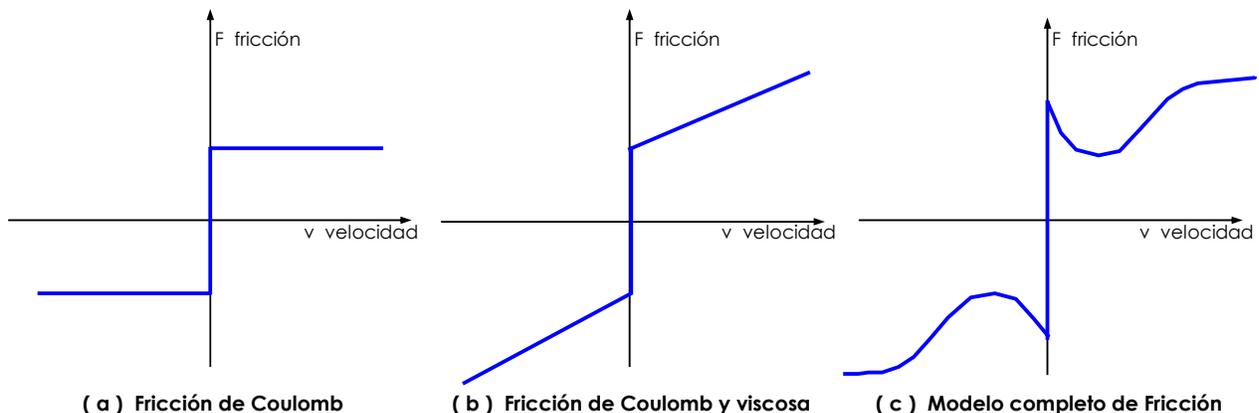
Modelos de Fricción Estática

El modelo clásico de la fuerza de fricción es proporcional a la carga, se resiste al movimiento y es independiente del área de contacto lo que era conocido por Leonardo Da Vinci. El modelo de fricción de Da Vinci fue redescubierto por Coulomb y es ampliamente usado hoy como el más simple modelo de fricción, descrito como

$$F(v) = (a)\text{sgn}[v]$$

donde v es la velocidad relativa y $F(v)$ es la correspondiente fuerza de torque. El parámetro a es tomado generalmente como una constante por simplicidad. La fricción de Coulomb es mostrada en el Gráfico 7.2(a). Un modelo de fricción mas detallado se muestra en el Gráfico 7.2(b) el cual incluye la *fricción viscosa*, un término proporcional a la velocidad.

Experimentos físicos han mostrado que en muchos casos la fuerza requerida relativa al inicio del movimiento es más grande que la fuerza que se opone al movimiento una vez que empieza. Este efecto es conocido como *fricción estática*. El modelo de fricción estática es complementado con el uso de una no-linealidad de la forma como se muestra en el Gráfico 7.2(c).



MODELOS DE FRICCIÓN

Grafico 7.2

Una fórmula empírica a veces usada para expresar la dependencia de la fuerza de fricción con respecto a la velocidad es:

$$F(v) = (a - be^{-c|v|} + d|v|\text{sgn}(v))$$

en el cual los parámetros a, b, c y d son escogidos para dar la pendiente deseada a la función de fricción.

Modelos de Fricción Dinámica

Si bien la fricción usualmente se modela como un mapeo estático discontinuo entre velocidad y fricción de torque, la cual depende del signo de la velocidad, hay algunas propiedades interesantes observadas en los sistemas con fricción que no pueden ser explicadas sólo por modelos estáticos. Esto es debido básicamente por el hecho de que la fricción no tiene una respuesta instantánea para un cambio en la velocidad (es decir tiene una dinámica interna).

Algunos ejemplos de las características de la dinámica son:

- **Movimiento stick-slip**, el cual consiste en la oscilación de un ciclo límite a bajas velocidades causada por el hecho que la fricción es más grande en reposo que durante el movimiento.
- **Desplazamiento presliding**, lo cual significa que la fricción se comporta como resorte cuando la fuerza aplicada es menor que la fuerza de fricción estática allá de donde empieza.
- **Retraso friccional** lo cual significa que hay histéresis en la relación entre fricción y velocidad.

Todas estas características de fricción estáticas y dinámicas están contenidas en el modelo dinámico y el modelo analítico propuesto en Canudas de Wit (1995). Este modelo es llamado *LuGre*.

El modelo LuGre esta dado por:

$$\begin{aligned} \frac{dz}{dt} &= \dot{q} - \frac{\sigma_0}{g(\dot{q})} z |\dot{q}| \\ g(\dot{q}) &= \alpha_0 + \alpha_1 e^{-\left(\frac{\dot{q}}{v_0}\right)^2} \\ F &= \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 \dot{q} \end{aligned}$$

donde \dot{q} es la velocidad angular y F es la fuerza de fricción. La primera de estas ecuaciones representa la dinámica del estado z de la fricción interna, la cual describe el promedio relativo a la deflexión de las superficies de contacto durante las fases de fricción estática. Este estado no es medible.

La función $g(\dot{q})$ describe la parte del *estado estable* del modelo o los movimientos de velocidad constante: v_0 es la velocidad Stribeck, $(\alpha_0 + \alpha_1)$ es la fricción estática, y α_0 es la fricción de Coulomb. Así el modelo completo de fricción está caracterizado por cuatro parámetros estáticos, $\alpha_0, \alpha_1, \alpha_2$ y v_0 así como dos parámetros dinámicos σ_0 y σ_1 . El parámetro σ_0 puede ser entendido

como un coeficiente de rigidez de las deformaciones microscópicas de z durante el desplazamiento presliding, mientras que σ_1 es un coeficiente de amortiguamiento asociado con $\frac{dz}{dt}$.

7.c. Fricción en brazos manipuladores

Función Fricción en Actuadores

Un modelo completo de fricción adecuado al diseño de controladores industriales está dado en el modelo de Canudas de Wit en (1991) como:

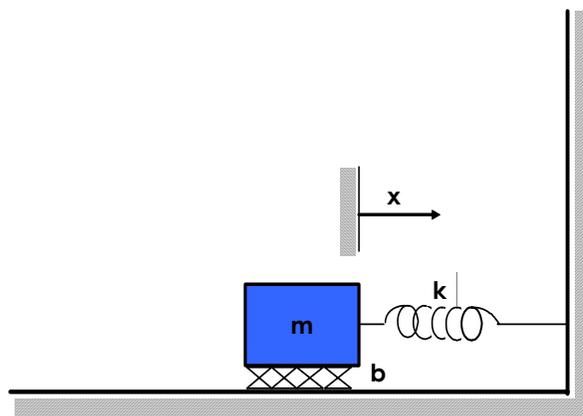
$$F(v) = \left[\alpha_0 + \alpha_1 e^{-\beta_1 |v|} + \alpha_2 (1 - e^{-\beta_2 |v|}) \right] \text{sgn}(v)$$

donde la fricción de Coulomb esta dada por α_0 (Nm), la fricción estática es $(\alpha_0 + \alpha_1)$ (Nm), y $\alpha_2 \left(Nm - \frac{\text{seg}}{\text{rad}} \right)$ representa el modelo de fricción viscoso. El efecto por el cual para pequeños valores de v la fuerza de fricción esta decreciendo con respecto a la velocidad es llamada *fricción negativa viscosa*, o el *efecto Stribeck*.

8. CONTROLADOR DE RETROALIMENTACION LINEAL TIPO PD

8.a. Un sistema amortiguado con fricción

Para analizar la naturaleza de los movimientos de las soluciones de una clase de sistemas no-lineales, utilizamos como ejemplo de un sistema mecánico simple amortiguado con fricción de segundo orden.



SISTEMA AMORTIGUADO RESORTE-MASA CON FRICCIÓN
Gráfico 8.1

La ecuación que representa el sistema mecánico es

$$m\ddot{x} + b\dot{x} + kx = 0 \dots\dots\dots (8.1)$$

Aplicando la Transformada de Laplace la ecuación (8.1) queda

$$ms^2 + bs + k = 0$$

Esta ecuación tiene las raíces

$$s_1 = -\frac{b}{2m} + \frac{\sqrt{b^2 - 4mk}}{2m}$$

$$s_2 = -\frac{b}{2m} - \frac{\sqrt{b^2 - 4mk}}{2m}$$

La ubicación de s_1 y s_2 determina los polos del sistema en el plano real-imaginario y dicta la naturaleza de los movimientos del sistema.

Observaciones:

Caso 1 Raíces reales y desiguales. En este caso cuando $b^2 > 4mk$, esto es, la fricción domina y se produce un "comportamiento no oscilatorio". Y para que el sistema sea estable debe cumplirse que $b > \sqrt{b^2 - 4mk}$ para que las raíces sean negativas. A esta respuesta se le llama **sobreamortiguado**.

Caso 2 Raíces complejas. En este caso cuando $b^2 < 4mk$; esto es la rigidez domina y se produce un "comportamiento oscilatorio". A esta respuesta se le llama **subamortiguado**.

Caso 3 Raíces reales e iguales. Este es el caso especial cuando $b^2 = 4mk$, aquí la fricción y rigidez están balanceadas, lo que produce un "comportamiento menos rápido y no oscilatorio". A esta respuesta se le llama **críticamente amortiguada**.

El tercer caso amortiguamiento crítico es generalmente una situación deseable. En los casos en que uno de los polos tiene una magnitud mucho mayor que la otra, el polo de mayor magnitud disminuirá a cero rápidamente en comparación con el otro polo dominante. Esta misma noción de la dominación se extiende a los sistemas de mayor orden, por ejemplo a menudo un sistema de tercer orden puede estudiarse como sistema de segundo orden si se consideran sólo dos polos dominantes.

Ejemplo 1 raíces reales y desiguales

Se desea determinar el movimiento del sistema representado en el esquema anterior Gráfico (8.1) y expresado por la ecuación (8.1), si los valores de los parámetros son: $m = 1$, $b = 5$ y $k = 6$.

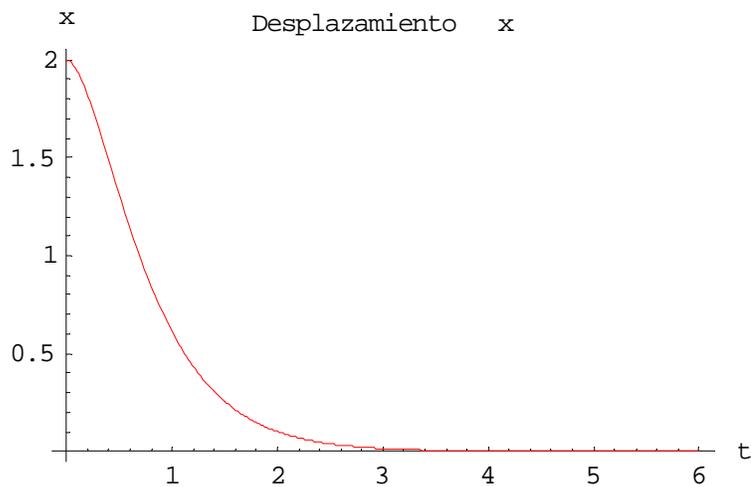
$$\ddot{x} + 5\dot{x} + 6x = 0$$

Aplicando Laplace $s^2 + 5s + 6 = 0$

la cual tiene las raíces $s_1 = -2$ y $s_2 = -3$, por lo tanto la respuesta tiene la forma

$$x(t) = c_1 e^{-2t} + c_2 e^{-3t}$$

Ahora, con las siguientes condiciones iniciales: $x(0) = 2$, $x'(0) = 0$ nos da la siguiente gráfica de comportamiento del sistema. (ver mayor información en programa de Mathematica, Amortiguación 8.1). Ver las *Observaciones* anteriores.



Ejemplo 2 raíces complejas

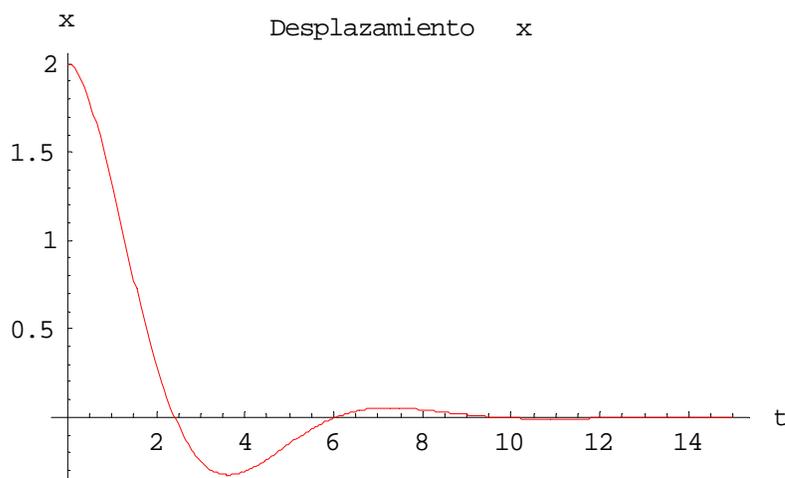
Se desea determinar el movimiento del sistema representado en el esquema anterior Gráfico (8.1) y expresado por la ecuación (8.1), si los valores de los parámetros son: $m = 1$, $b = 1$ y $k = 1$.

La ecuación característica es $s^2 + s + 1 = 0$

La cual tiene las raíces $s_1 = -\frac{1}{2} \pm \frac{\sqrt{3}}{2}i$. Por lo tanto, la respuesta tiene la forma

$$x(t) = e^{-\frac{1}{2}t} \left(c_1 \cos \frac{\sqrt{3}}{2}t + c_2 \operatorname{sen} \frac{\sqrt{3}}{2}t \right)$$

Ahora, con las siguientes condiciones iniciales: $x(0) = 2$, $x'(0) = 0$ nos da la siguiente gráfica de comportamiento del sistema. (ver mayor información en programa de Mathematica, Amortiguación 8.2). Ver las *Observaciones* anteriores.



Ejemplo 3 raíces reales e iguales

Se busca determinar el movimiento del sistema representado en el esquema anterior Gráfico (8.1) y expresado por la ecuación (8.1), si los valores de los parámetros son: $m = 1$, $b = 4$ y $k = 4$.

En el caso de raíces reales e iguales (raíces repetidas), la solución tiene la forma:

$$x(t) = C_1 e^{s_1 t} + C_2 t e^{s_2 t}$$

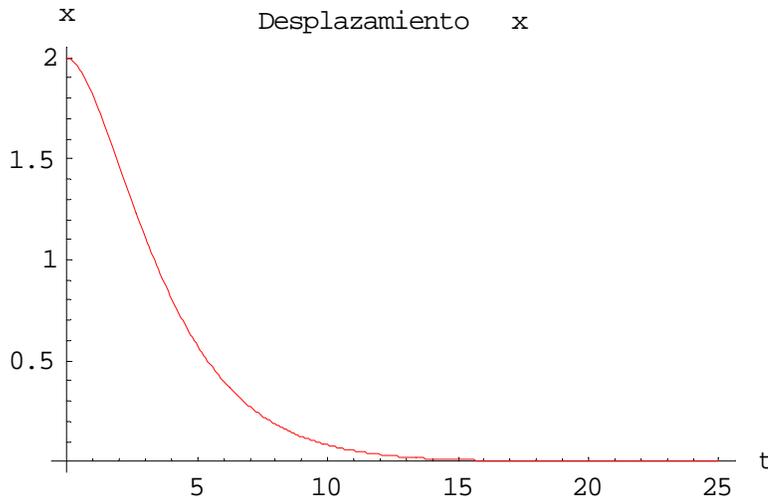
en donde en este caso $s_1 = s_2 = -\frac{b}{2m}$, por lo que la ecuación puede escribirse así

$$x(t) = (C_1 + C_2 t) e^{-\frac{b}{2m} t}$$

La ecuación característica es $s^2 + 4s + 4 = 0$

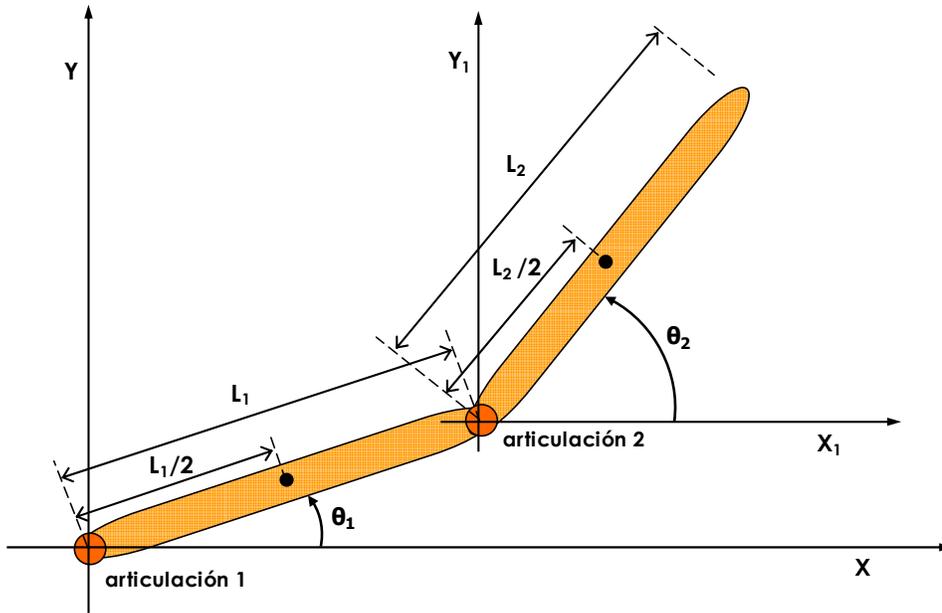
La cual tiene la raíz doble $s_1 = s_2 = -2$.

Ahora, con las siguientes condiciones iniciales: $x(0) = 2$, $x'(0) = 0$ nos da la siguiente gráfica de comportamiento del sistema. (ver mayor información en programa de Mathematica, Amortiguación 8.3). Ver las *Observaciones* anteriores.



8.b. Ecuaciones Euler-Lagrange para un brazo robótico

Deducción de las ecuaciones de movimiento de Euler-Lagrange para un brazo robótico de dos eslabones con movimiento planar. El brazo está fijo en la base y tiene dos actuadores, uno en cada una de las articulaciones.



BRAZO ROBOTICO DE DOS ESLABONES ARTICULADOS

Para el primer eslabón del brazo se considera que las componentes horizontal y vertical de un vector denotado como **scm1** al centro de masa "m1" con la longitud del eslabón **L1**, corresponden a $\frac{L1}{2} \cos \theta_1$ y $\frac{L1}{2} \text{sen} \theta_1$. Para el segundo eslabón del brazo se considera que las componentes horizontal y vertical de un vector denotado como **scm2** al centro de masa "m2" con la longitud del eslabón **L2**, corresponden a $L1 \cos \theta_1 + \frac{L2}{2} \cos \theta_2$ y $L1 \text{sen} \theta_1 + \frac{L2}{2} \text{sen} \theta_2$.

$$scm1 = \left\{ \frac{1}{2} L1 \cos(\theta_1), \frac{1}{2} L1 \text{sen}(\theta_1) \right\}$$

$$scm2 = \left\{ L1 \cos(\theta_1) + \frac{1}{2} L2 \cos(\theta_2), L1 \text{sen}(\theta_1) + \frac{1}{2} L2 \text{sen}(\theta_2) \right\}$$

Derivando con respecto al tiempo los vectores

$$\frac{d}{dt}(scm1) = \left\{ -\frac{1}{2} L1 \text{sen}(\theta_1) \dot{\theta}_1, \frac{1}{2} L1 \cos(\theta_1) \dot{\theta}_1 \right\}$$

$$\frac{d}{dt}(scm2) = \left\{ -L1 \text{sen}(\theta_1) \dot{\theta}_1 - \frac{1}{2} L2 \text{sen}(\theta_2) \dot{\theta}_2, \frac{1}{2} L1 \cos(\theta_1) \dot{\theta}_1 + \frac{1}{2} L2 \cos(\theta_2) \dot{\theta}_2 \right\}$$

Ahora se determina la energía cinética debida a la traslación $T = \frac{1}{2} mv^2$

$$T_{tr1} = \frac{1}{2} m1 (d(scml))^2 = \frac{1}{8} \dot{\theta}_1^2 L1^2 m1$$

$$T_{tr2} = \frac{1}{2} m_2 (d(scm_2))^2 = \frac{1}{8} m_2 (4\dot{\theta}_1^2 L_1^2 + \dot{\theta}_2^2 L_2^2 + 4\dot{\theta}_1 \dot{\theta}_2 L_1 L_2 \cos(\theta_1 - \theta_2))$$

Los momentos de inercia son:

$$I_1 = \frac{L_1^2 m_1}{12} \quad I_2 = \frac{L_2^2 m_2}{12}$$

A continuación se determina la energía cinética debida a la rotación en el sistema

$$T_{rot1} = \frac{1}{2} I_1 (\dot{\theta}_1^2) = \frac{1}{24} \dot{\theta}_1^2 (L_1^2) m_1$$

$$T_{rot2} = \frac{1}{2} I_2 (\dot{\theta}_2^2) = \frac{1}{24} \dot{\theta}_2^2 (L_2^2) m_2$$

La energía cinética total del sistema esta dada por $T = T_{tr1} + T_{rot1} + T_{tr2} + T_{rot2}$

$$T = \frac{1}{6} \dot{\theta}_1^2 L_1^2 m_1 + \frac{1}{24} \dot{\theta}_2^2 L_2^2 m_2 + \frac{1}{8} m_2 (4\dot{\theta}_1^2 L_1^2 + \dot{\theta}_2^2 L_2^2 + 4\dot{\theta}_1 \dot{\theta}_2 (L_1)(L_2) \cos(\theta_1 - \theta_2))$$

La energía potencial esta expresada por $V = mgh$, así en este caso es

$$V = \frac{1}{2} g (L_1)(m_1) \text{sen}(\theta_1) + g (m_2) \left((L_1) \text{sen}(\theta_1) + \frac{1}{2} (L_2) \text{sen}(\theta_2) \right)$$

El Lagrangiano ($\mathbf{L} = \mathbf{T} - \mathbf{V}$) del sistema queda

$$L = \frac{1}{6} \left(\dot{\theta}_2^2 (L_2^2) m_2 + \dot{\theta}_1^2 (L_1^2) (m_1 + 3m_2) + 3\dot{\theta}_1 (\dot{\theta}_2) (L_1)(L_2) m_2 \cos(\theta_1 - \theta_2) \right) - 3g (L_1(m_1 + 2m_2) \sin(\theta_1) + L_2 m_2 \sin(\theta_2))$$

Derivando para deducir las ecuaciones de movimiento

$$\frac{\partial L}{\partial \theta_1} = -\frac{1}{2} g (L_1)(m_1) \cos(\theta_1) - g (L_1)(m_2) \cos(\theta_1) - \frac{1}{2} (\theta_1)(\theta_2) (L_1)(L_2) (m_2) \text{sen}(\theta_1 - \theta_2)$$

$$\frac{\partial L}{\partial \theta_2} = -\frac{1}{2} g (L_2)(m_2) \cos(\theta_2) + \frac{1}{2} (\theta_1)(\theta_2) (L_1)(L_2) (m_2) \text{sen}(\theta_1 - \theta_2)$$

Las ecuaciones de movimiento de Euler-Lagrange quedan como:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} = \frac{1}{6} L_1 \left(2\ddot{\theta}_1 (L_1) m_1 + 6\ddot{\theta}_1 (L_1) m_2 + 3g (m_1 + 2m_2) \cos(\theta_1) + 3\ddot{\theta}_2 (L_2) (m_2) \cos(\theta_1 - \theta_2) + 3\dot{\theta}_2^2 (L_2) (m_2) \text{sen}(\theta_1 - \theta_2) \right)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} = \frac{1}{6} L_2 m_2 \left(2\ddot{\theta}_2 (L_2) + 3\ddot{\theta}_1 (L_1) \cos(\theta_1 - \theta_2) + 3g \cos(\theta_2) - 3\dot{\theta}_1^2 (L_1) \text{sen}(\theta_1 - \theta_2) \right)$$

considerando los torques de entradas

$$\frac{1}{6} L_1 \left(2\ddot{\theta}_1 (L_1) m_1 + 6\ddot{\theta}_1 (L_1) m_2 + 3g (m_1 + 2m_2) \cos(\theta_1) + 3\ddot{\theta}_2 (L_2) (m_2) \cos(\theta_1 - \theta_2) + 3\dot{\theta}_2^2 (L_2) (m_2) \text{sen}(\theta_1 - \theta_2) \right) = \tau_1$$

$$\frac{1}{6}L_2m_2 \left(\begin{array}{l} 2\ddot{\theta}_2(L_2) + 3\ddot{\theta}_1(L_1)\cos(\theta_1 - \theta_2) \\ + 3g\cos(\theta_2) - 3\dot{\theta}_1^2(L_1)\sin(\theta_1 - \theta_2) \end{array} \right) = \tau_2$$

y así el Modelo de la Planta considerando la Fricción resulta:

$$\frac{1}{6}L_1 \left(\begin{array}{l} 2\ddot{\theta}_1(L_1)m_1 + 6\ddot{\theta}_1(L_1)m_2 + 3g(m_1 + 2m_2)\cos(\theta_1) \\ + 3\ddot{\theta}_2(L_2)(m_2)\cos(\theta_1 - \theta_2) + 3\dot{\theta}_2^2(L_2)(m_2)\sin(\theta_1 - \theta_2) \end{array} \right) + \alpha_0 + \alpha_1 e^{-\beta_1|\dot{\theta}_1|} + \alpha_2 \left(1 - e^{-\beta_2|\dot{\theta}_1|} \right) \cdot \text{sgn}(\dot{\theta}_1) = \tau_1$$

$$\frac{1}{6}L_2m_2 \left(\begin{array}{l} 2\ddot{\theta}_2(L_2) + 3\ddot{\theta}_1(L_1)\cos(\theta_1 - \theta_2) \\ + 3g\cos(\theta_2) - 3\dot{\theta}_1^2(L_1)\sin(\theta_1 - \theta_2) \end{array} \right) + \alpha_0 + \alpha_1 e^{-\beta_1|\dot{\theta}_2|} + \alpha_2 \left(1 - e^{-\beta_2|\dot{\theta}_2|} \right) \cdot \text{sgn}(\dot{\theta}_2) = \tau_2$$

8.c. Dinámica del brazo robótico

La dinámica de un robot manipulador de n – articulaciones puede ser expresada en la forma Euler-Lagrange de la siguiente forma matricial:

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \quad \dots\dots\dots (8.2)$$

con $q(t) \in R^n$ el vector de juntas variables, $M(q)$ la matriz de inercias, $V_m(q, \dot{q})$ la matriz de fuerzas de coriolis/centrípeta, $G(q)$ el vector de gravedad y $F(\dot{q})$ la fuerza de fricción. Las perturbaciones desconocidas acotadas (incluyendo dinámicas no modeladas sin estructurar) son denotadas por τ_d y el torque de control de entradas es $\tau(t)$.

Dada una trayectoria deseada del brazo $q_d(t) \in R^n$ el error de seguimiento es

$$e(t) = q_d(t) - q(t) \quad \dots\dots\dots (8.3)$$

En uso estándar para robots el error de seguimiento filtrado es

$$r = \dot{e} + \Lambda e \quad \dots\dots\dots (8.4)$$

donde $\Lambda = \Lambda^T > 0$ es una matriz parámetro de diseño, usualmente seleccionada diagonal. Diferenciando $r(t)$ y usando (8.2), la dinámica del brazo puede escribirse en términos del error de seguimiento filtrado como:

$$M\dot{r} = -V_m r - \tau + f + \tau_d \quad \dots\dots\dots (8.5)$$

donde la función no-lineal del robot es

$$f(x) = M(q)(\ddot{q}_d + \Lambda \dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}) \quad \dots\dots\dots (8.6)$$

y, por ejemplo, podemos seleccionar un vector x de la siguiente manera

$$x = [e^T, \dot{e}^T, q_d^T, \dot{q}_d^T, \ddot{q}_d^T]^T \dots\dots\dots (8.7)$$

La función $f(x)$ normalmente es desconocida ó conocida de manera incompleta. Debe ser computada o compensada por el controlador de retroalimentación. Hay que notar que $f(x)$ es discontinua ya que contiene términos de fricción desconocidos.

Definiendo ahora un torque de control de entrada como:

$$\tau_0 = \hat{f} + K_v r \dots\dots\dots (8.8)$$

la matriz de ganancia $K_v = K_v^T > 0$ y $\hat{f}(x)$ como un estimado de $f(x)$. La estructura de esta entrada de control es importante ya que determina la estructura del sistema de control por retroalimentación. Hay que notar que contiene un término de control proporcional derivativo *PD*.

$$K_v r = K_v \dot{e} + K_v \Lambda e \dots\dots\dots (8.9)$$

El *PD* es el control estandar para el diseño lazos cerrados de seguimiento de movimiento para sistemas industriales. Sin embargo, esto también tiene un término de compensación $\hat{f}(x)$ que va a ser usado para compensar los efectos de no-linealidades desconocidas.

El sistema de control debe de alguna manera proporcionar este estimado $\hat{f}(x)$. Para verificar la estabilidad del diseño del controlador hay que examinar la dinámica del lazo cerrado del error que está dada por:

$$M\dot{r} = -(K_v + V_m)r + \hat{f} + \tau_d \dots\dots\dots (8.9)$$

donde la funcional de la estimación del error es:

$$\tilde{f} = f - \hat{f} \dots\dots\dots (8.10)$$

Hay que notar que de acuerdo a (8.4) tenemos

$$\frac{e}{r} = (sI + \Lambda)^{-1} \quad \frac{\dot{e}}{r} = s(sI + \Lambda)^{-1} \dots\dots\dots (8.11)$$

Así que $\|e\| \leq \|(sI + \Lambda)^{-1}\| \|r\| \quad \|\dot{e}\| \leq \|s(sI + \Lambda)^{-1}\| \|r\| \dots\dots\dots (8.12)$

$$\|e\| \leq \frac{\|r\|}{\sigma_{\min}(\Lambda)} \quad \|\dot{e}\| \leq \|r\| \dots\dots\dots (8.13)$$

Con $\sigma_{\min}(\Lambda)$ el mínimo valor singular de Λ . Esto significa que tanto el error de seguimiento $e(t)$ como su derivada $\dot{e}(t)$ están acotadas simplemente por el diseño del control, ecuación (8.8), de tal manera que el error filtrado $r(t)$ está acotado, reduciendo la complejidad del problema de diseño.

Las siguientes propiedades estándar de la dinámica del robot son requeridas y se mantienen para cualquier brazo robótico de elementos rígidos que giran.

Propiedad 1: $M(q)$ es una matriz simétrica definida positiva acotada por

$$m_1 I \leq M(q) \leq m_2 I$$

con m_1, m_2 constantes positivas conocidas.

Propiedad 2: $V_m(q, \dot{q})$ está limitada por $v_b(q) \|\dot{q}\|$, con $v_b(q) \in C^1(S)$

Propiedad 3: La matriz $\dot{M} - 2V_m$ es anti-simétrica.

Propiedad 4: Las perturbaciones desconocidas satisfacen $\|\tau_d\| < b_d$, con b_d una constante positiva conocida.

8.d. Diseño del controlador de retroalimentación lineal tipo PD

Se utiliza para controlar la planta como primera aproximación un Controlador de Retroalimentación Lineal del tipo *PD* cuyas siglas significan Proporcional-Derivativo y que es el más aplicado en la Industria.

Como primer paso para el análisis que determina los valores de las ganancias mínimos de K_p y K_d que se requieren en el Controlador *PD*, para que logre que la planta alcance una posición deseada, para lo cual se toma como ejemplo el sistema de un brazo robótico de un eslabón.

El siguiente es un caso particular de un *Brazo Robótico de un eslabón* considerando la horizontal como su eje de referencia. La ecuación de su movimiento queda representada por:

$$3.67875 \cos \theta_1 + 0.125 \ddot{\theta}_1 = \tau \quad \dots\dots\dots (8.d.1)$$

Posición deseada

según el control *PD* para una posición deseada $= \frac{\pi}{2}$ con velocidad deseada = 0 y con Condiciones

Iniciales: posición $= \frac{3\pi}{2}$ y velocidad = 0. Tenemos

$$e = \theta_{1d} - \theta_1$$

$$r = e + K_d(de) \quad ; \quad de = -\dot{\theta}_1$$

$$r = \frac{\pi}{2} - \theta_1 + K_d(-\dot{\theta}_1) \quad ; \quad \tau = K_p(r) \dots\dots\dots (8.d.2)$$

Valores propuestos para K_p y K_d

Aquí se considera que el valor de K_p debe ser mayor a V el cual es el coeficiente del término de Coriolis. En este caso particular $V = 3.67875$ y el valor de K_d se ajusta a una regla empírica que estima un valor de entre 5 y 20% del valor de K_p . Se proponen los siguientes valores para las ganancias $K_p = 4$ y $K_d = 0.25$ y se sustituyen en la ecuación (8.d.1).

$$3.67875 \cos \theta_1 + 0.125\ddot{\theta}_1 = 4\left(\frac{\pi}{2} - \theta_1 - 0.25\dot{\theta}_1\right) \dots\dots\dots (8.d.3)$$

$$0.125\ddot{\theta}_1 = 6.2831 - 4\theta_1 - \dot{\theta}_1 - 3.67875 \cos \theta_1$$

Simplificando y despejando $\ddot{\theta}_1$ se tiene:

$$\ddot{\theta}_1 = 50.2654 - 32\theta_1 - 8\dot{\theta}_1 - 29.43 \cos \theta_1 \dots\dots\dots (8.d.4)$$

ahora se establece un sistema de ecuaciones de estado

$$\theta_1 = x_1 \quad ; \quad \dot{\theta}_1 = x_2$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -32x_1 - 29.43 \cos(x_1) - 8x_2 + 50.2654 \dots (8.d.5) \text{ (sistema de estado con controlador PD)}$$

Calculando el Jacobiano del sistema (8.d.5)

$$Df = \begin{pmatrix} 0 & 1 \\ -32 + 29.43 \text{sen}(x_1) & -8 \end{pmatrix}$$

evaluando en las condiciones deseadas

$$Df = \begin{pmatrix} 0 & 1 \\ -32 + 29.43 \text{sen}\left(\frac{\pi}{2}\right) & -8 \end{pmatrix}$$

Resulta una matriz A

$$A = \begin{pmatrix} 0 & 1 \\ -2.57 & -8 \end{pmatrix}$$

La matriz tiene traza negativa y su determinante es positivo, por estas condiciones se trata de punto sumidero y se deduce que **el sistema es estable**. Como el sistema anterior es estable el Controlador **PD** hará que la planta cumpla con las condiciones de posición y velocidad deseadas.

Analisis

Bien, ahora se analizará el porqué de los resultados anteriores. Retomando la ecuación (8.d.3)

$$3.67875 \cos \theta_1 + 0.125 \ddot{\theta}_1 = K_p \left[\left(\frac{\pi}{2} - \theta_1 \right) + K_d (-\dot{\theta}_1) \right]$$

Cambiando los coeficientes numéricos por letras

$$3.67875 \cos \theta_1 + 0.125 \ddot{\theta}_1 = K_p \left(\frac{\pi}{2} \right) - K_p \theta_1 - K_p K_d \dot{\theta}_1$$

\downarrow
 b

\downarrow
 a

$$\frac{a \ddot{\theta}_1}{a} = -\frac{b \cos \theta_1}{a} - \frac{K_p \theta_1}{a} - \frac{K_p K_d \dot{\theta}_1}{a} + \frac{K_p \left(\frac{\pi}{2} \right)}{a}$$

el factor a es un escalar que divide toda la ecuación para simplificar el primer término.

$$\ddot{\theta}_1 = \frac{K_p \left(\frac{\pi}{2} \right)}{a} - \frac{K_p \theta_1}{a} - \frac{K_p K_d \dot{\theta}_1}{a} - \frac{b \cos(\theta_1)}{a}$$

ahora se establece un sistema de ecuaciones de estado

$$\theta_1 = x_1 \quad ; \quad \dot{\theta}_1 = x_2$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{K_p x_1}{a} - \frac{b \cos(x_1)}{a} - \frac{K_p K_d x_2}{a} + \frac{K_p \left(\frac{\pi}{2} \right)}{a}$$

$$Df = \begin{pmatrix} 0 & 1 \\ \frac{b \operatorname{sen} \left(\frac{\pi}{2} \right)}{a} - \frac{K_p}{a} & -\frac{K_p K_d}{a} \end{pmatrix}$$

elementos de la matriz $\mathbf{a}_{11}=0$, $\mathbf{a}_{12} = 1$, $\mathbf{a}_{21} = \frac{b \operatorname{sen} \left(\frac{\pi}{2} \right)}{a} - \frac{K_p}{a}$, $\mathbf{a}_{22} = -\frac{K_p K_d}{a}$

Observaciones

1° El elemento \mathbf{a}_{22} es un polo por lo cual debe ser negativo, para esto K_p y K_d deben ser del mismo signo.

2° Para que la matriz tenga una trayectoria de solución estable el elemento \mathbf{a}_{21} debe ser negativo por lo cual la ganancia K_p debe ser positiva y además se debe cumplir $K_p > |b|$.

3° Por lo tanto K_p y K_d siempre deben ser positivas.

4° Una regla empírica indica que el valor de K_d puede estar entre el 5% y el 20% del valor de K_p .

NOTA RELEVANTE

En robots manipuladores de dos brazos o más, la Estabilidad Asintótica de manera lineal en todos los brazos en el sentido estricto, es decir a cero, es una tarea difícil de alcanzar porque aparecen los términos compuestos en la ecuación del sistema debidos a la acción de los brazos un(os) sobre el(los) otro(s), así que, es mas factible que el sistema alcance la estabilidad en condiciones menos severas. En puntos de estado de equilibrio, sin perturbaciones, es más práctico lograr que en el sistema se logre la convergencia del error de manera oscilante decreciente hasta el grado de precisión deseado y en otros casos que se alcance un estado estable Acotado Uniforme y Ultimamente *UUB* .

Para lograr lo dicho anteriormente, los valores de las ganancias K_p requieren estar varias veces mas arriba de los coeficientes de la Matriz de Coriolis V . El ajuste del acotamiento de la función estará en función al grado de precisión que se desee tener.

8.e. Simulaciones y Observaciones.

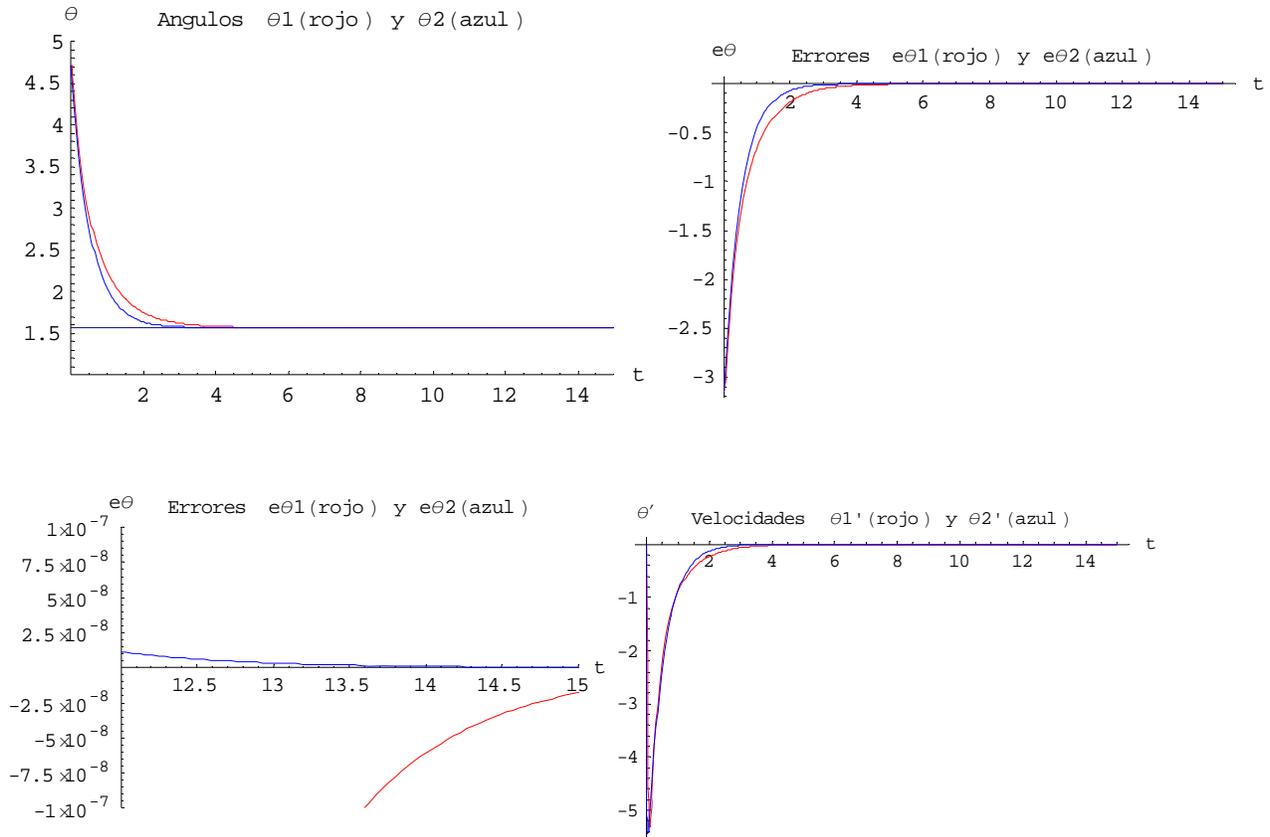
Se realizaron varios programas en Matemática para cubrir los casos que sirven al análisis. Entre las variantes se incluye el tener o no fricción, alcanzar una posición deseada o el seguimiento de trayectorias de referencia. La parte transitoria queda definida desde el inicio hasta que llega el sistema al estado estable.

Simulación 8.1.

Brazo con dos eslabones, sin fricción, posición deseada = vertical eslabones hacia arriba.

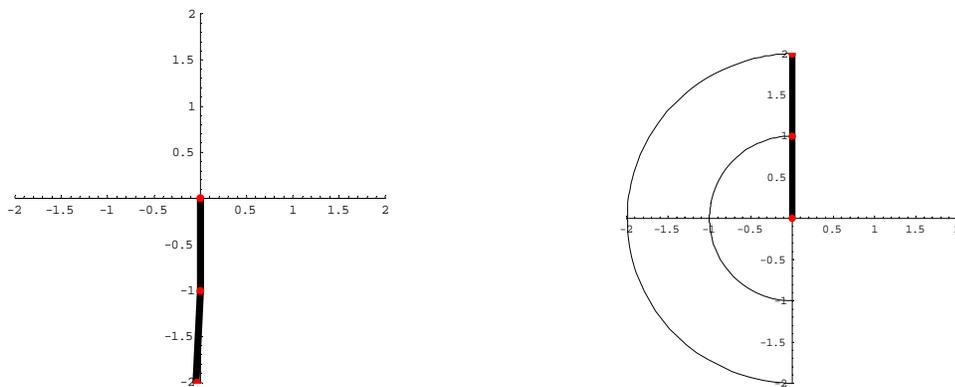
Programa de Mathematica **PD_2e_SFricción_posición**

Diagonal $K = 20$, Diagonal $\Lambda = 0.5$ tiempo 15 seg.



Observaciones:

Se observa un buen comportamiento transitorio y una buena convergencia del error en el estado estable, no hay oscilaciones indeseables, se logra una buena precisión al tiempo final con valores a centésimas de micra de radian. No hay alteraciones indeseables en las velocidades-



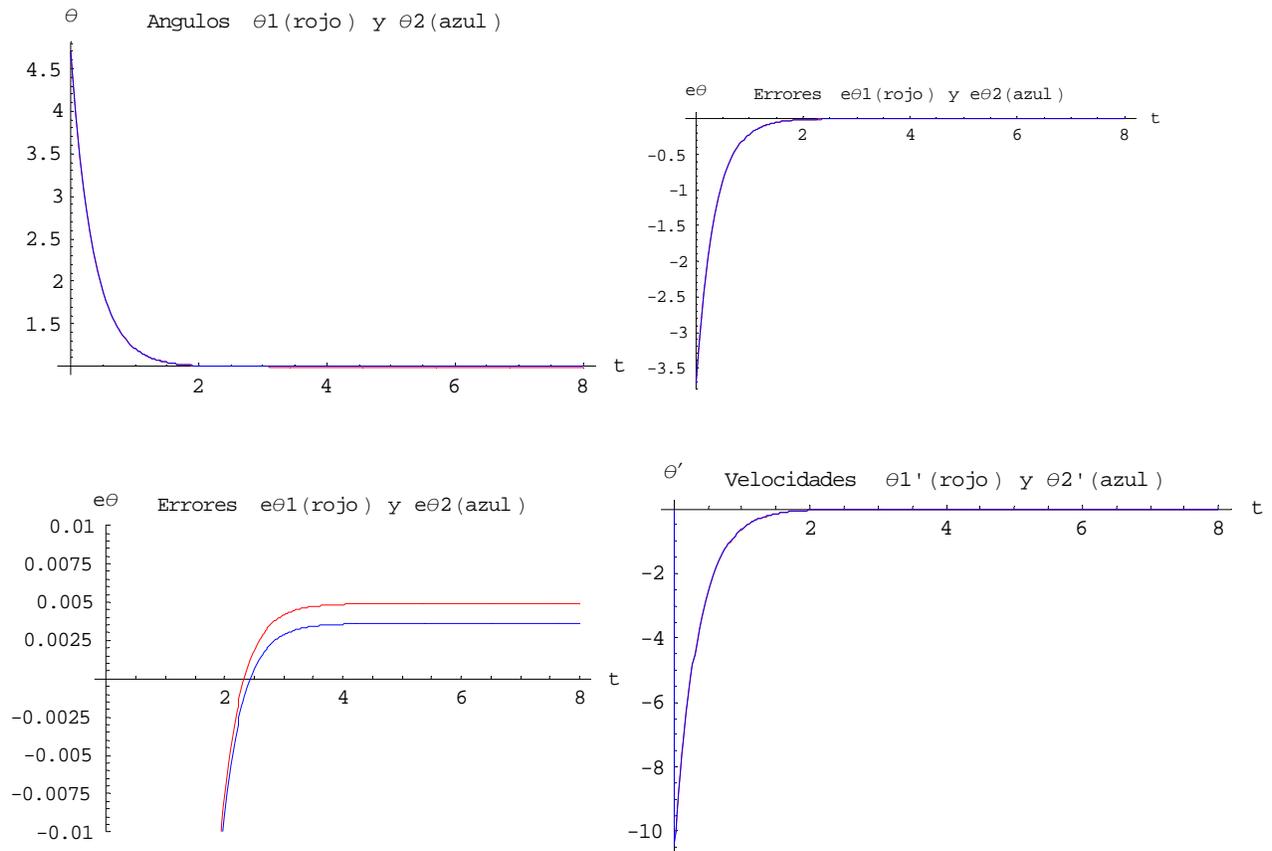
Gráfica de la Simulación de la trayectoria que sigue el brazo para alcanzar la posición deseada

Simulación 8.2

Brazo con dos eslabones, sin fricción, posición deseada: a 1 radian a partir de la horizontal

Programa de Mathematica **PD_2e_SFricción_posición_2**

Diagonal $K = 900, 200$, Diagonal $\Lambda = 0.35$



Observaciones:

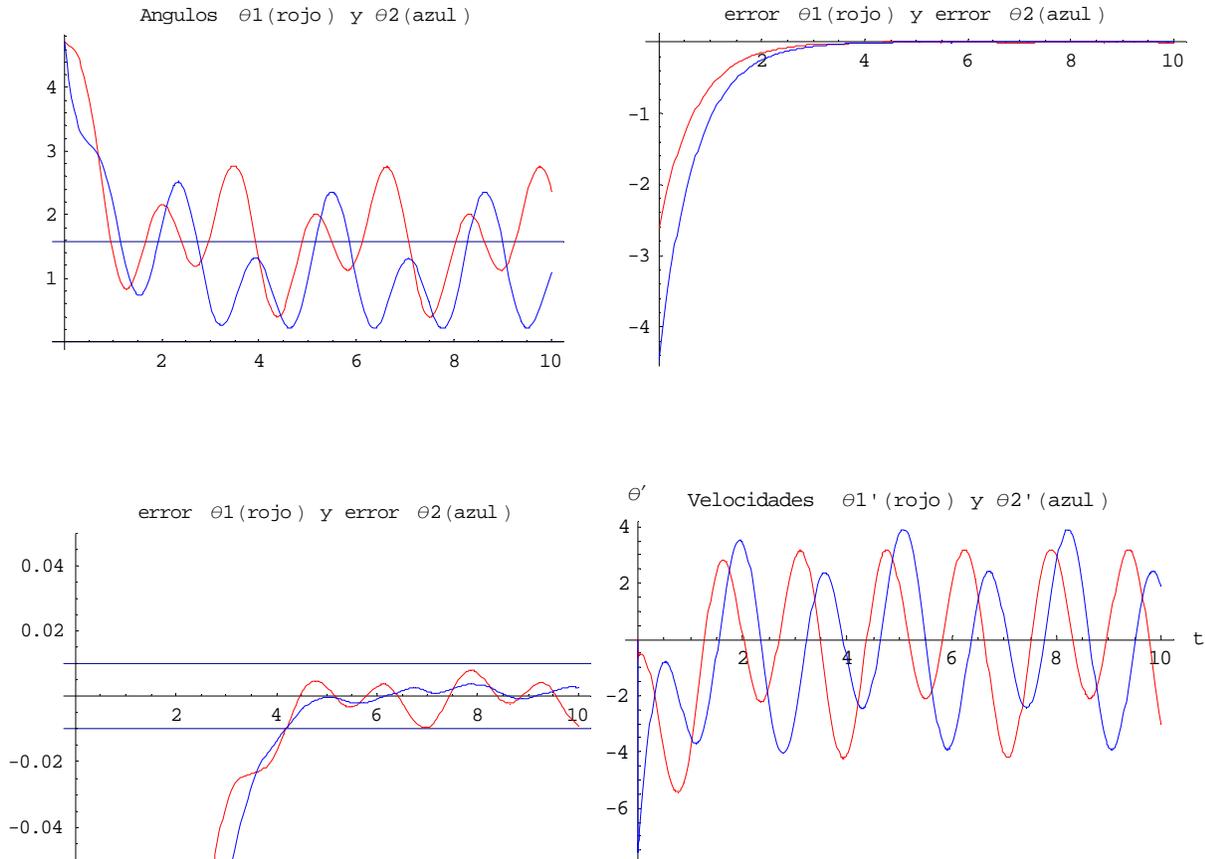
Se observa un buen comportamiento en la sección transitoria. El sistema alcanza un estado estable acotado, con una precisión del error a milésimas de radian. El controlador PD tiene valores de las ganancias que se incrementan sustancialmente respecto al ejemplo anterior, ya que debe mantenerse en esta posición que no es un punto de equilibrio.

Simulación 8.3

Brazo con dos eslabones, sin fricción, seguimiento de trayectoria de referencia I, la roseta

Programa de Mathematica **PD_2e_SFricción_trayectoria_I**

Diagonal $K = 600,350$, Diagonal $\Lambda = 0.7$



Observaciones:

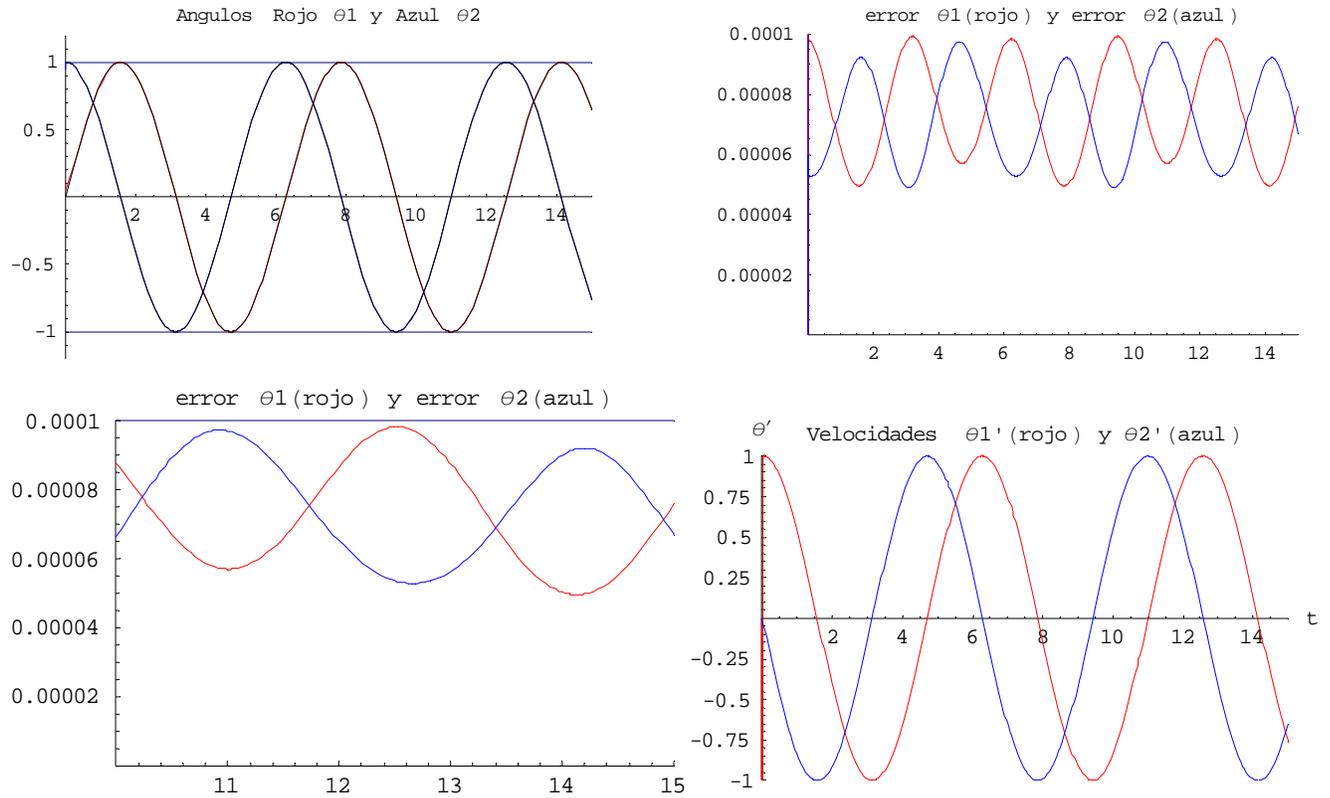
Se observa un buen comportamiento transitorio con ondulaciones suaves. El sistema converge al estado Estable Acotado Uniforme y Último y se logra una precisión a milésimas de radián. Las velocidades no presentan anomalías a este grado de precisión. No puede reducirse el margen de error en este tipo de seguimiento de trayectoria sin que se presenten alteraciones en las velocidades.

Simulación 8.4

Brazo con dos eslabones, sin fricción, seguimiento de trayectoria de referencia Seno-Coseno

Programa de Mathematica **PD_Sfricción_tray.Seno-Coseno_Aprox.Final**

$$K = 82000, 14000, \text{ Diagonal } \Lambda = 0.003$$



Observaciones:

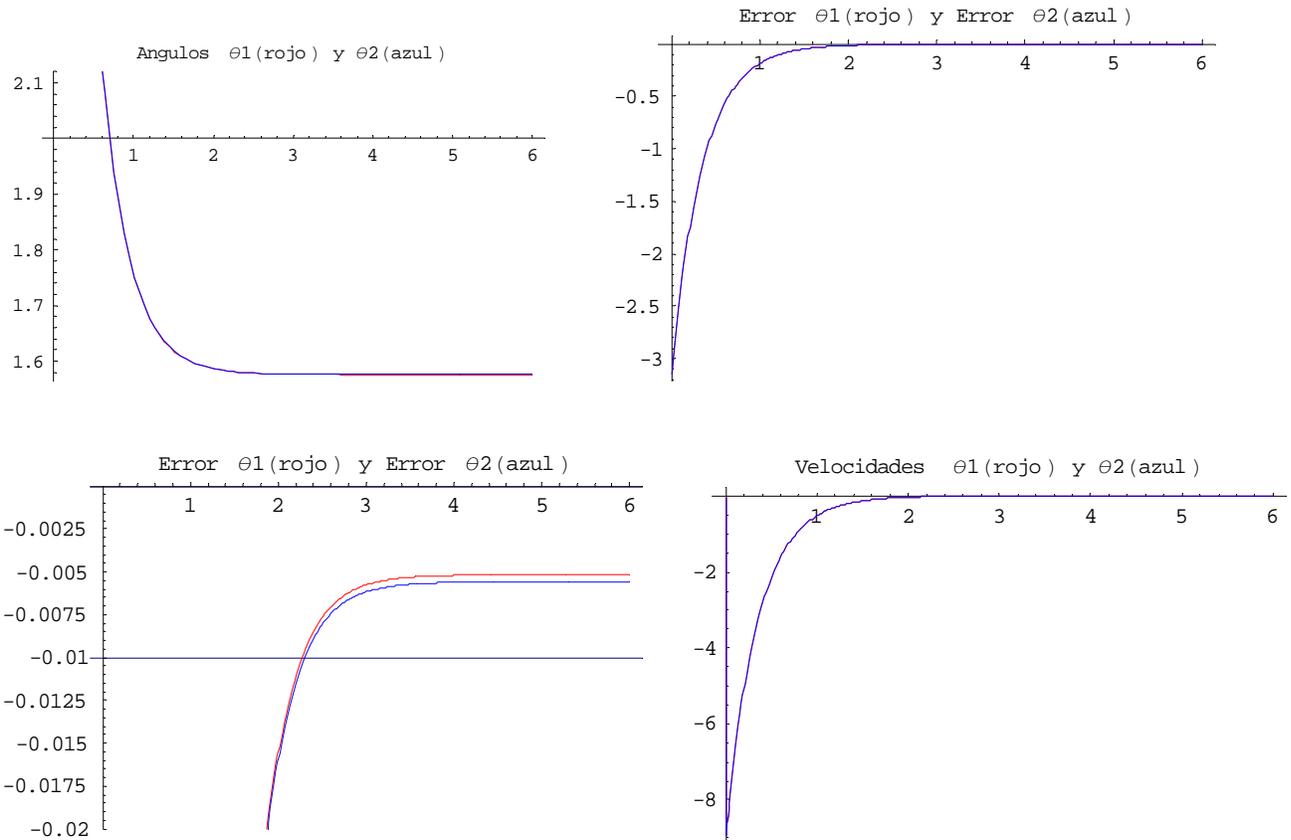
En esta simulación se reduce el margen de error a diezmilésimas de radian ajustando los valores de las ganancias. Como no tiene perturbaciones modela sin alteraciones las funciones de referencia.

Simulación 8.5

Brazo con dos eslabones, con fricción, posición deseada arriba.

Programa de Mathematica **PD_2e_Cfricción_posición**

Diagonal $K = 7000$, Diagonal $\Lambda = 0.35$



Observaciones

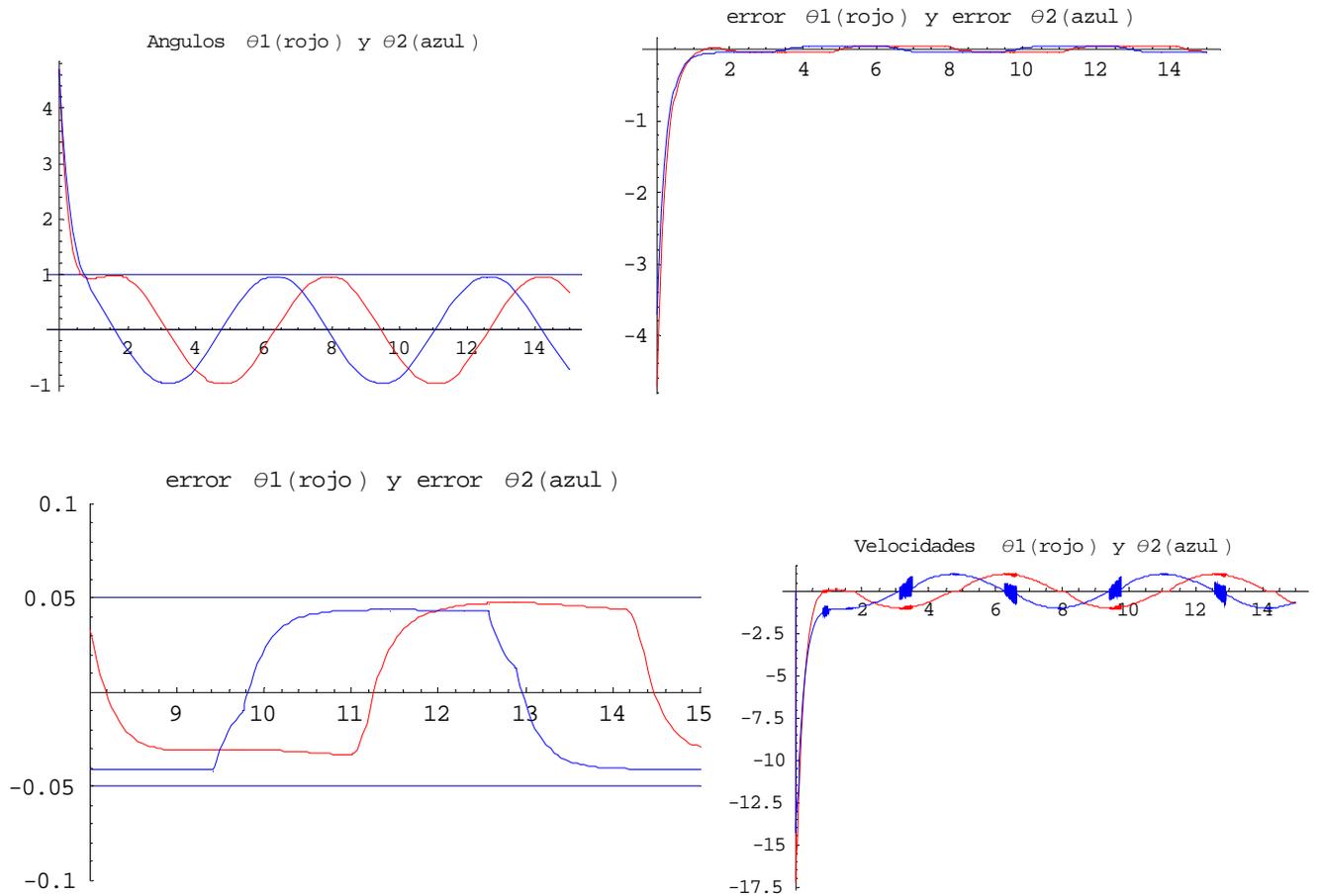
Para tener una convergencia a milésimas de radián fue necesario incrementar mucho el valor del coeficiente de ganancia K ; esto por que hay mantener una posición y rechazar el efecto de la fricción. Se alcanza una Estabilidad Acotada Uniforme y Última, aunque sólo por un lapso de hasta 10 seg., después se vuelve inestable.

Simulación 8.6

Brazo con dos eslabones, Con Fricción, seguimiento de trayectoria de referencia II.

Programa de Mathematica **PD_2e_CFricción_trayectoria_II**

Diagonal $K = 920$, Diagonal $\Lambda = 0.25$



Observaciones:

La precisión del seguimiento alcanzada a poco menos de 5 centésimas de radian no es adecuada sin embargo evidencia que empiezan a aparecer anomalías en la gráfica de velocidades lo cual indica que no modela las no linealidades causadas por la perturbación.

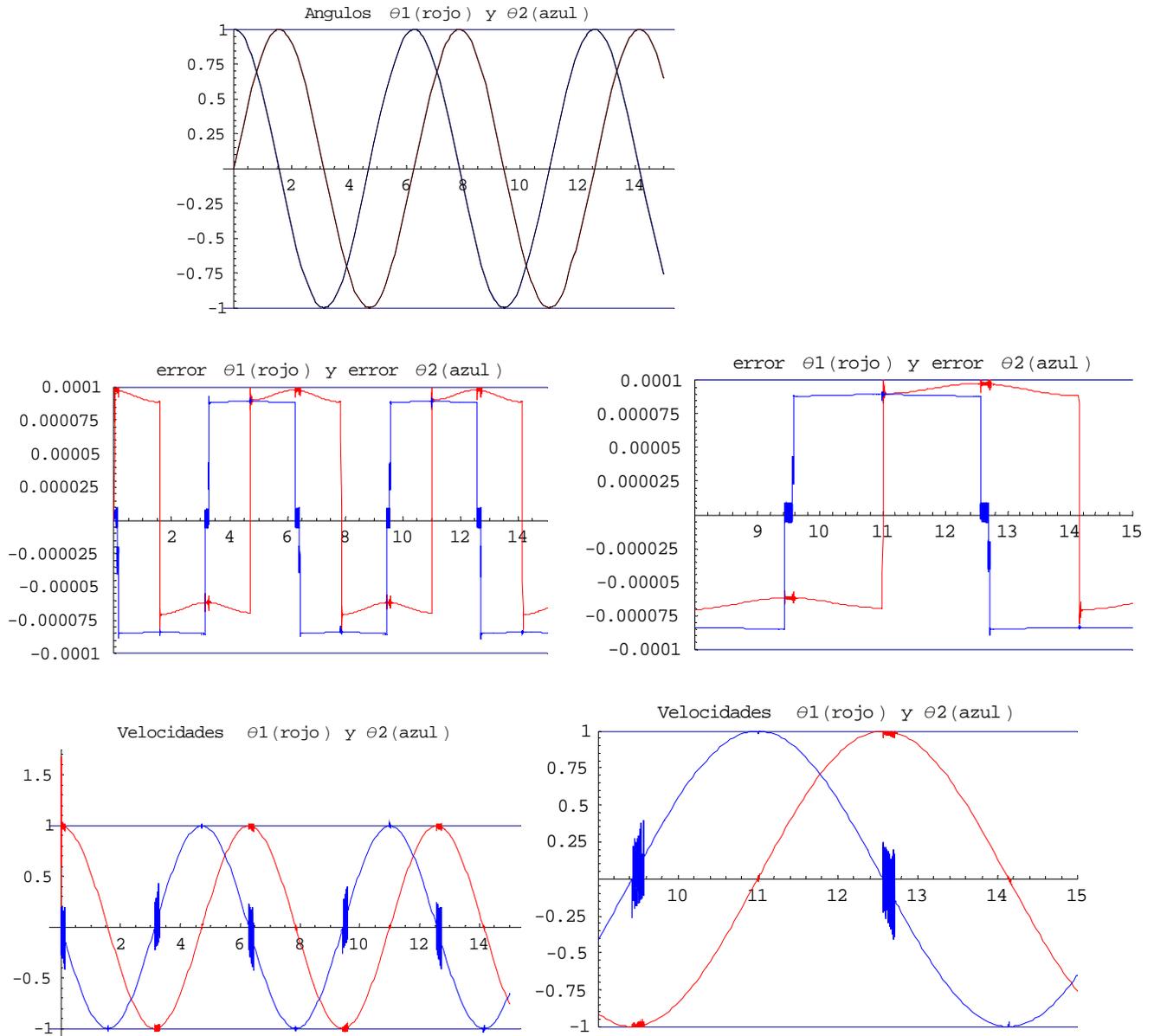
Observando la gráfica del error, por la forma que adquiere, nos indica que el Controlador PD no modela la perturbación de la fricción, sólo la rechaza, conforme al nivel de seguimiento que le es posible alcanzar.

Simulación 8.7

Brazo con dos eslabones, Con Fricción, seguimiento de trayectoria de referencia II.

Programa de Mathematica **PD_CFricción_tray.Seno-Coseno_Aprox.Final**

Diagonal $K = 450000$, Diagonal $\Lambda = 0.0003$



Observaciones:

Al llegar a esta tolerancia del error el Controlador PD produce zonas críticas que muestran debilidad en el sistema que pueden llegar a ocasionar inestabilidad. Las gráficas del error muestran que no modela las no linealidades de la fricción, solo las rechaza. Se presentan alteraciones en determinados puntos en las gráficas del error y de las velocidades donde la no linealidad de la fricción se presenta.

9. CONTROLADOR PD Y RED NEURONAL

9.a. Estructura del controlador PD y red neuronal

Ahora se deriva un controlador de Red Neuronal para la dinámica del robot definida anteriormente. Se propone el algoritmo de ajuste de pesos usual de retropropagación. Se verá que con el ajuste de retropropagación la Red Neuronal sólo puede garantizar el adecuado funcionamiento en lazo cerrado bajo condiciones ideales (no realistas). Un algoritmo de ajuste modificado es propuesto mas adelante en el siguiente inciso, para el funcionamiento del controlador de Red Neuronal bajo condiciones realistas. Así, se supone que la función no-lineal del robot expresada por la ecuación (8.6)

$$f(x) = M(q)(\ddot{q}_d + \Lambda \dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(\dot{q}) \dots\dots\dots (8.6)$$

es dada por una Red Neuronal como en la ecuación (9.1) para algunas constantes de pesos "ideales" la Red Neuronal en W y V .

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x) \dots\dots\dots (9.1)$$

Donde la reconstrucción del error por la red $\varepsilon(x)$ esta limitada por una constante conocida ε_N . A menos de que la red sea *minima*, los pesos adecuados "ideales" no pueden ser únicos. Los "mejores" pesos pueden entonces ser definidos como aquellos que minimizan el supremo de la norma de $\varepsilon(x)$ sobre S . Este resultado no es la mejor propuesta, como nosotros solo necesitamos saber que tales pesos ideales existen; sus valores actuales no se requieren.

De acuerdo al Teorema 6.d.A este supuesto de aproximación suave siempre se mantiene para funciones continuas.

$$y = W^T \sigma(V^T x) \dots\dots\dots (6.b.3)$$

Teorema 6.d.A

Sea $N_1 = n$, $N_3 = m$ y sea σ cualquier función de aplastamiento. Entonces el rango funcional de la Red Neuronal, ecuación (6.b.3), es denso en $C^m(S)$.

Esto es en severo contraste con el caso para el control adaptativo donde los supuestos de aproximación tales como *LIP*, linealidad-en-los-parámetros, no se pueden conservar. La suavidad de este supuesto es la principal ventaja para el uso de redes no-lineales multicapas sobre las redes lineales de dos capas. Por conveniencia de notación se define la matriz de todos los pesos como:

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \dots\dots\dots (9.2)$$

Los dos primeros supuestos serán verdaderos en cada situación práctica, y son usuales en la literatura existente. Los hechos son fáciles de probar dados los supuestos.

Supuesto 1: Los pesos ideales están acotados por valores positivos conocidos así que

$$\|V\|_F \leq V_M, \|W\|_F \leq W_M \quad \circ$$

$$\|Z\|_F \leq Z_M \dots\dots\dots (9.3)$$

con Z_M conocido.

Supuesto 2: La trayectoria deseada esta limitada en el sentido, por ejemplo,

$$\begin{bmatrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{bmatrix} \leq Q_d \dots\dots\dots (9.4)$$

donde $Q_d \in R$ es una constante conocida.

Hecho 3: Para cada tiempo t , $x(t)$ dado por la ecuación (8.7)

$$x = [e^T, \dot{e}^T, q_d^T, \dot{q}_d^T, \ddot{q}_d^T]^T \dots\dots\dots (8.7)$$

esta acotada por

$$\|x\| \leq c_1 Q_d + c_2 \|r\| \dots\dots\dots (9.5)$$

para constantes positivas computables c_i (c_2 decrece cuando Λ aumenta).

Sean \hat{V}, \hat{W} algunos estimados de valores de pesos ideales, como los provistos, por ejemplo por algoritmos de ajuste de pesos que serán presentados. Se definen las derivaciones de los pesos o los errores de estimación de pesos como:

$$\tilde{V} = V - \hat{V}, \quad \tilde{W} = W - \hat{W}, \quad \tilde{Z} = Z - \hat{Z} \dots\dots\dots (9.6)$$

y el error de salida de la capa oculta para una x dada como:

$$\tilde{\sigma} = \sigma - \hat{\sigma} \equiv \sigma(V^T x) - \sigma(\hat{V}^T x) \dots\dots\dots (9.7)$$

La expansión de las series de Taylor para una x dada puede ser escrita como:

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{V}^T x) \tilde{V}^T x + O(\tilde{V}^T x)^2 \dots\dots\dots (9.8)$$

con $\sigma'(z) \equiv \frac{d\sigma(z)}{dz} \Big|_{z=\hat{z}}$ y $O(z)^2$ denotando términos de segundo orden.

Denotando $\hat{\sigma}' = \sigma'(\hat{V}^T x)$ tenemos:

$$\tilde{\sigma} = \sigma'(\hat{V}^T x) \tilde{V}^T x + O(\tilde{V}^T x)^2 = \hat{\sigma}' \tilde{V}^T x + O(\tilde{V}^T x)^2 \dots\dots\dots (9.9)$$

Se pueden tomar diferentes órdenes de la serie de Taylor dependiendo como se escoja para $\sigma(\bullet)$.
 Nótese que tomamos el siguiente:

$$O(\tilde{V}^T x)^2 = [\sigma(V^T x) - \sigma(\hat{V}^T x)] - \sigma'(\hat{V}^T x) \tilde{V}^T x \dots\dots\dots (9.10)$$

Estructura del Controlador y Dinámica del Sistema de Error

Se define el estimado funcional de la Red Neuronal de (8.6) por

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \dots\dots\dots (9.11)$$

con \hat{V} , \hat{W} los valores actualizados (estimados) de los pesos ideales de la Red Neuronal V, W provistos por los algoritmos de ajuste que mas adelante serán discutidos.

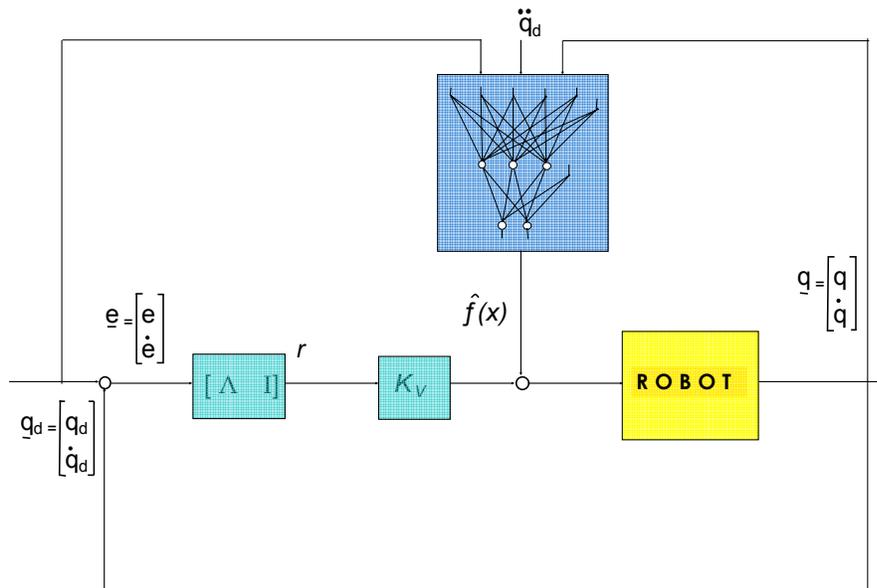
Con τ_0 definido en (8.8)

$$\tau_0 = \hat{f} + K_v r \dots\dots\dots (8.8)$$

se selecciona la entrada de control

$$\tau = \tau_0 - v = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v \dots\dots\dots (9.12)$$

con $v(t)$ una función que será detallada mas adelante y que provee de robustez en el enfrentamiento con los términos de alto orden de la serie de Taylor. La propuesta estructura de control con Red Neuronal se muestra en el Gráfico 9.1 donde $\underline{q} \equiv [q^T, \dot{q}^T]^T$, $\underline{e} \equiv [e^T, \dot{e}^T]^T$.



CONTROLADOR PD Y RED NEURONAL
Caso Ideal **Gráfico 9.1**

Usando este controlador, la dinámica del error filtrado en lazo cerrado se convierte en.

$$M\dot{r} = -(K_v + V_m)r + W^T \sigma(V^T x) - \hat{W}^T \sigma(\hat{V}^T x) + (\varepsilon + \tau_d) + v$$

Adicionando y sustrayendo $W^T \hat{\sigma}$ se produce:

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \hat{\sigma} + W^T \tilde{\sigma} + (\varepsilon + \tau_d) + v$$

con $\hat{\sigma}$ y σ definida en (9.7). Adicionando y sustrayendo ahora $\hat{W}^T \tilde{\sigma}$ nos da

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \hat{\sigma} + \hat{W}^T \tilde{\sigma} + \tilde{W}^T \tilde{\sigma} + (\varepsilon + \tau_d) + v \dots\dots\dots (9.13)$$

El paso clave está en el uso ahora de las series de aproximación de Taylor (9.9) para $\tilde{\sigma}$ de acuerdo con el cual el sistema del error en lazo cerrado es

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \hat{\sigma} + \hat{W}^T \tilde{\sigma} \tilde{V}^T x + w_1 + v \dots\dots\dots (9.14)$$

donde los términos de perturbación son

$$w_1(t) = \tilde{W}^T \hat{\sigma} \tilde{V}^T x + W^T O(\tilde{V}^T x)^2 + (\varepsilon + \tau_d) \dots\dots\dots (9.15)$$

Desafortunadamente, usando este sistema del error no se produce un conjunto compacto exterior en el cual una cierta función derivada de Lyapunov es negativa. Por lo tanto finalmente escribimos el sistema del error como:

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T (\hat{\sigma} - \hat{\sigma} \hat{V}^T x) + \hat{W}^T \hat{\sigma} \tilde{V}^T x + w + v \equiv -(K_v + V_m)r + \zeta_1 \dots\dots\dots (9.16)$$

donde los términos de perturbación son:

$$w(t) = \tilde{W}^T \hat{\sigma} \tilde{V}^T x + W^T O(\tilde{V}^T x)^2 + (\varepsilon + \tau_d) \dots\dots\dots (9.17)$$

Es importante hacer notar que $\varepsilon(x)$ el error de reconstrucción de la Red Neuronal, τ_d las perturbaciones del robot, y los términos de las series de expansión de Taylor de alto orden, todos tienen exactamente la misma influencia como perturbaciones en el sistema de error. El siguiente límite clave es requerido. La importancia está en permitirle a uno sobre-acotar $w(t)$ en cada paso para cada función computable conocida. Siguiendo algunas inequidades de normas usuales.

Hecho 4: El término de perturbación (9.17) está acotado de acuerdo a

$$\begin{aligned} \|w(t)\| &\leq (\varepsilon_N + b_d + c_3 Z_M) + c_6 Z_M \|\tilde{Z}\|_F + c_7 Z_M \|\tilde{z}\|_F \|r\| \\ \circ \quad \|w(t)\| &\leq c_0 + c_1 \|\tilde{Z}\|_F + c_2 \|\tilde{z}\|_F \|r\| \dots\dots\dots (9.18) \end{aligned}$$

con c_i constantes positivas conocidas computables.

Actualización de Pesos

Ahora se dan algunos algoritmos de ajuste de pesos de Redes Neuronales que garantizan la estabilidad del seguimiento del sistema de lazo cerrado bajo varias *supuestas*. Se requiere demostrar que el error de seguimiento $r(t)$ es adecuadamente pequeño y que los pesos de la Red Neuronal \hat{V}, \hat{W} permanecen acotados, por lo cual el control $\tau(t)$ está acotado. Las características clave de todos estos algoritmos son que la estabilidad está garantizada, no hay fase de aprendizaje fuera de línea así que el control de la Red Neuronal comienza inmediatamente, y que los pesos de la Red Neuronal son fáciles de inicializar sin el requerimiento de "estabilización inicial de pesos".

9.b. Reglas de ajuste de pesos

Caso Ideal - ajuste de pesos por retropropagación:

El siguiente resultado detalla el funcionamiento del lazo cerrado en un cierto caso idealizado que demanda:

- 1) No hay error en la reconstrucción funcional de la red;
- 2) No hay perturbaciones no modeladas en la dinámica del brazo robótico;
- 3) No hay términos de series de Taylor de alto orden.

Teorema 9.A

Sea que la trayectoria deseada este acotada y supóngase que el término de perturbación $w_1(t)$ en (9.14) es igual a cero.

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \hat{\sigma} + \hat{W}^T \hat{\sigma}' \tilde{V}^T x + w_1 + v \dots\dots\dots (9.14)$$

Sea que la entrada de control para (8.2) la dada por (9.12) con $v(t) = 0$

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \dots\dots\dots (8.2)$$

$$\tau = \tau_0 - v = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v \dots\dots\dots (9.12)$$

y el ajuste de pesos proporcionado por:

$$\dot{\hat{W}} = F \hat{\sigma} r^T \dots\dots\dots (9.19)$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'^T \hat{W}r)^T \dots\dots\dots (9.20)$$

y con cualesquiera matrices constantes definidas positivas F, G . Entonces el error de seguimiento $r(t)$ va a cero con t y los estimados de pesos \hat{V}, \hat{W} están acotados.

Demostración:

Se define una función de Lyapunov candidata

$$L = \frac{r^T M r}{2} + \text{tr} \left(\frac{(\tilde{W}^T \tilde{W})}{2F} \right) + \text{tr} \left(\frac{(\tilde{V}^T \tilde{V})}{2G} \right) \dots\dots\dots (9.21)$$

Diferenciándola produce:

$$\dot{L} = r^T \dot{M} r + \frac{r^T \dot{M} r}{2} + \text{tr} \left(\frac{\tilde{W}^T \dot{\tilde{W}}}{F} \right) + \text{tr} \left(\frac{\tilde{V}^T \dot{\tilde{V}}}{G} \right)$$

De aquí que haciendo sustitución de (9.14) (con $w_1 = 0, v = 0$) se produzca

$$\dot{L} = -r^T K_v r + \frac{r^T (\dot{M} - 2V_m) r}{2} + \text{tr} \left(\tilde{W}^T \left(\frac{\dot{\tilde{W}}}{F} + \hat{\sigma} r^T \right) \right) + \text{tr} \left(\tilde{V}^T \left(\frac{\dot{\tilde{V}}}{G} + x r^T \hat{W}^T \hat{\sigma}' \right) \right)$$

La propiedad anti-simétrica hace el segundo término cero, y ya que $\hat{W} = W - \tilde{W}$ con W constante,

así que $\frac{d\tilde{W}}{dt} = -\frac{d\hat{W}}{dt}$ (y así de manera similar para V), las reglas de ajuste producen

$$\dot{L} = -r^T K_v r$$

Ya que $L > 0$, y $\dot{L} \leq 0$ esto muestra estabilidad en el sentido de Lyapunov así que r, \tilde{V} y \tilde{W} (y aquí mismo \hat{V}, \hat{W}) están acotados. Además

$$\int_0^\infty -\dot{L} dt < \infty \dots\dots\dots (9.22)$$

La extensión de LaSalle se usa ahora para mostrar que $r(t)$ en efecto va a cero. La limitación de r garantiza la acotación de e y \dot{e} , de aquí que la acotación de la trayectoria deseada muestra que q, \dot{q} y x están acotación. La propiedad 2 entonces muestra la acotación de $V_m(q, \dot{q})$. Ahora $\ddot{L} = -2r^T K_v \dot{r}$ y la acotación de $M^{-1}(q)$ y de todas las señales sobre el lado derecho de (9.14) verifica la acotación de \ddot{L} , y de ahí la uniforme continuidad de \dot{L} . Esto le permite a uno utilizar el Lema de Barbalat en conexión con (9.22) para concluir que \dot{L} va a cero con t , y así como $r(t)$ tiende a cero.

Propiedad 2: $V_m(q, \dot{q})$ está limitada por $v_b(q) \|\dot{q}\|$, con $v_b(q) \in C^1(S)$

Hay que notar que el problema de la red de inicialización de pesos que esta ocurriendo en otras aproximaciones aquí no aparece. En efecto, seleccionando los pesos $\hat{W}(0), \hat{V}(0)$ en cero toma la Red Neuronal fuera del circuito y deja solamente el lazo de seguimiento exterior.

Es bien sabido que el término $K_v r$ del PD en (9.12) puede estabilizar la planta sobre una base interna. Una demostración formal revela que K_v debe ser bastante grande y el error filtrado bastante pequeño. El valor exacto de K_v , necesario para la estabilización inicial está dado en (8.4), pensando en propósitos prácticos es sólo necesario seleccionar K_v grande.

$$r = \dot{e} + \Lambda e \dots\dots\dots (8.4)$$

Hay que notar que (9.19) y (9.20) no son nada mas que la versión en tiempo continuo del algoritmo de retropropagación. En el caso de la sigmoideal escalar, por ejemplo

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) \dots\dots\dots 3.(37)$$

así que $\hat{\sigma}'^T \hat{W}r = \sigma(\hat{V}^T x) [1 - \sigma(\hat{V}^T x)] \hat{W}r \dots\dots\dots 3.(38)$

el cual es el error filtrado pesado por la estimación actual \hat{W} y multiplicado por el usual producto que involucra las salidas de la capa oculta.

El *Teorema 9.A* indica cuando la retropropagación sola debe ser suficiente, a saber, cuando la perturbación $w_1(t)$ es igual a cero. Observando el primer término en (9.17) revela que esta es una fuerte suposición mas que la simple linealización de la función del robot $f(x)$ en (8.5).

$$w(t) = \tilde{W}^T \hat{\sigma}'^T V^T x + W^T O(\tilde{V}^T x)^2 + (\varepsilon + \tau_d) \dots\dots\dots (9.17)$$

$$M\dot{r} = -V_m r - \tau + f + \tau_d \dots\dots\dots (8.5)$$

Esto es aún cuando $\varepsilon(x) = 0, \tau_d = 0$ y $f(x)$ es lineal, el ajuste por retropropagación no está garantizado el lograr el exitoso seguimiento de la trayectoria deseada. Nótese que $f(x)$ es lineal sólo en el caso del brazo robótico de una articulación. La suposición $w_1(t) = 0$, significa, mas aún, que la Red Neuronal puede aproximar exactamente la función sobre todo el espacio R^n . Esta es una fuerte suposición. La falla de la retropropagación simple en el caso general. En efecto en el caso de la Red Neuronal de dos capas $V = I$ (es decir, lineal en los parámetros), es fácil mostrar que, usando la regla de actualización (9.19), los pesos de \tilde{W} no son generalmente acotados a menos que la salida $\sigma(x)$ de la capa oculta obedezca a un estricta condición de PE (excitación persistente).

En el caso de tres capas (no-lineal), las condiciones PE no son fáciles de derivar como lo que se muestra con las propiedades de observabilidad de un cierto sistema bilineal. Así que, la retropropagación usada en una red que no puede reconstruir exactamente $f(x)$, o sobre un brazo robótico con acotadas perturbaciones no-modeladas, o cuando $f(x)$ es no-lineal, no puede garantizar producir pesos acotados en los sistemas de lazo cerrado.

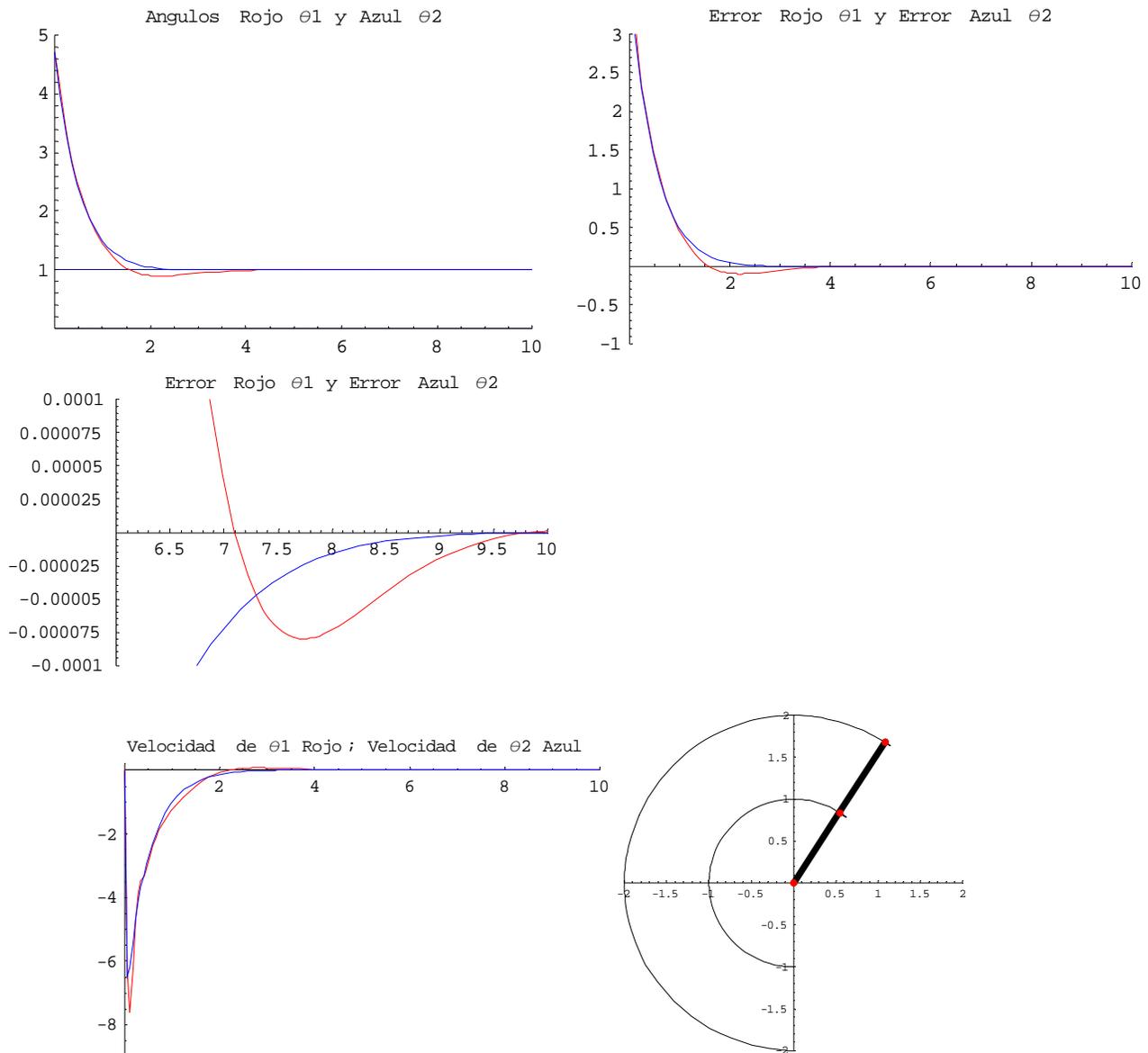
9.c Simulaciones y Observaciones

Simulación 9.1

Brazo con dos eslabones, **Sin Fricción**, posición deseada: 1 radian a partir de la horizontal

Programa de Mathematica **PD+RN_SFricción_posición_2**

Diagonal K = 8, Diagonal $\Lambda = 2$ Número de Neuronas: 6, Diagonal T = 10, Diagonal S = 10



Observaciones:

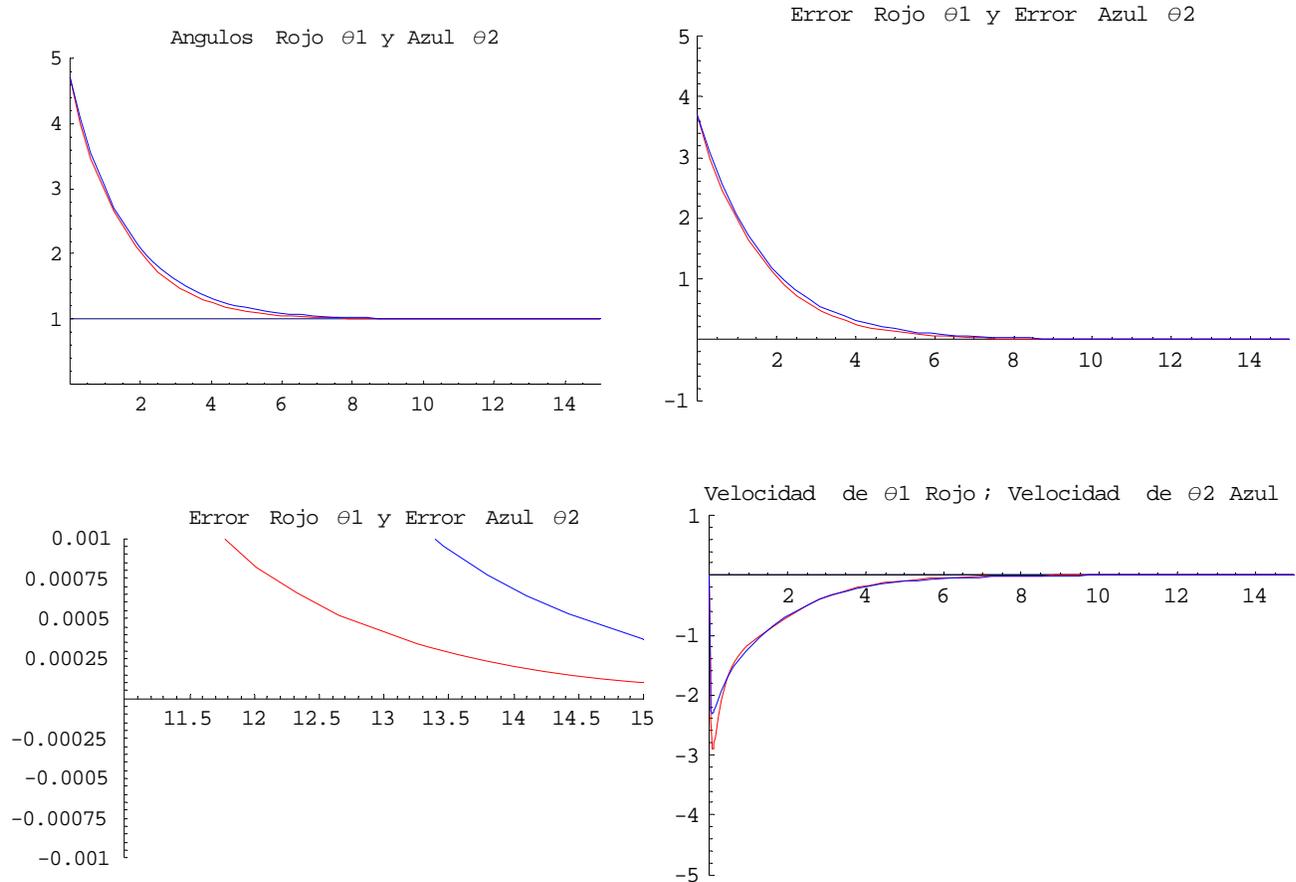
Este tipo de Controlador en este caso logra una excelente aproximación de la posición con un error que se reduce a micras de radian en el tiempo asignado. Esto se logra con una Red Neuronal de 6 neuronas y un controlador PD con valores pequeños de las ganancias. La grafica de velocidades no presenta anomalías. El sistema tiende a una estabilidad asintótica presentándose suaves oscilaciones decrecientes.

Simulación 9.2

Brazo con dos eslabones, **Con fricción** posición: a 1 radian de la horizontal

Programa de Mathematica **PD+RN_CFricción_posición_2**

Diagonal K = 18, Diagonal $\Lambda = 0.6$ Número de Neuronas: 6, Diagonal T = 10, Diagonal S = 10



Observaciones:

Con este tipo de Controlador se logra una buena aproximación a la posición deseada con un error que se reduce a diezmilésimas de radian. La perturbación por la fricción esta rechazada. Las velocidades no presentan alteraciones drásticas.

Los valores de las ganancias que se requieren son bajos y se utilizan matrices diagonales simétricas con valores iguales-

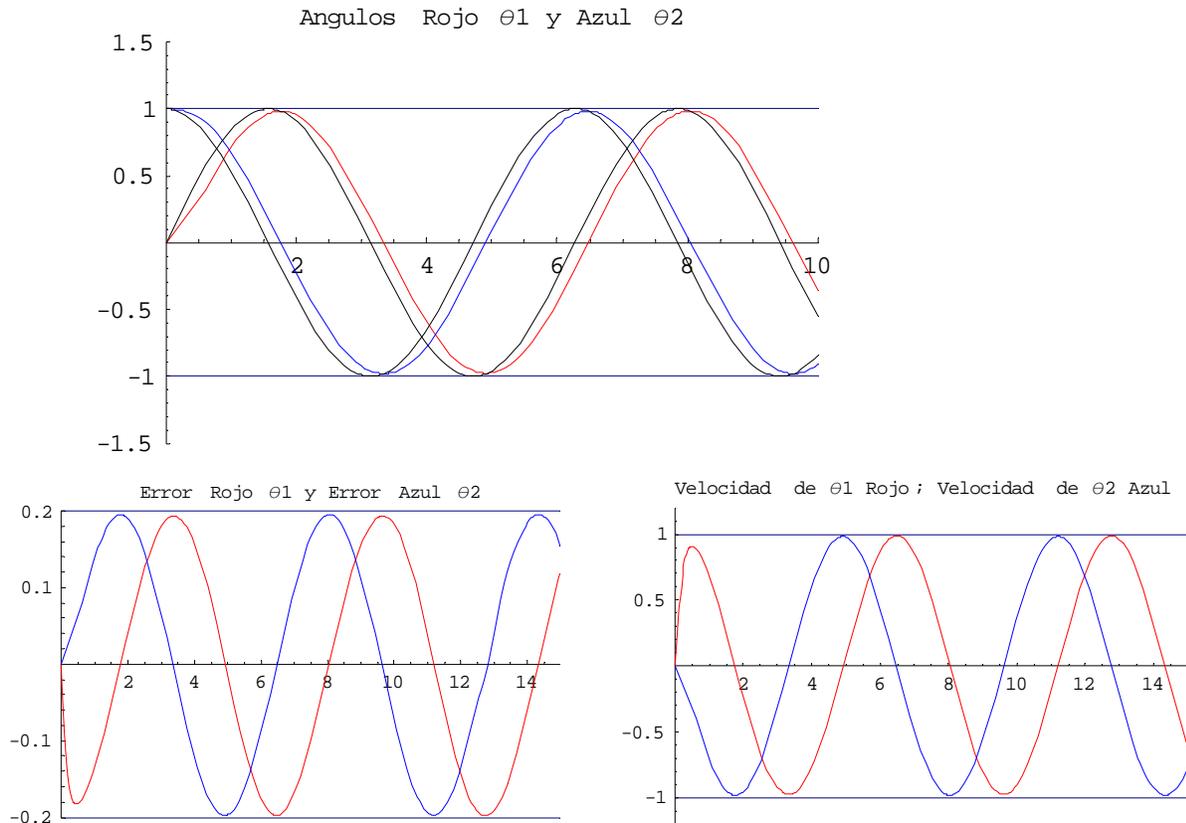
No se puede asegurar que se comporte asintóticamente estable por un largo tiempo mas allá del estipulado en el programa, debido a la acción permanente de la fricción y la retroalimentación.

Simulación 9.3

Brazo con dos eslabones, **Sin Fricción**, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

Programa de Mathematica **PD+RN_SFricción_Aprox.1**

Diagonal K = 30, Diagonal $\Lambda = 5$ Número de Neuronas: 10, Diagonal T = 30, Diagonal S = 30



Observaciones:

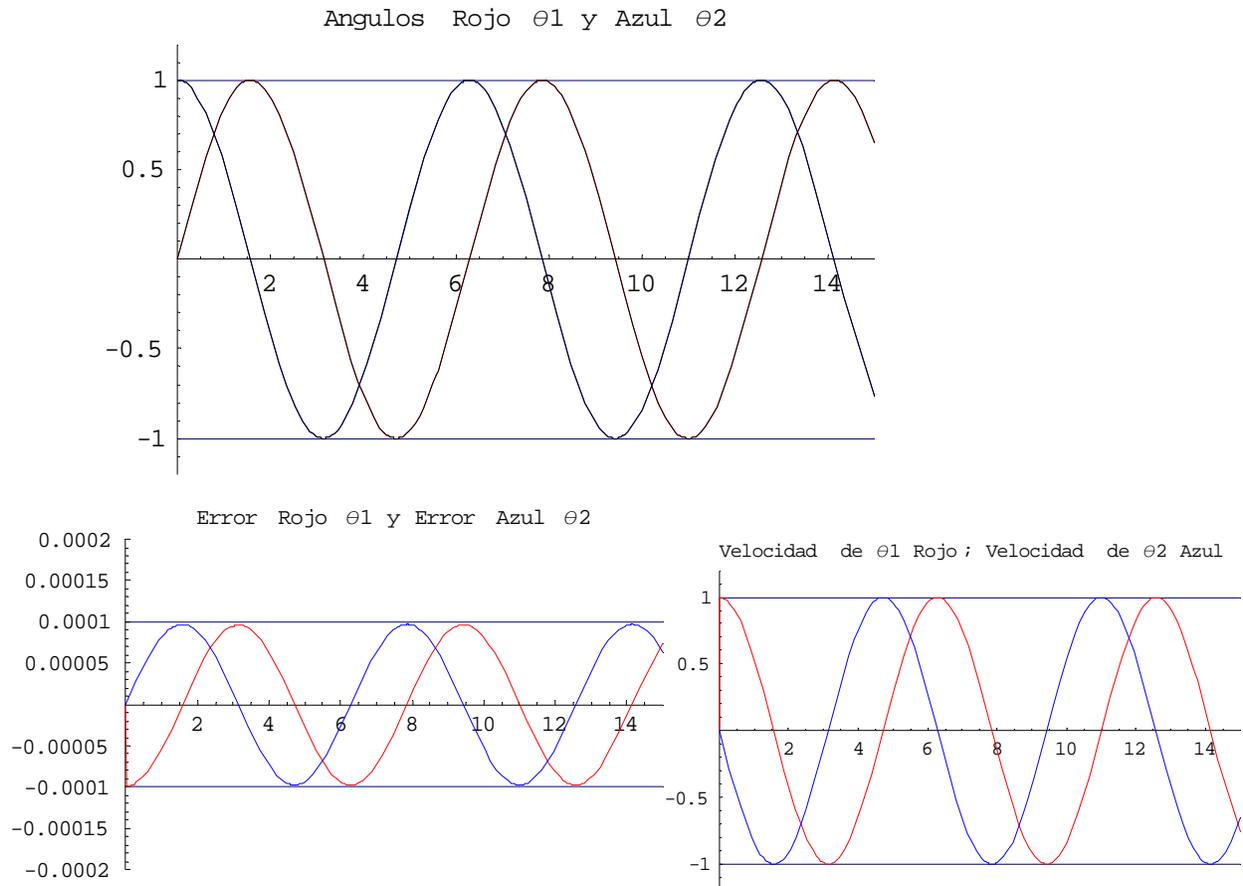
Aquí se presenta el resultado de una primera aproximación a la trayectoria de referencia con un valor del error considerado alto (a décimas de radián). El seguimiento inicia en un punto no sólo cercano, sino que perteneciente a la función de referencia, esto para que se apliquen teoremas referentes a la convergencia del error de seguimiento y estabilidad del sistema. Las gráficas del error y de las velocidades no presentan alteraciones.

Simulación 9.4

Brazo con dos eslabones, sin fricción, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

Programa de Mathematica **PD+RN_SFricción_traySC_Aprox.Final**

Diagonal K = 380, Diagonal $\Lambda = 10300$ Número Neuronas: 10, Diagonal T = 30, Diagonal S = 30



Observaciones:

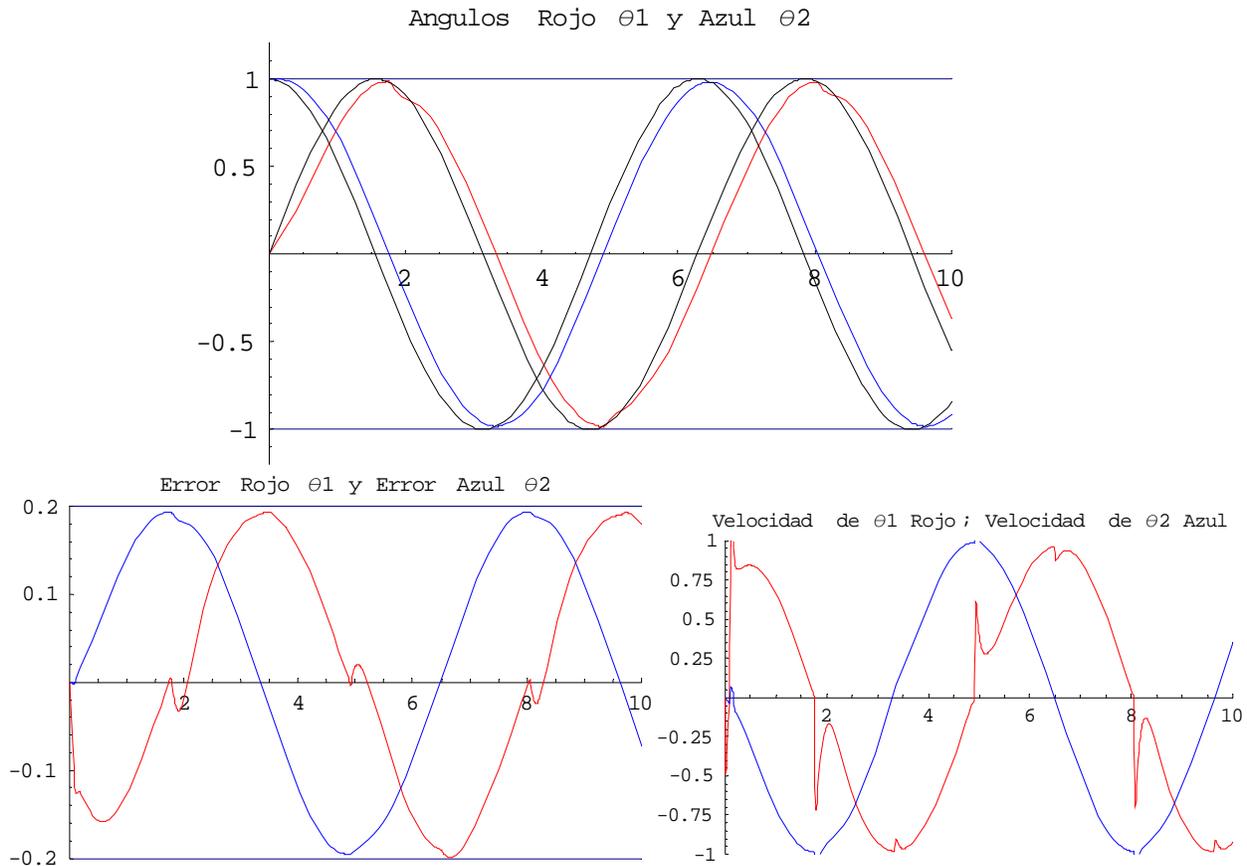
En esta segunda aproximación al seguimiento de trayectoria de referencia. Se alcanza un tolerancia del error de una diezmilésima de radian. El sistema llega un estado Estable Acotado Uniforme y Último *UUB*. En las gráficas del error y la velocidad se conservan las características de las funciones de referencia lo cual indica que aún puede seguir reduciéndose el tamaño arbitrario del error incrementando únicamente la ganancia Λ .

Simulación 9.5

Brazo con dos eslabones, Con fricción, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

Programa de Mathematica **PD+RN_CFricción_Aprox.1**

Diagonal K = 30, Diagonal $\Lambda = 5$ Número de Neuronas: 10, Diagonal T = 30, Diagonal S = 30



Observaciones:

Se presenta una aproximación al seguimiento de la trayectoria de referencia que permite observar que el sistema con Controlador PD y Red Neuronal que se actualiza por retropropagación solamente, no modela la perturbación ocasionada por la fricción ya que se presentan no linealidades pronunciadas en los cambios de dirección de las pendientes, localizadas en la parte media de las sinusoides principalmente.

Aunque el Controlador no modela bien los puntos donde se acentúa la fricción; si modela la trayectoria que debe seguir, esto se observa en la gráfica de los errores donde estos tienen la forma de las funciones que deben seguir. Las no-linealidades de la fricción no son modeladas, sólo son rechazadas.

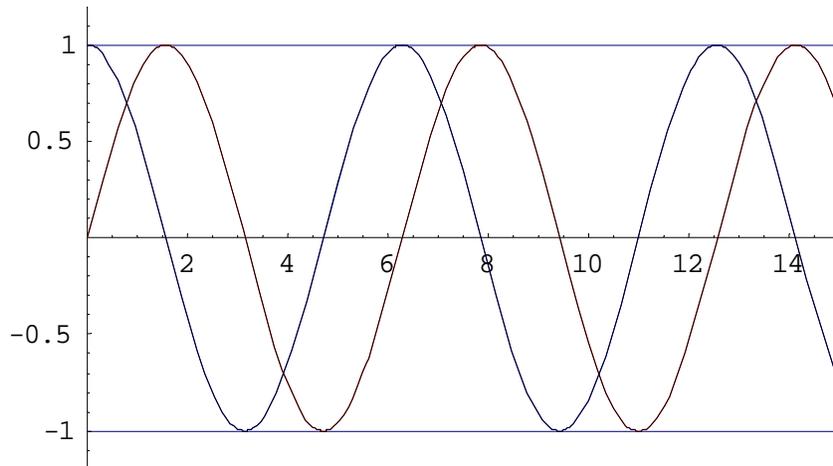
Simulación 9.6

Brazo con dos eslabones, Con fricción, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

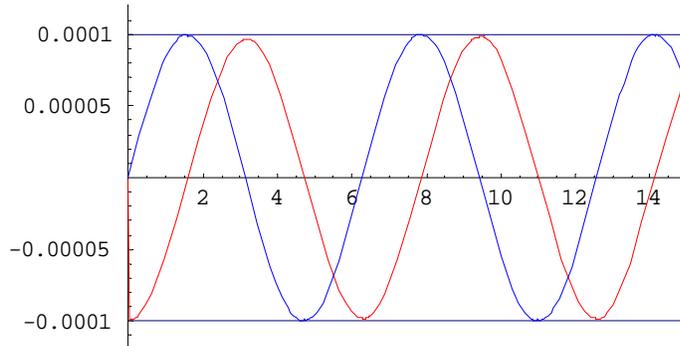
Programa de Mathematica **PD+RN_CFricción_traySC_Aprox.Final**

Diagonal K = 380, Diagonal $\Lambda = 10000$ Número Neuronas: 10, Diagonal T = 30, Diagonal S = 30

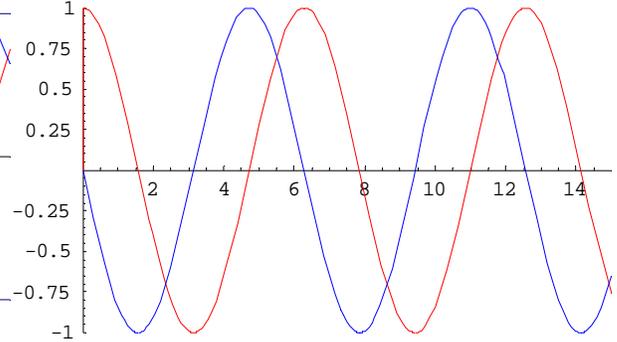
Angulos Rojo θ_1 y Azul θ_2



Error Rojo θ_1 y Error Azul θ_2



Velocidad de θ_1 Rojo; Velocidad de θ_2 Azul



Observaciones:

En esta segunda aproximación al seguimiento de trayectoria de referencia. Se alcanza un tolerancia del error de una diezmilésima de radian. El sistema llega un estado Estable Acotado Uniforme y Último *UUB*.

En las gráficas del error y la velocidad se conservan las características de las funciones de referencia lo cual indica que aún puede seguir reduciéndose el tamaño arbitrario del error incrementando únicamente la ganancia Λ .

Las no-linealidades causadas por la perturbación de la fricción a este nivel de precisión no aparecen; son absorbidas por el sistema adaptativo y la retropropagación, sin embargo como se indica en el inciso 9 esto no siempre se puede garantizar.

10. CONTROLADOR PD Y RED NEURONAL AUMENTADA ROBUSTA

10.a. Ampliación de la red neuronal

10.a.1. Redes Neuronales para aproximación de funciones con discontinuidades

Como se vió antes la aparición de la no linealidad de la fricción en actuadores es discontinua en el origen. Una de las más importantes propiedades de las Redes Neuronales RNs para propósitos de control es la propiedad de aproximación universal. Desafortunadamente, esta propiedad es generalmente probada como en el inciso anterior para funciones continuas. Sin embargo en la mayoría de los sistemas de control industrial reales hay funciones que no son suaves (es decir, continuas por partes) para las cuales los resultados de aproximación en la literatura son escasos.

Se ha encontrado en los intentos por aproximar tramos de funciones continuas usando funciones de activación suaves que se requieren muchos nodos en la capa oculta de la Red Neuronal y muchas iteraciones de entrenamiento y aún así no producen muy buenos resultados.

El resultado principal en esta sección es un Teorema para funciones continuas por partes o funciones con saltos. Se ha encontrado que para aproximar tales funciones adecuadamente, es necesario aumentar la Red estandar que usa funciones de activación suaves con *nodos extras* conteniendo un cierto *conjunto básico de aproximación funciones de salto* de funciones de activación no suaves. Esta novedosa estructura de Red Neuronal Aumentada permite la aproximación de tramos de funciones continuas del tipo que aparecen en las no-linealidades del actuador que controla el movimiento.

La red neuronal aumentada consiste de neuronas teniendo funciones de activación sigmoidales estandar más algunas *neuronas adicionales* teniendo una clase especial de funciones de activación no suaves llamada "*funciones básicas de aproximación de funciones con salto*". Esta Red Neuronal Aumentada con neuronas adicionales puede aproximar cualquier tramo de función continua con discontinuidades en un número finito de puntos conocidos.

En adición a funciones sigmoidales continuas uno requiere un conjunto de *funciones básicas discontinuas*. Se tienen dos conjuntos adecuados, *las funciones de aproximación de salto polinomiales* y *las funciones de aproximación de salto sigmoideas*.

Teorema 10.1 *Aproximación de tramos de funciones continuas*

Sea dada cualquier función acotada $f : S \rightarrow \mathfrak{R}$ esto es continua y analítica sobre un conjunto compacto $S \subset \mathfrak{R}$ excepto en $x = c$, donde la función f tiene un salto finito y es continua por la derecha. Entonces dado cualquier $\varepsilon > 0$ existe una suma F de la forma

$$F(x) = g(x) + \sum_{k=0}^N a_k f_k(x-c) \dots\dots\dots (10.1)$$

tal que $|F(x) - f(x)| < \varepsilon \dots\dots\dots (10.2)$

para cada x en S donde $g(\bullet)$ es una función en $C^\infty(S)$ y las funciones básicas de aproximación de salto polinomiales f_k están definidas como:

$$f_k(x) = \begin{cases} 0 \dots\dots \text{para } x < 0 \\ x^k \\ \frac{x^k}{k!} \dots\dots \text{para } x \geq 0 \end{cases} \dots\dots\dots (10.3)$$

Las funciones básicas de aproximación de salto polinomiales f_k tienen un defecto al usarse en control por retroalimentación y es que no están acotadas. Esto puede producir inestabilidades no acotadas en un conjunto de funciones básicas que están acotadas. Se desprende el resultado principal que nos permite reemplazar el conjunto básico de aproximación de salto polinomial por sigmoideas discontinuas.

Lema 10.A *Funciones de aproximación de salto sigmoideas*

Cualquier combinación lineal de funciones de aproximación de salto polinomiales f_k , $\sum_{k=0}^N a_k f_k(x)$ puede ser representada como:

$$\sum_{k=0}^N a_k f_k(x) = z(x) + \sum_{k=0}^N b_k \varphi_k(x) \dots\dots\dots (10.4)$$

Donde las funciones de aproximación de salto sigmoideas φ_k están definidas como:

$$\varphi_k(x) = \begin{cases} 0 \dots\dots \text{para } x < 0 \\ (1 - e^{-x})^k \dots\dots \text{para } x \geq 0 \end{cases} \dots\dots\dots (10.5)$$

Y donde $z(x)$ es una función que pertenece a $C^N(S)$.

Demostración.

Es suficientes con probar que existen coeficientes b_k tales que las primeras N derivadas de la

expresión $\sum_{k=0}^N a_k f_k(x) - \sum_{k=0}^N b_k \varphi_k(x) \dots\dots\dots (10.6)$

son continuas.

Para $x < 0$, esta expresión tiene un valor constante cero. Por lo tanto, uno debe mostrar que existen coeficientes b_i tales que para $x > 0$, las primeras N derivadas de la expresión son iguales a cero.

Para $x > 0$, uno tiene

$$\sum_{k=0}^N a_k f_k(x) - \sum_{k=0}^N b_k \varphi_k(x) = \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^N b_k (1 - e^{-x})^k$$

Usando la fórmula binomial uno puede expandir esta expresión en:

$$\begin{aligned} & \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^N b_k \left[\sum_{i=0}^k \binom{k}{i} (-e^{-x})^i \right] \\ &= \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^N b_k \left[\sum_{i=0}^k \binom{k}{i} (-1)^i (e^{-ix}) \right] \\ &= \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^N b_k \left[\sum_{i=0}^k \binom{k}{i} (-1)^i \sum_{j=0}^{\infty} \frac{(-ix)^j}{j!} \right] \\ &= \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^N \sum_{i=0}^k b_k \binom{k}{i} (-1)^i \sum_{j=0}^{\infty} (-1)^j \frac{x^j}{j!} \\ &= \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^N \sum_{i=0}^k \sum_{j=0}^{\infty} b_k \binom{k}{i} (-1)^i (-1)^j \frac{x^j}{j!} \\ &= \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{j=0}^{\infty} \sum_{k=0}^N \sum_{i=0}^k b_k \binom{k}{i} (-1)^i (-1)^j \frac{x^j}{j!} \\ &= \sum_{k=0}^N a_k \frac{x^k}{k!} - \sum_{k=0}^{\infty} \sum_{m=0}^N \sum_{i=0}^m b_m \binom{m}{i} (-1)^i (-1)^k \frac{x^k}{k!} \end{aligned}$$

Con el fin de que las primeras N derivadas de la expresión (10.6) sean cero, los coeficientes de x^k para $k = 0, 1, \dots, N$ deben ser cero en esta ecuación. Por lo tanto, se requiere que:

$$\begin{aligned} a_k - \sum_{m=0}^N \sum_{i=0}^m b_m \binom{m}{i} (-1)^i (-i)^k &= 0 \\ a_k - \sum_{m=0}^N b_m \sum_{i=0}^m \binom{m}{i} (-1)^i (-i)^k &= 0 \end{aligned}$$

Usando el Lema técnico el cual dice que

$$\sum_{i=0}^m \binom{m}{i} (-1)^i (-i)^k = 0$$

Para cada $m > k$, uno obtiene

$$a_k - \sum_{m=0}^k b_m \sum_{i=0}^m \binom{m}{i} (-1)^i (-i)^k = 0$$

Para $k = 1, 2, \dots, N$ o

$$a_k - b_k \sum_{i=0}^k \binom{k}{i} (-1)^i (-i)^k - \sum_{m=0}^{k-1} b_m \sum_{i=0}^m \binom{m}{i} (-1)^i (-i)^k = 0$$

Para $k = 1, 2, \dots, N$. Por lo tanto, uno obtiene la relación recurrente para los coeficientes

$$b_k = \frac{a_k - \sum_{m=0}^{k-1} b_m \sum_{i=0}^m \binom{m}{i} (-1)^i (-i)^k}{\sum_{i=0}^k \binom{k}{i} (-1)^i (-i)^k}$$

Para $k = 1, 2, \dots, N$ y con $b_0 = a_0$. Q.E.D.

Las funciones de aproximación de salto sigmoideas φ_k para $k = 0, 1, 2, 3$ tienen una mayor ventaja sobre los conjuntos de base polinomial en el diseño del control por retroalimentación ya que las primeras están acotadas.

El siguiente resultado importante se desprende directamente. El uso de conjuntos base de aproximación de salto sigmoideas para funciones de aproximación de salto en las Redes Neuronales.

Teorema 10.2 *Resultados de aproximación con red neuronal general para funciones con saltos*

Sea que hay una función acotada dada $f : S \rightarrow \mathfrak{R}$, la cual es continua y analítica sobre un conjunto compacto S excepto en $x = c$, donde la función f tiene un salto finito y es continua por la derecha. Dado cualquier $\varepsilon > 0$, existe una suma F de la forma

$$F(x) = \sum_{k=0}^L a_k \sigma(m_k x + n_k) + \sum_{k=0}^N a_k \varphi_k(x - c) \dots\dots\dots (10.7)$$

tal que $|F(x) - f(x)| < \varepsilon \dots\dots\dots (10.8)$

para cada x en S donde $\sigma(x)$ es una función sigmoidea estandar y $\varphi_k(\bullet)$ son las funciones de aproximación de salto sigmoideas.

Este Teorema muestra que cualquier función con saltos puede ser aproximada por una Red Neuronal estandar teniendo sigmoideas continuas estandar (el primer término de (10.7)) mas funciones de salto sigmoideas discontinuas (el segundo término de (10.7)).

Se puede mostrar, que en vez de utilizar las funciones de aproximación de salto sigmoideas, ecuaciones (10.5) se pueden usar Funciones de Aproximación de Salto Sigma

$$\varphi_k(x) = \begin{cases} 0 \dots\dots\dots \text{para } x < 0 \\ \left(\frac{1 - e^{-x}}{1 + e^{-x}} \right)^k \dots\dots \text{para } x \geq 0 \end{cases} \dots\dots\dots (10.9)$$

○ funciones básicas de salto basadas en Tangente Hiperbólica

$$\varphi_k(x) = \begin{cases} 0 & \text{para } x < 0 \\ \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^k & \text{para } x \geq 0 \end{cases} \dots\dots\dots (10.10)$$

10.a.2. Red neuronal aumentada en el controlador

Aquí se presenta la estructura y los algoritmos de entrenamiento derivados para la Red Neuronal Multicapa Aumentada. Esta RN aumentada es capaz de aproximar funciones con saltos, considerando que los puntos de discontinuidad son conocidos. Ya que los puntos de discontinuidad son conocidos en muchas características no lineales de sistemas de movimiento industrial (es decir, el modelo de fricción es discontinuo en cero, la inversa zona muerta es discontinua en cero, etc.) esta RN aumentada es una herramienta útil para la compensación de los efectos parásitos y no-linealidades de actuadores sistemas de control industrial.

La capa aumentada de RN es mostrada en la figura 10.1. Uno puede ver que la Red Neuronal Aumentada es una combinación de una Red Neuronal estandar (con funciones de activación estandar $\sigma(\cdot)$) mas algunas *neuronas extra* con funciones de activación sigma discontinuas φ_k . Comparando la Red Neuronal Aumentada con la Red Neuronal Estandar uno puede ver que la primera tiene dos conjuntos de funciones de activación en la capa oculta, σ_k y φ_k , dos conjuntos de pesos para la primera capa, V_1^T y V_2^T y dos conjuntos de pesos de la segunda capa W_1^T y W_2^T . Con esta estructura la Red Neuronal tiene:

$$y = \sum_{\ell=1}^L w_{1\ell}^1 \sigma(v_{\ell 1}^1 x_1 + v_{\ell 0}^1) + \sum_{\ell=1}^N w_{1\ell}^2 \varphi_\ell(v_{\ell 1}^2 x_1 + v_{\ell 0}^2) + w_{10} \dots\dots\dots (10.11)$$

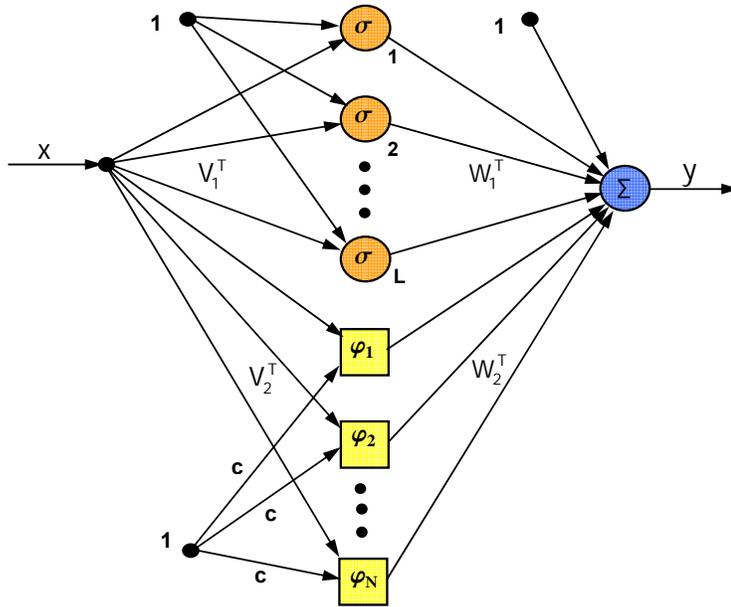
De manera similar como en el caso de Red Neuronal estandar, la salida de la Red Neuronal Aumentada está dada por una matriz de la forma:

$$y = W_1^T \sigma(V_1^T x) + W_2^T \varphi(V_2^T x) \dots\dots\dots (10.12)$$

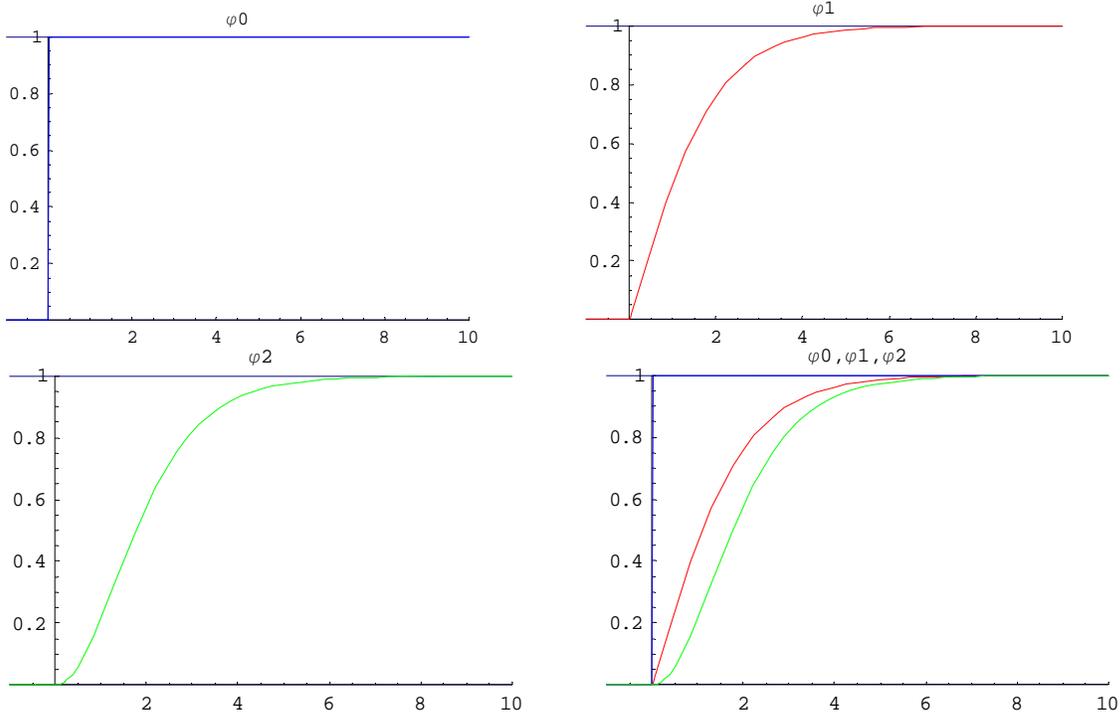
Si las funciones de activación de la capa oculta σ está definida como las sigmoideas, uno toma las funciones básicas de salto φ_i como en (10.5), Entonces se usa el Teorema 10.2 que dice que esta Red Neuronal aumentada puede aproximar cualquier función con un salto finito, proporcionando la localización del salto.

Los algoritmos de ajuste estandar de la Red Neuronal deben ser modificados ya que la estructura de la Red ha sido cambiada. El conjunto de los pesos de la segunda capa W_1^T y W_2^T son entrenados en la forma usual de las Redes Neuronales Multicapa, ambas son tratadas de la misma manera incluyendo los pesos de los sesgos. Los pesos de la primera capa V_1^T son actualizados en la forma usual de las Redes (es decir, aproximación por retropropagación). Sin embargo los pesos de

V_2^T y el sesgo, están fijos y seleccionados para que correspondan a los puntos de discontinuidad del salto. Efectivamente la porción de aproximación del salto de la Red Neuronal Aumentada es una Red de una capa, ya que tiene solamente una capa simple de pesos ajustables.



Red Neuronal Multicapa Aumentada
Figura 10.1



Gráficas funciones de activación llamadas Funciones de Aproximación de Salto Sigma
Figura 10.2

10.b. Control adaptativo robusto en la red neuronal

En la planta libre-de perturbaciones, el objetivo era diseñar un controlador adaptativo estable que asegurara un seguimiento-del modelo asintótico. En la presencia de perturbaciones, el seguimiento-del modelo exacto ya no es posible, y el objetivo es diseñar un controlador que asegure márgenes en todas las señales del sistema y satisfaga especificaciones de ejecución.

Nuevas aproximaciones son necesarias para asegurar la limitación de todas las señales en el sistema y dando lugar a un cuerpo de trabajo llamado *Teoría de Control Adaptativo Robusto*. Robustez en este sentido implica que el sistema adaptativo se desempeñe esencialmente de la misma manera, aún cuando perturbaciones exógenas o endógenas estén presentes.

El Caso General: Para enfrentar la estabilidad y la ejecución del seguimiento de un controlador de brazo robótico de Red Neuronal en el difícil caso general, requerimos:

1. La modificación de las reglas de ajuste de los pesos y
2. La inclusión de un término que da robustez $v(t)$.

El problema en este caso es que, pensando que no es difícil concluir que $r(t)$ esta acotado, es imposible sin estas modificaciones mostrar que los pesos de las Redes Neuronales están acotados en general. La limitación de los pesos es necesaria para verificar que el control de entradas $\tau(t)$ permanece acotado.

El siguiente Teorema se considera como una extensión de la teoría de Lyapunov. La perturbación τ_d , el error de reconstrucción ε de la Red Neuronal, y la no-linealidad de $f(x)$ hace imposible mostrar que la derivada de Lyapunov \dot{L} es no-positiva para toda $r(t)$ y valores de pesos.

En efecto, sólo posible mostrar que \dot{L} es negativa en el exterior de un conjunto compacto en el espacio de estado. Esto, sin embargo, permite concluir la limitación del seguimiento del error y de los pesos de la Red Neuronal. De hecho, los límites explícitos son descubiertos durante la demostración. La requerida extensión de Lyapunov es la última parte de la demostración.

Teorema 10.3 Actualización de Pesos con términos de robustez

Sea que la trayectoria deseada este acotada por (9.4).

$$\begin{Bmatrix} q_d \\ \dot{q}_d \\ \ddot{q}_d \end{Bmatrix} \leq Q_d \dots\dots\dots (9.4)$$

Tómese el control de entradas de (8.2) en (9.12)

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau \dots\dots\dots (8.2)$$

$$\tau = \tau_0 - v = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v \dots\dots\dots (9.12)$$

con un término de robustez:

$$v(t) = -K_z \left(\|\hat{Z}\|_F + Z_M \right) r \dots\dots\dots (10.13)$$

y la ganancia $K_z > C_2 \dots\dots\dots (10.14)$

con C_2 la constante conocida en (9.18)

$$\|w(t)\| \leq c_0 + c_1 \|\tilde{Z}\|_F + c_2 \|\tilde{Z}\|_F \|r\| \dots\dots\dots (9.18)$$

Sea el ajuste de pesos de la Red Neuronal el proporcionado por:

$$\dot{\hat{W}} = F \hat{\sigma} r^T - F \hat{\sigma} \hat{V}^T x r^T - kF \|r\| \hat{W} \dots\dots\dots (10.15)$$

$$\dot{\hat{V}} = Gx (\hat{\sigma}^T \hat{W} r)^T - kG \|r\| \hat{V} \dots\dots\dots (10.16)$$

con cualesquiera matrices constantes $F = F^T > 0$, $G = G^T > 0$ y un parámetro de diseño escalar $k > 0$. Entonces para un control de ganancia K_v bastante grande, el error de seguimiento filtrado $r(t)$ y los estimados de pesos \hat{V}, \hat{W} de la Red Neuronal están *UUB*, con los límites prácticos dados específicamente por los lados derechos de (10.17) y (10.18). Más aún, el error de seguimiento puede ser conservado tan pequeño como se desee por el incremento de la ganancia K_v en (9.12).

$$\tau = \tau_0 - v = \hat{W}^T \sigma(\hat{V}^T x) + K_v r - v \dots\dots\dots (9.12)$$

Demostración: Sea la propiedad de aproximación (6.d.4) mantenida con una precisión dada ε_N para toda x en el conjunto compacto $U_x \equiv \{x \| \|x\| \leq b_x\}$ con $b_x > c_1 Q_d$ en (9.5).

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x) \dots\dots\dots (6.d.4)$$

$$\|x\| \leq c_1 Q_d + c_2 \|r\| \dots\dots\dots (9.5)$$

Se define $U_r = \left\{ r \| \|r\| \leq \frac{b_x - c_1 Q_d}{c_2} \right\}$. Sea $r(0) \in U_r$. Entonces la propiedad de aproximación se mantiene.

Seleccionando ahora la función de Lyapunov (9.21), diferenciando y sustituyendo ahora para el sistema de error (9.13) se produce

$$L = \frac{r^T M r}{2} + tr \left(\frac{(\tilde{W}^T \tilde{W})}{2F} \right) + tr \left(\frac{(\tilde{V}^T \tilde{V})}{2G} \right) \dots\dots\dots (9.21)$$

$$M\dot{r} = -(K_v + V_m) r + \tilde{W}^T \hat{\sigma} + \tilde{W}^T \tilde{\sigma} + \tilde{W}^T \tilde{\sigma} + (\varepsilon + \tau_d) + v \dots\dots\dots (9.13)$$

$$\dot{L} = -r^T K_v r + \frac{r^T (\dot{M} - 2V_m) r}{2} + tr \tilde{W}^T \left(\left(\frac{\dot{\tilde{W}}}{F} \right) + \hat{\sigma} r^T - \hat{\sigma}' \hat{V}^T x r^T \right) + tr \tilde{V}^T \left(\left(\frac{\dot{\tilde{V}}}{G} \right) + x r^T \hat{W}^T \hat{\sigma}' \right) + r^T (w + v)$$

Las reglas de ajuste dan:

$$\begin{aligned} \dot{L} = & -r^T K_v r + k \|r\| tr \tilde{W}^T (W - \tilde{W}) + k \|r\| tr \tilde{V}^T (V - \tilde{V}) + r^T (w + v) = \\ & -r^T K_v r + k \|r\| tr \tilde{Z}^T (Z - \tilde{Z}) + r^T (w + v) \end{aligned}$$

ya que $tr \tilde{Z}^T (Z - \tilde{Z}) = \langle \tilde{Z}, Z \rangle_F - \|\tilde{Z}\|_F^2 \leq \|\tilde{Z}\|_F \|Z\|_F - \|\tilde{Z}\|_F^2$ aquí resulta

$$\begin{aligned} \dot{L} \leq & -K_{v\min} \|r\|^2 + k \|r\| \|\tilde{Z}\|_F (Z_M - \|\tilde{Z}\|_F) - K_Z (\|\hat{Z}\|_F + Z_M) \|r\|^2 + \|r\| \|w\| \\ \leq & -K_{v\min} \|r\|^2 + k \|r\| \|\tilde{z}\|_F (Z_M - \|\tilde{Z}\|_F) - K_Z (\|\hat{Z}\|_F + Z_M) \|r\|^2 \\ & + \|r\| [C_0 + C_1 \|\tilde{Z}\|_F + C_2 \|\tilde{Z}\|_F \|r\|] \\ \leq & -\|r\| [K_{v\min} \|r\| + k \|\tilde{Z}\|_F (\|\tilde{Z}\|_F - Z_M) - C_0 - C_1 \|\tilde{Z}\|_F] \end{aligned}$$

donde $K_{v\min}$ es el mínimo valor singular de K_v y la última desigualdad se mantiene debido a

$$K_z > C_2 \dots \dots \dots (10.14)$$

Así, \dot{L} es negativa tan grande como el término entre paréntesis es positivo.

Definiendo $C_3 = Z_M + \frac{C_1}{k}$ y completando el cuadrado resulta:

$$\begin{aligned} K_{v\min} \|r\| + k \|\tilde{Z}\|_F (\|\tilde{Z}\|_F - C_2) - C_0 = \\ k \left(\|\tilde{Z}\|_F - \frac{C_3}{2} \right)^2 - \frac{kC_3^2}{4} + K_{v\min} \|r\| - C_0 \end{aligned}$$

la cual es garantizada positiva tan grande como

$$\|r\| > \frac{\frac{kC_3^2}{4} + C_0}{K_{v\min}} \equiv b_r \dots \dots \dots (10.17)$$

$$\circ \quad \|\tilde{Z}\|_F > \frac{C_3}{2} + \sqrt{\frac{C_3^2}{4} + \frac{C_0}{k}} \equiv b_z \dots \dots \dots (10.18)$$

$$\text{donde} \quad C_3 = Z_M + \frac{C_1}{k} \dots \dots \dots (10.19)$$

Así que, \dot{L} es un conjunto compacto exterior negativo. La forma del lado derecho de (10.17) muestra que la ganancia de control K_v puede ser seleccionada lo bastante grande para que

$b_r < \frac{(b_z - C_1 Q_d)}{C_2}$. Entonces cualquier trayectoria $r(t)$ empezando en U_r y evolucionando completamente dentro de U_r . De acuerdo con una extensión del Teorema de Lyapunov estandar se demuestra que la *UUB* de ambas $\|r\|$ y $\|\tilde{Z}\|_F$.

Hay que hacer algunas observaciones con respecto a esto:

Primera. Ya que cualquier curso de $\|r\|$ o de $\|\tilde{Z}\|_F$ mas allá de los límites dados en (10.17) y (10.18) respectivamente, se dirigen a una disminución en la función L de Lyapunov, se desprende que el lado derecho de (10.17) y (10.18) pueden ser tomados como límites prácticos de $\|r\|$ y $\|\tilde{Z}\|_F$ respectivamente, en el sentido de que las normas serán restringidas a estos valores mas ε para cualquier arbitrariamente pequeño $\varepsilon > 0$.

Una comparación con los resultados para *control adaptativo* muestra que en ε_N el error de reconstrucción para Redes Neuronales, las perturbaciones limitadas b_d y los términos de alto orden de la serie de Taylor, todos se encuentran involucrados en las constantes C_0 y C_3 incrementando los límites sobre $\|r\|$ y $\|\tilde{Z}\|_F$ de una manera muy interesante. Nótese de (10.17), sin embargo, que los límites de un error de seguimiento arbitrariamente pequeño pueden ser alcanzados por la selección de una ganancia de control grande K_v . (Si K_v es tomado como una matriz diagonal, $K_{v\min}$ es simplemente el elemento ganancia mas pequeño). Por otra parte (10.18) revela que los errores de pesos de la Red Neuronal están fundamentalmente limitados por Z_M (continuo hasta C_3). El parámetro k ofrece un intercambio en el diseño relativo entre las posibles magnitudes de $\|r\|$ y $\|\tilde{Z}\|_F$.

Los primeros términos de (10.15) y (10.16) no son nada mas que el algoritmo de retropropagación estandar. Los últimos términos corresponden a la modificación-*e* con el uso estandar de control adaptivo para garantizar estimaciones de parámetros acotadas; ellos forman una clase especial de términos en la actualización de pesos.

El segundo término en (10.15) es nuevo y merece discusión.

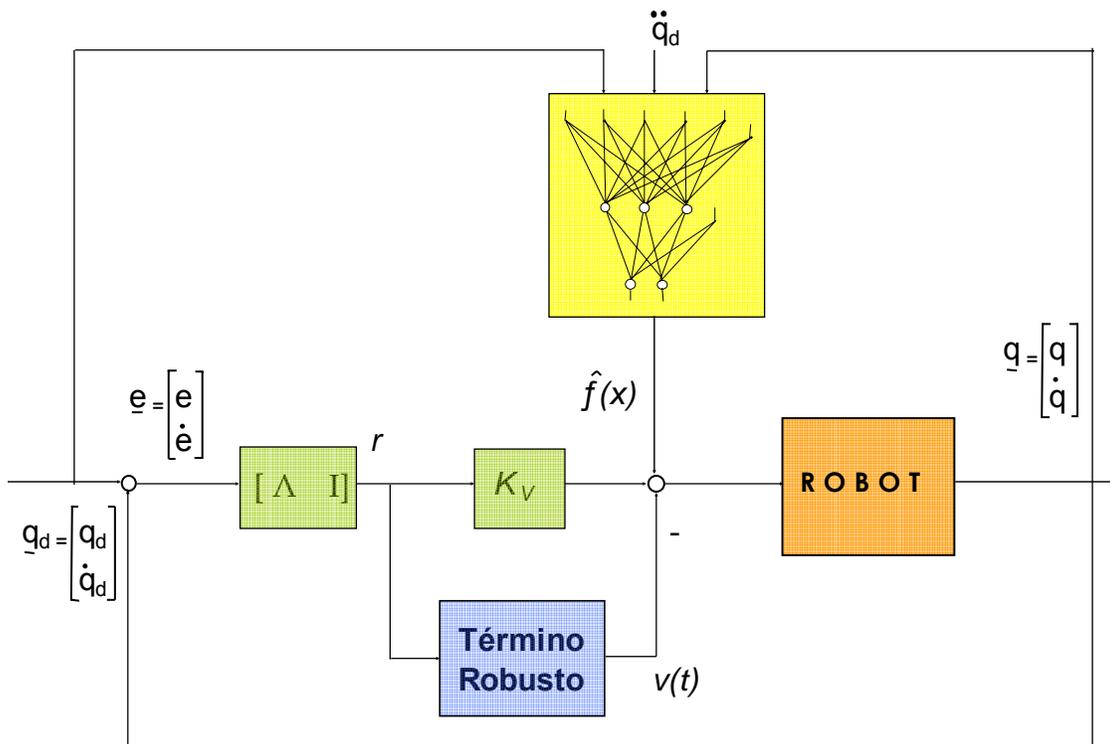
Los términos de retropropagación estandar pueden ser pensados como señales propagadas con retraso en una Red Neuronal de retropropagación no-lineal que contiene multiplicadores. El segundo término en 3.(41) es visto como correspondiente a una onda que viaja hacia delante en la

red de retropropagación que proporciona una corrección de segundo orden para el ajuste de pesos de \hat{W} .

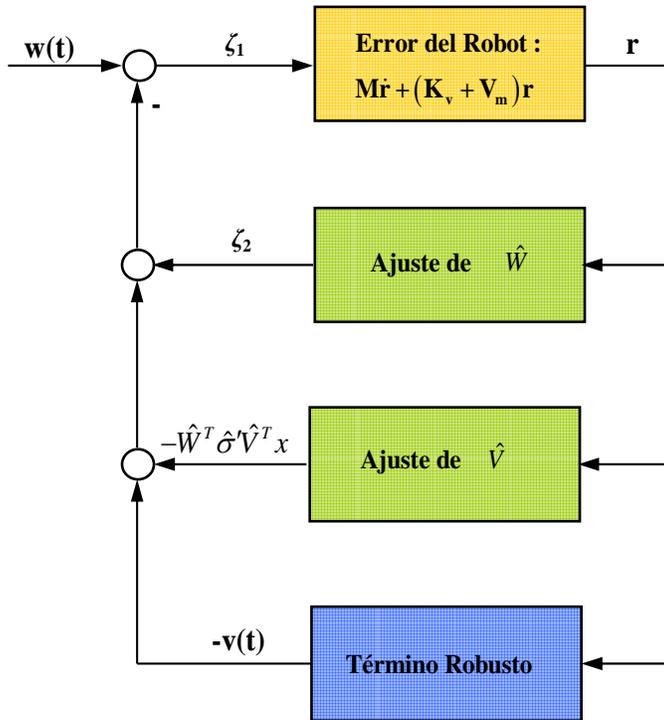
Nótese que hay libertad de diseño en el grado de complejidad de la Red Neuronal (es decir, tamaño). Para una mayor complejidad de la Red Neuronal (es decir, más neuronas en la capa oculta) las constantes de limitación disminuirán, resultando errores de seguimiento más pequeños.

Por otra parte, una Red Neuronal simplificada con pocas neuronas en la capa oculta trae consigo un límite del error más grande, esta degradación puede ser compensada para, tan grande como la cota ε_N es conocida, se selecciona un valor grande para K_z en la señal de robustez $v(t)$, o para Λ en (8.4).

$$r = \dot{e} + \Lambda e \dots\dots\dots (8.4)$$



**CONTROLADOR PD y RED NEURONAL
con término robusto figura 10.3**



Sistema del error en lazo cerrado de la Red Neuronal
 Figura 10.4

10.c. Control por pasividad en la red neuronal

Algunas nociones de estabilidad son necesarias para proceder. Consideremos el sistema no-lineal.

$$\dot{x} = f(x, u, t)$$

$$y = h(x, t)$$

Con el estado $x(t) \in \mathbb{R}^n$. Decimos que la solución es uniformemente últimamente acotada (*UUB*) si existe un conjunto compacto $U \subset \mathbb{R}^n$ tal que para toda $x(t_0) = x_0 \in U$, existe un $\varepsilon > 0$ y un número $T(\varepsilon, x_0)$ tal que $\|x(t)\| < \varepsilon$ para toda $t \geq t_0 + T$. Como puede verse en la demostración de los teoremas, el conjunto compacto U , esta relacionado con el conjunto compacto en el cual la propiedad de aproximación de la Red Neuronal (ecuación 6.d.4) se mantiene. Hay que notar que U puede hacerse mas grande si se seleccionan mas neuronas en la capa oculta.

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x) \dots \dots \dots (6.d.4)$$

Algunos aspectos de pasividad serán importantes subsecuentemente. Un sistema con entrada $u(t)$ y salida $y(t)$ se dice que es pasivo si se verifica una ecuación de la llamada "forma de poder".

$$\dot{L}(t) = y^T u - g(t) \dots\dots\dots (10.20)$$

Con $L(t)$ acotada por abajo y $g(t) \geq 0$. Esto es

$$\int_0^T y^T(\tau) u(\tau) d\tau \geq \int_0^T g(\tau) d\tau - \gamma^2 \dots\dots\dots (10.21)$$

para toda $T \geq 0$ y alguna $\gamma \geq 0$

Se dice que es sistema es disipativo si este es pasivo y en adición

$$\int_0^\infty y^T(\tau) u(\tau) d\tau \neq 0 \text{ implica que } \int_0^\infty g(\tau) d\tau > 0 \dots\dots\dots (10.22)$$

Una clase especial de disipatividad ocurre si $g(t)$ es una llamada función cuadrática de $\|x\|$, con coeficientes limitados, donde $x(t)$ es el estado interno del sistema. Se llama a este estado de pasividad estricta. Entonces la norma L_2 del estado esta acotada por arriba en términos del producto interno L_2 de la salida y entrada (es decir, el poder entregado para el sistema). Esto se usa provechosamente para concluir sobre algunas propiedades de limitaciones internas del sistema sin los supuestos usuales de observabilidad (es decir, persistencia de excitación, estabilidad, etc.)

El sistema del error en lazo cerrado que aparece en la figura 10.4, con la señal ζ_2 definida como:

$$\zeta_2(t) = -\tilde{W}^T \hat{\sigma} \quad \text{para el sistema del error (9.14)}$$

$$\zeta_2(t) = -\tilde{W}^T (\hat{\sigma} - \hat{\sigma}^* \hat{V}^T x) \quad \text{para el sistema del error (9.16)} \dots\dots\dots (10.23)$$

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T \hat{\sigma} + \hat{W}^T \hat{\sigma}^* \tilde{V}^T x + w_1 + v \dots\dots\dots (9.14)$$

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T (\hat{\sigma} - \hat{\sigma}^* \hat{V}^T x) + \hat{W}^T \hat{\sigma}^* \tilde{V}^T x + w + v \equiv -(K_v + V_m)r + \zeta_1 \dots\dots (9.16)$$

En el caso formal, la señal $w(t)$ debe ser reemplazada por $w_1(t)$. Hay que notar el funcionamiento de la Red Neuronal, el cual esta descompuesto en dos bloques efectivos apareciendo en una configuración de retroalimentación típica, en contraste con el funcionamiento de la Red Neuronal del controlador en la figura 10.3.

La pasividad es importante en un sistema de lazo cerrado como garantía de la acotación de señales y de un adecuado funcionamiento, aún en la presencia de perturbaciones imprevistas adicionales tan grandes como sean pero acotadas. En general una Red Neuronal no puede garantizar ser pasiva. Sin embargo los siguientes resultados muestran que los algoritmos de ajuste de

pesos aquí hacen de hecho garantizar propiedades de pasividad deseable de la Red Neuronal y asimismo del sistema en lazo cerrado.

El primer resultado es considerando el sistema de error (9.14)

Teorema 10.4 Algoritmos de ajuste de pesos Pasivos

Los algoritmos de ajuste de pesos por retropropagación

$$\dot{\hat{W}} = F \hat{\sigma} r^T \dots\dots\dots (9.19)$$

$$\dot{\hat{V}} = Gx(\hat{\sigma}'^T \hat{W}r)^T \dots\dots\dots (9.20)$$

(9.19) y (9.20) hacen el primer mapeo de $r(t)$ a $-\tilde{W}^T \hat{\sigma}$, y el segundo mapeo de $r(t)$ a $-\hat{W}^T \hat{\sigma}' \tilde{V}^T x$, ambos mapeos pasivos.

Demostración:

Las dinámicas con respecto a \tilde{W}, \tilde{V} son

$$\dot{\tilde{W}} = -F \hat{\sigma} r^T \dots\dots\dots (10.24)$$

$$\dot{\tilde{V}} = -Gx(\hat{\sigma}'^T \hat{W}r)^T \dots\dots\dots (10.25)$$

1) Seleccionando la función no-negativa

$$L = tr \left(\frac{\tilde{W}^T \tilde{W}}{2F} \right)$$

y evaluando \dot{L} a lo largo de las trayectorias de (10.24) se produce

$$\dot{L} = tr \left(\frac{\tilde{W}^T \dot{\tilde{W}}}{F} \right) = -tr(\tilde{W}^T \hat{\sigma} r^T) = r^T (-\tilde{W}^T \hat{\sigma})$$

el cual es en Forma de Poder (10.20).

2) Seleccionando la función no-negativa

$$L = tr \left(\frac{\tilde{V}^T \tilde{V}}{2G} \right)$$

y evaluando \dot{L} a lo largo de las trayectorias de (10.25) produce

$$\dot{L} = tr \left(\frac{\tilde{V}^T \dot{\tilde{V}}}{G} \right) = -tr(\tilde{V}^T x(\hat{\sigma}'^T \hat{W}r)^T) = r^T (-\hat{W}^T \hat{\sigma}' \tilde{V}^T x)$$

la cual está en Forma de Poder

Así, el sistema de error del robot en la figura 10.4 es Estado Estrictamente Pasivo (SSP) y los bloques de los errores de pesos son pasivos, esto garantiza la disipatividad de el sistema de lazo

cerrado. Usando el Teorema de Pasividad uno puede ahora concluir que las señales entrada-salida de cada bloque están acotadas tanto como las entradas externas están limitadas.

Desafortunadamente, aunque disipativo, el sistema de lazo cerrado no es *SSP* así que, cuando la perturbación $w_1(t)$ es no cero entonces no se producen limitaciones de los estados internos de los bloques de peso (es decir, \tilde{W}, \tilde{V}) a menos que estos bloques sean observables, esto es, excitado persistentemente (*PE*). Desafortunadamente, esto no produce un método conveniente para definir *PE* en Redes Neuronales de tres capas, como los dos bloques ajustadores de pesos están acoplados, formando de hecho un sistema bilineal. Por contraste, las condiciones *PE* para el caso de dos capas $V = I$ (es decir Red Neuronal lineal) son fáciles de deducir.

El siguiente resultado muestra porqué una condición *PE* no es necesaria con el algoritmo de actualización de pesos modificada del Teorema 10.3. Esto es en el contexto del sistema de error (9.16).

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}^T (\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) + \hat{W}^T \hat{\sigma}'\tilde{V}^T x + w + v \equiv -(K_v + V_m)r + \zeta_1 \dots (9.16)$$

Teorema 10.5 *Algoritmos de ajuste de pesos modificados SSP*

Los algoritmos de ajuste de pesos modificados (10.15), (10.16) hacen el mapeo de $r(t)$ a $-\tilde{W}^T (\hat{\sigma} - \hat{\sigma}'\hat{V}^T x)$, y el mapeo de $r(t)$ a $-\hat{W}^T \hat{\sigma}'\tilde{V}^T x$ ambos mapeos *SSP*.

Demostración: La dinámica revisada relativa a \tilde{W}, \tilde{V} están dadas por:

$$\dot{\tilde{W}} = -F\hat{\sigma}r^T + F\hat{\sigma}'\hat{V}^T x r^T + kF\|r\|\hat{W} \dots (10.26)$$

$$\dot{\tilde{V}} = -Gx(\hat{\sigma}'^T \hat{W}r)^T + kG\|r\|\hat{V} \dots (10.27)$$

1) Seleccionando la función no-negativa

$$L = \text{tr} \left(\frac{\tilde{W}^T \tilde{W}}{2F} \right)$$

y calculando \dot{L} resulta

$$\dot{L} = \text{tr} \left(\frac{\tilde{W}^T \dot{\tilde{W}}}{F} \right) = \text{tr} \left\{ \left[-\tilde{W}^T (\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) \right] r^T + k\|r\|\tilde{W}^T \hat{W} \right\}$$

ya que $\text{tr}(\tilde{W}^T (W - \tilde{W})) = \langle \tilde{W}, W \rangle_F - \|\tilde{W}\|_F^2 \leq \|\tilde{W}\|_F \|W\|_F - \|\tilde{W}\|_F^2$ resulta que

$$\begin{aligned} \dot{L} &\leq r^T \left[-\tilde{W}^T (\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) \right] - k\|r\| \left(\|\tilde{W}\|_F^2 - \|\tilde{W}\|_F \|W\|_F \right) \\ &\leq r^T \left[-\tilde{W}^T (\hat{\sigma} - \hat{\sigma}'\hat{V}^T x) \right] - k\|r\| \left(\|\tilde{W}\|_F^2 - W_M \|\tilde{W}\|_F \right) \end{aligned}$$

el cual está en Forma de Poder con la última función cuadrática en $\|\tilde{W}\|_F$

2) Seleccionando la función no-negativa

$$L = \text{tr} \left(\frac{\tilde{V}^T \tilde{V}}{2G} \right)$$

y calculando \dot{L} resulta

$$\begin{aligned} \dot{L} &= \text{tr} \left(\frac{\tilde{V}^T \dot{\tilde{V}}}{G} \right) = r^T \left(-\hat{W}^T \hat{\sigma}' \tilde{V}^T x \right) - k \|r\| \left(\|\tilde{V}\|_F^2 - \langle \tilde{V}, V \rangle_F \right) \\ &\leq r^T \left(-\hat{W}^T \hat{\sigma}' \tilde{V}^T x \right) - k \|r\| \left(\|\tilde{V}\|_F^2 - V_M \|\tilde{V}\|_F \right) \end{aligned}$$

la cual está en Forma de Poder con la última función cuadrática en $\|\tilde{V}\|_F$.

Estas son exactamente las formas especiales de \dot{L} que le permiten a uno mostrar el acotamiento de \tilde{W} y de \tilde{V} cuando los primeros términos (entradas de poder) están acotadas, como en la demostración del Teorema 10.3.

Se debe notar que el *SSP* de ambos, la dinámica del robot y el ajuste de pesos de los bloques garantizan *SSP* para el sistema en lazo cerrado, así que las normas de los estados internos están limitadas en términos del poder liberado para cada bloque. Entonces el acotamiento de las señales de entrada-salida asegura la limitación de los estados sin ninguna otra clase de requerimientos de observabilidad.

Definimos una Red Neuronal como Pasiva si, en la formulación del error se garantiza la pasividad de los subsistemas de ajuste de pesos. Entonces una condición *PE* extra es necesaria para garantizar acotación de los pesos. Definimos una Red Neuronal como Robusta si, en la formulación del error, se garantiza el *SSP* del subsistema de ajuste de pesos. Entonces no es necesaria una condición extra de *PE* para la acotación de los pesos.

Hay que notar que:

1. El *SSP* del sistema de error de la planta en lazo abierto es necesario agregar para la estabilización del seguimiento y
2. Las propiedades de Pasividad de la Red Neuronal dependen del algoritmo de ajuste de pesos usados.

10.d. Controlador final PD y red neuronal con redes aumentadas

10.d.1. Aproximación de la Red Neuronal y la No-linealidad en el problema de Parámetros

Aquí se presenta un controlador de red neuronal basado en una aproximación del error filtrado y empleando una red neuronal aumentada para aproximar las funciones no-lineales desconocidas (10.28) en la dinámica del brazo robótico, la cual incluye fricción.

$$\tau = \tau_0 - v = \hat{W}_1^T \sigma(\hat{V}_1^T x) + \hat{W}_2^T \varphi(V_2^T x) + K_v r - v \dots\dots\dots(10.28)$$

Una seria deficiencia en el uso de controladores estandar adaptativos en robots es el requerimiento de linealidad en los parámetros desconocidos del sistema (*LIP*, *Linealidad en los Parámetros desconocidos del sistema*). El *LIP* restringe severamente la clase de sistemas industriales que pueden ser controlados, y además no considera fricción. La restricción de *LIP* se supera usando la propiedad universal de aproximación de las redes neuronales la cual se mantiene para funciones no-lineales generales. El principal objetivo aquí va a ser el el diseño de un controlador como el de la figura 10.3 pero modificado por la compensación de la fricción. En vez de requerir el conocimiento de la estructura del sistema, actuando como en un control adaptativo, una red neuronal es usada para aproximar la dinámica no modelada. El uso de la red neuronal aumentada da al controlador un mejor desempeño si hay una fricción considerable desconocida, ya que la fricción es discontinua en cero.

Para probar la estabilidad de todo el sistema de lazo cerrado, algunos supuestos que se toman en las situaciones prácticas son requeridos.

Supuesto 1. Trayectoria de referencia limitada.

La trayectoria deseada esta limitada asi que

$$\begin{pmatrix} \|q_d(t)\| \\ \|\dot{q}_d(t)\| \\ \|\ddot{q}_d(t)\| \end{pmatrix} \leq q_B \dots\dots\dots(c.14) \quad \text{con } q_B \text{ un límite escalar conocido}$$

Teorema 10.6. La entrada x esta acotada

Para cada tiempo t , $x(t)$ esta limitada por

$$\|x\| \leq c_1 + c_2 \|r\| \leq 2q_B + c_0 \|r(0)\| + c_2 \|r\| \dots\dots\dots(10.29)$$

para constantes positivas computables c_0, c_1, c_2

Demostración

La solución del sistema *LIP* (8.4) con el vector de valor inicial $q(t_0)$ es:

$$r = \dot{e} + \Lambda e \dots\dots\dots (8.4)$$

$$e(t) = e_0 \varepsilon^{-\Lambda(t-t_0)} + \int_{t_0}^t \varepsilon^{-\Lambda(t-\tau)} r(\tau) d\tau \quad \forall t \geq t_0$$

donde $e_0 = q_d(t_0) - q(t_0)$ Así

$$\|e\| \leq \|e_0\| + \frac{\|r\|}{\sigma_{\min}(\Lambda)}$$

con $\sigma_{\min}(\Lambda)$ el valor singular mínimo de Λ . La entrada de la red neuronal puede ser escrita como:

$$x = \begin{pmatrix} q \\ \dot{q} \\ q_d \\ \dot{q}_d \\ \ddot{q}_d \end{pmatrix} = \begin{pmatrix} -e \\ -r + \Lambda e \\ q_d \\ \dot{q}_d \\ \ddot{q}_d \end{pmatrix} + \begin{pmatrix} q_d \\ \dot{q}_d \\ 0 \\ 0 \\ 0 \end{pmatrix} \dots\dots\dots (10.30)$$

Entonces un límite puede ser dado como

$$\begin{aligned} \|x\| &\leq (1 + \sigma_{\max}(\Lambda)) \|e\| + 2q_B + \|r\| \\ &\leq ((1 + \sigma_{\max}(\Lambda)) \|e_0\| + 2q_B) + \left\{ 1 + \frac{1}{\sigma_{\min}(\Lambda)} + \frac{\sigma_{\max}(\Lambda)}{\sigma_{\min}(\Lambda)} \right\} \|r\| \\ &= c_1 + c_2 \|r\| \end{aligned}$$

con

$$\begin{aligned} c_1 &= [1 + \sigma_{\max}(\Lambda)] \|e_0\| + 2q_B \\ c_2 &= 1 + \frac{1}{\sigma_{\min}(\Lambda)} + \frac{\sigma_{\max}(\Lambda)}{\sigma_{\min}(\Lambda)} \end{aligned}$$

ahora considerando la expresión (8.13) uno tiene $\|e\| < \frac{\|r\|}{\sigma_{\min}(\Lambda)}$ para todo t . De aquí se obtiene que

$$c_0 = \frac{1 + \sigma_{\max}(\Lambda)}{\sigma_{\min}(\Lambda)}$$

La función no-lineal del robot $f(x)$ es una función discontinua debida a la fricción.

$$f(x) = M(\dot{q})(\ddot{q}_d + \Lambda \dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(q, \dot{q}) \dots\dots\dots (10.31)$$

Mas aún, esta función es desconocida y debe estimarse por el sistema de control tanto que pueda ser compensada por este. Usando la propiedad universal de aproximación de las redes neuronales y los resultados para la aproximación de funciones con salto, hay una red de dos capas tal que:

$$f(x) = W_1^T \sigma(V_1^T x) + W_2^T \phi(V_2^T x) + \varepsilon(x) \dots\dots\dots (10.32)$$

con la aproximación del error limitada a un conjunto compacto por $\|\varepsilon(x)\| < \varepsilon_N$, con ε_N un límite conocido.

Siendo W_1, W_2, V_1 y V_2 los *pesos objetivo ideales* que dan una buena aproximación de $f(x)$. Ellos son desconocidos y pueden no ser únicos. De cualquier modo es requerido que estos existan. En la derivación del controlador de red neuronal, entonces se muestra como el ajuste los pesos de las redes neuronales se puedan acercar lo bastante a los pesos ideales.

Se definen $\hat{W}_1, \hat{W}_2, \hat{V}_1$ y \hat{V}_2 como estimaciones de los pesos ideales de las redes neuronales, los cuales los cuales son dados por los algoritmos de ajuste de las redes. Ya que los pesos de V_2^T están dados por el diseño dependiendo de la localización del salto, entonces $V_2^T = \hat{V}_2^T$ Se definen las estimaciones del error como:

$$\tilde{W}_1 = W_1 - \hat{W}_1, \quad \tilde{W}_2 = W_2 - \hat{W}_2 \quad \text{y} \quad \tilde{V}_1 = V_1 - \hat{V}_1 \dots\dots\dots (10.33)$$

La aproximación de funciones no-lineales como en (10.31) es:

$$\hat{f}(x) = \hat{W}_1^T \sigma(\hat{V}_1^T x) + \hat{W}_2^T \phi(V_2^T x) \dots\dots\dots (10.34)$$

Sea Z la matriz de todos los pesos ajustados de la red

$$Z = \begin{pmatrix} W_1 & 0 & 0 \\ 0 & W_2 & 0 \\ 0 & 0 & V_1 \end{pmatrix} \dots\dots\dots (10.35)$$

Se debe notar que los pesos de V_2 no están incluidos en (10.35) ya que estos son conocidos y se seleccionan de acuerdo a la localización conocida del salto, la cual ocurre en el origen para la fricción.

Se requiere hacer el siguiente *supuesto*, el cual siempre se mantiene. Es simplemente establecer que se pueden seleccionar en un número bastante grande para cubrir los límites de los desconocidos pesos ideales de la red.

Supuesto 2. Acotación de los pesos objetivo ideales de la red neuronal

Sobre cualquier subconjunto compacto de \mathfrak{R}^n los pesos ideales de la red están limitados tanto como

$$\|Z\|_F \leq Z_B \dots\dots\dots (10.36)$$

con Z_B conocido

Ahora se mostrará como superar el problema *LIP*, el cual comúnmente ha restringido el desempeño de los algoritmos de ajuste del control adaptativo para sistemas que son *LIP*. La red

neuronal es no lineal en los parámetros ajustables, ya que los pesos ajustados \hat{V}_1 están en los argumentos de las funciones de activación no lineales.

Se define el error de salida de la capa oculta para un vector x dado como:

$$\tilde{\sigma} = \sigma(V_1^T x) - \sigma(\hat{V}_1^T x) \dots\dots\dots (10.37)$$

Usando las series de Taylor, se tiene:

$$\sigma(V_1^T x) = \sigma(\hat{V}_1^T x) + \sigma'(\hat{V}_1^T x)\tilde{V}_1^T x + O(\tilde{V}_1^T x)^2 \dots\dots\dots (10.38)$$

donde σ' denota la primer derivada y $O(\tilde{V}_1^T x)^2$ representa los términos de alto orden. La importancia de esta ecuación es que esta reemplaza $\tilde{\sigma}$ la cual es no-lineal en \tilde{V}_1 por una expresión lineal en \tilde{V}_1 mas los términos de alto orden. El Jacobiano $\sigma'(\hat{V}_1^T x)$ es computable en términos de señales medidas $x(t)$ y las actualizaciones de los pesos de \hat{V}_1 .

Es mostrado por Lewis, Yesildirek y Lu (1996) que los términos de alto orden en (10.38) estan limitados por:

$$\|O(\tilde{V}_1^T x)^2\| \leq c_3 + c_4 q_B \|\tilde{V}_1\|_F + c_5 \|\tilde{V}_1\|_F \|r\| \dots\dots\dots (10.39)$$

para algunas constantes c_3, c_4, c_5

10.d.2. Estructura del controlador y dinámica del sistema de error

La entrada de control para el robot manipulador es seleccionada como:

$$\tau = \tau_0 - v = \hat{W}_1^T \sigma(\hat{V}_1^T x) + \hat{W}_2^T \phi(V_2^T x) + K_v r - v \dots\dots\dots (10.40)$$

donde el torque $\tau_0(t)$ está definido por (8.8).

$$\tau_0 = \hat{f} + K_v r \dots\dots\dots (8.8)$$

La estimación de la función no-lineal ha sido proporcionada por la red neuronal como en (10.34).

$$\hat{f}(x) = \hat{W}_1^T \sigma(\hat{V}_1^T x) + \hat{W}_2^T \phi(V_2^T x) \dots\dots\dots (10.34)$$

La función $v(t)$ es un término que robustece que va a ser seleccionado mas tarde para ayudar a garantizar la estabilidad de el sistema en lazo cerrado. La estructura de este controlador de red neuronal se muestra en la Figura 10.3. El neurocontrolador es un sistema de control retroalimentado multilazo, con un lazo de seguimiento *PD* exterior, y un lazo interno de red neuronal que compensará las no-linealidades desconocidas, incluyendo la fricción. Este lazo interno que cancela las no-

linealidades desconocidas por la estimación de ellas es conocido como un lazo de linealización retroalimentado.

La dinámica del error en lazo cerrado es usada para demostrar la estabilidad, está dado por:

$$M\dot{r} = -(K_v + V_m)r + W_1^T \sigma(V_1^T x) + W_2^T \phi(V_2^T x) - \hat{W}_1^T \sigma(\hat{V}_1^T x) - \hat{W}_2^T \phi(V_2^T x) + \varepsilon + \tau_d + \nu \dots (10.41)$$

Usando (10.37) se tiene

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}_1^T \sigma + W_1^T \tilde{\sigma} + \tilde{W}_2^T \phi + \varepsilon + \tau_d + \nu \dots (10.42)$$

Aplicando la aproximación con series de Taylor para $\tilde{\sigma}$, se puede obtener la expresión para la dinámica del error en lazo cerrado.

$$M\dot{r} = -(K_v + V_m)r + \tilde{W}_1^T (\hat{\sigma} - \hat{\sigma}' \hat{V}_1^T x) + \hat{W}_1^T \hat{\sigma}' \hat{V}_1^T x + \tilde{W}_2^T \phi + \omega + \nu \dots (10.43)$$

El término de perturbaciones viene de los términos de mas alto orden en las series de Taylor y esta dado por:

$$\omega = \tilde{W}_1^T \hat{\sigma}' \hat{V}_1^T x + W_1^T O(\tilde{V}_1^T x)^2 + \varepsilon + \tau_d \dots (10.44)$$

se muestra en Lewis, Yesildirek y Lu (1996) que este término por perturbación esta limitado por:

$$\|\omega(t)\| \leq C_0 + C_1 \|\tilde{Z}\|_F + C_2 \|\tilde{Z}\|_F \|r\| \dots (10.45)$$

con c_i constantes positivas conocidas

10.d.3. ajuste de pesos de la red neuronal y análisis de estabilidad

La estructura del controlador de red neuronal esta dado en la Figura 10.3. Se necesita mostrar como ajustar los pesos de la red neuronal de tal manera que la estabilidad en lazo cerrado esté garantizada. Se usarán las leyes de ajuste de pesos de redes neuronales, derivadas de la teoría de Lyapunov, que garantizan el seguimiento estable con estabilidad interna. Se muestra que el error de seguimiento es convenientemente pequeño, los pesos de las redes neuronales son limitados y el control $\tau(t)$ esta acotado. Los pesos de las redes son ajustados *en línea* (on line) en tiempo real, sin ser requerido el aprendizaje preliminar.

El siguiente supuesto especifica la región de convergencia del controlador neuronal de dos capas.

Supuesto 3. Requerimientos de condiciones iniciales.

Se supone que la trayectoria deseada $q_d, \dot{q}_d, \ddot{q}_d$ esta limitada por q_B como en el Supuesto 1.

Se definen las constantes conocidas c_0, c_2 por el Lema siguiente. Sea la propiedad de aproximación de Redes Neuronales

$$f(x) = W_1^T \sigma(V_1^T x) + W_2^T \phi(V_2^T x) + \varepsilon(x) \dots (10.32)$$

tomada para la función desconocida $f(x)$ dada en

$$f(x) = M(\dot{q})(\ddot{q}_d + \Lambda \dot{e}) + V_m(q, \dot{q})(\dot{q}_d + \Lambda e) + G(q) + F(q, \dot{q}) \dots (10.31)$$

con una exactitud dada por ε_N para toda x en un conjunto compacto $S_x = \{x \mid \|x\| < b_x\}$ con

$$b_x > q_B. \text{ Se define } S_r = \left\{ r \mid \|r\| < \frac{(b_x - q_B)}{(c_0 + c_2)} \right\} \text{ Sea } r(0) \in S_r.$$

El conjunto S_r especifica el conjunto de los errores de seguimiento iniciales permitidos $r(0)$.

La idea es que el robot deba iniciar en un punto posicionado bastante cerca de la trayectoria deseada. Hay que notar que la exactitud de la aproximación de las redes neuronales determina la magnitud permitida del error de seguimiento inicial $r(0)$. Para una Red Neuronal mas grande (es decir con mas unidades en la capa oculta) ε_N es pequeña para un radio mas grande de b_x ; así el conjunto de condiciones iniciales permitidas S_r es mas grande. La clave del rol de las condiciones iniciales requeridas esta en presentar la dependencia del conjunto de las condiciones iniciales S_r sobre los parámetros de diseño. Las constantes q_B, c_0, c_2 y b_x no necesitan ser explícitamente determinadas.

Una característica clave del requerimiento de condiciones iniciales es la independencia de los pesos iniciales de la red neuronal. Esto en contraste con otras técnicas donde las demostraciones de estabilidad dependen de la selección de algunos pesos iniciales estabilizantes de la red neuronal, lo cual es muy difícil de hacer.

El siguiente resultado principal muestra como seleccionar el término robustecedor $v(t)$ y el ajuste de la red de tal manera que muestre que el sistema en lazo cerrado es estable. Muestra que el error de seguimiento es *UUB* (acotación última uniforme). Este resultado completa el diseño de la compensación de la fricción del neurocontrolador. Se muestra que el algoritmo de ajuste por retropropagación estándar debe ser modificado para sistemas de control con retroalimentación.

Teorema 10.7 Ajuste de pesos con retropropagación aumentada

Sea la trayectoria deseada limitada por q_B como en el **Supuesto 1** y el error inicial de seguimiento $r(0)$ satisface la condición inicial del **Supuesto 3**. Sean los pesos objetivo ideales, limitados como en el **Supuesto 2**. Tómesese la entrada de control para la dinámica del robot como en (10.40)

$$\tau = \tau_0 - v = \hat{W}_1^T \sigma(\hat{V}_1^T x) + \hat{W}_2^T \varphi(V_2^T x) + K_v r - v \dots (10.40)$$

con una ganancia que satisfaga

$$K_{v\min} > \frac{\left(C_0 + \frac{kC_3^2}{4}\right)(C_0 + C_2)}{b_x - q_B} \dots\dots\dots (10.46)$$

donde c_3 está definido en la demostración y c_0 está definido en (10.45). Sea el término robustecedor

$$v(t) = -K_z \left(\|\hat{Z}\|_F + Z_B r \right) \dots\dots\dots (10.47)$$

con ganancia

$$K_z > C_2 \dots\dots\dots (10.48)$$

Sea el ajuste de pesos de la red neuronal aumentada proporcionado por

$$\dot{\hat{W}}_1 = S \hat{\sigma} r^T - S \hat{\sigma}' \hat{V}_1^T x r^T - kS \|r\| \hat{W}_1 \dots\dots\dots (10.49)$$

$$\dot{\hat{V}}_1 = T x r^T \hat{W}_1^T \hat{\sigma}' - kT \|r\| \hat{V}_1 \dots\dots\dots (10.50)$$

$$\dot{\hat{W}}_2 = E \varphi r^T - kE \|r\| \hat{W}_2 \dots\dots\dots (10.51)$$

con cualesquiera matrices constantes $S = S^T > 0$, $T = T^T > 0$, $E = E^T > 0$ y $k > 0$ un parámetro de diseño con escalar pequeño. Entonces el error de seguimiento filtrado $r(t)$ y los pesos estimados de la red neuronal \hat{V}_1, \hat{W}_1 y \hat{W}_2 son UUB con límites dados por (10.52) y (10.53). Además el error de seguimiento puede ser conservado tan pequeño como se desee por el incremento de las ganancias K_v en (10.40)

Demostración:

Seleccionando una función de Lyapunov como:

$$L = \frac{1}{2} r^T M r + \frac{1}{2} \text{tr} \left(\tilde{W}_1^T S^{-1} \tilde{W}_1 \right) + \frac{1}{2} \text{tr} \left(\tilde{V}_1^T T^{-1} \tilde{V}_1 \right) + \frac{1}{2} \text{tr} \left(\tilde{W}_2^T E^{-1} \tilde{W}_2 \right)$$

Derivándola

$$\dot{L} = r^T \dot{M} r + \frac{1}{2} r^T \dot{M} r + \frac{1}{2} \text{tr} \left(\tilde{W}_1^T S^{-1} \dot{\tilde{W}}_1 \right) + \frac{1}{2} \text{tr} \left(\tilde{V}_1^T T^{-1} \dot{\tilde{V}}_1 \right) + \frac{1}{2} \text{tr} \left(\tilde{W}_2^T E^{-1} \dot{\tilde{W}}_2 \right)$$

Asumiendo que el error de seguimiento inicial satisface la condición inicial del Supuesto 3 incluyendo la dinámica de error del sistema (10.43) se tiene que:

$$\begin{aligned} \dot{L} = & -r^T (K_v + V_m) r + r^T \tilde{W}_1^T (\hat{\sigma} - \hat{\sigma}' \hat{V}_1^T x) + r^T \hat{W}_1^T \hat{\sigma}' \hat{V}_1^T x + r^T \tilde{W}_2^T \varphi \\ & + r^T \omega + r^T v + \frac{1}{2} r^T \dot{M} r + \text{tr} \left(\tilde{W}_1^T S^{-1} \dot{\tilde{W}}_1 \right) + \text{tr} \left(\tilde{V}_1^T T^{-1} \dot{\tilde{V}}_1 \right) + \text{tr} \left(\tilde{W}_2^T E^{-1} \dot{\tilde{W}}_2 \right) \end{aligned}$$

Usando la propiedad 3 de la dinámica del robot y las reglas de ajuste se produce:

$$\begin{aligned} \dot{L} &= -r^T K_v r + k \|r\| \operatorname{tr} \left\{ \tilde{W}_1^T (W_1 - \tilde{W}_1) \right\} + k \|r\| \operatorname{tr} \left\{ \tilde{V}_1^T (V_1 - \tilde{V}_1) \right\} \\ &+ k \|r\| \operatorname{tr} \left\{ \tilde{W}_2^T (W_2 - \tilde{W}_2) \right\} + r^T (\omega + \nu) \\ &= -r^T K_v r + k \|r\| \operatorname{tr} \left\{ \tilde{Z}^T (Z - \tilde{Z}) \right\} + r^T (\omega + \nu) \end{aligned}$$

Se define

$$C_3 = Z_B + \frac{C_1}{k}$$

Procediendo como en Lewis (1996) uno obtiene que \dot{L} es negativo tan amplio como cualquiera de

los dos $\|r\| > \frac{C_0 + \frac{kC_3^2}{4}}{K_{v\min}} \equiv b_r \dots\dots\dots (10.52)$

o $\|\tilde{Z}\|_F > \frac{C_3}{2} + \sqrt{\frac{C_0}{k} + \frac{C_3^2}{4}} \equiv b_z \dots\dots\dots (10.53)$

Por consiguiente \dot{L} es negativa externa a un conjunto compacto y $\|r\|$ y $\|\tilde{Z}\|_F$ son *UUB* tan amplio como el resto del control válido dentro de este conjunto. Sin embargo el límite de la ganancia *PD* (10.46) implica que el conjunto compacto definido por $\|r\| \leq b_r$ esta contenido en S_r asi que la propiedad de aproximación se mantiene en todo. Q.E.D.

En la práctica, no es necesario conocer las constantes c_0, c_2, C_0 y C_3 para seleccionar la ganancia K_v del *PD* de acuerdo a (10.46). Se puede seleccionar sólo una ganancia bastante grande como se muestra en las simulaciones.

10.e. Simulaciones

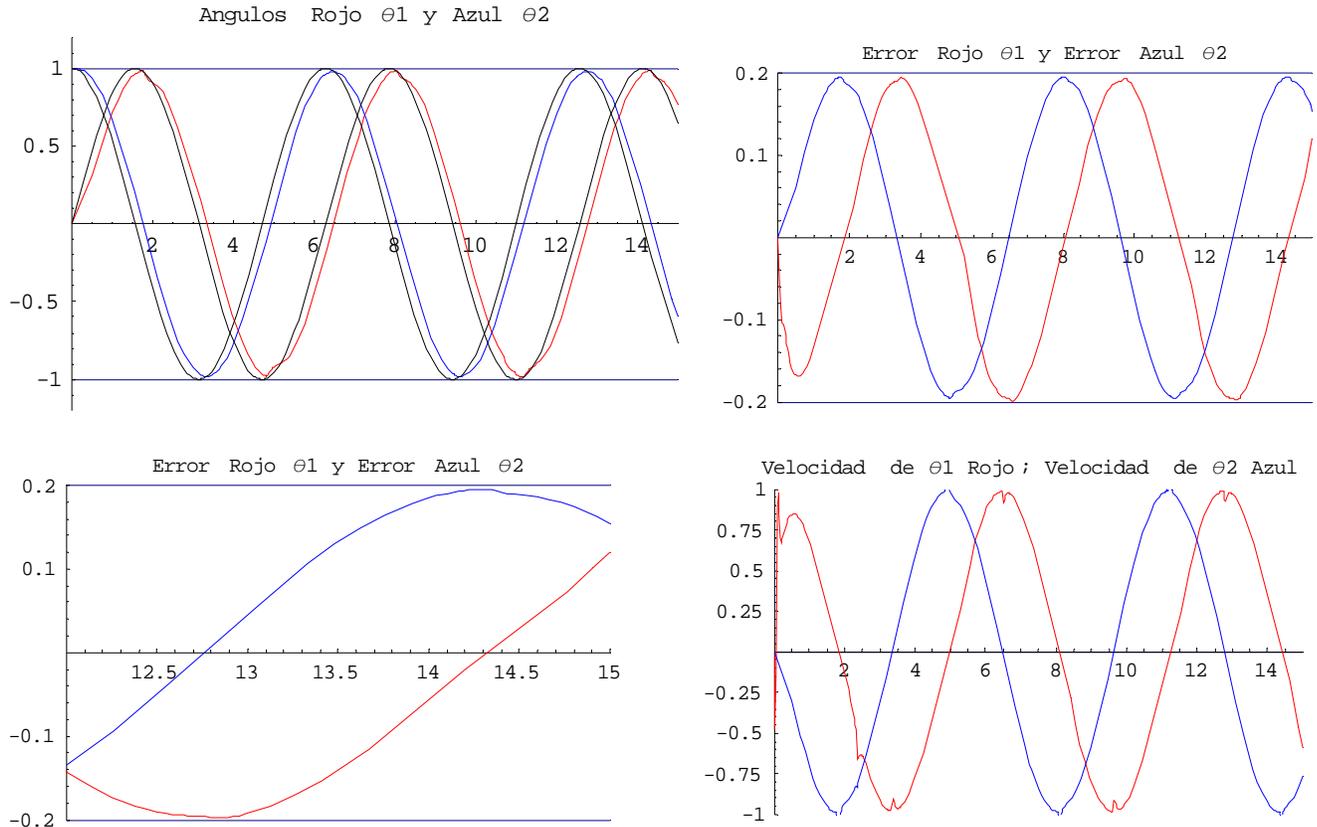
Simulación 10.1

Brazo con dos eslabones, Con fricción, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

Programa de Mathematica **PD+RNA_Model-Cancel**

Diagonal K = 30, Diagonal $\Lambda = 5$ Neuronas Normales: 10, Diagonal T = 30, Diagonal S = 30

Neuronas Adicionales: 8, Diagonales $\alpha, \beta, \gamma, \eta, \lambda, \mu = 35$



Observaciones:

Los diferentes efectos no-lineales causados por la fricción que se observan en las gráficas de la Simulación 9.6 en el cual el sistema tiene un Controlador PD y una Red Neuronal aquí desaparecen. La utilización de neuronas adicionales con funciones de activación para saltos del Controlador PD y Red Neuronal Aumentada Robusta adapta y modela las perturbaciones causadas por la fricción y luego las cancela con el término de robustez que está integrado en la ecuación diferencial que actualiza los pesos. En las gráficas que ilustran el error se puede observar el resultado de esta cancelación y como el error vuelve a tener una representación muy parecida a la función de la trayectoria de referencia, como si no tuviese el efecto causado por la fricción.

Para la programación de esta simulación se consideraron 4 neuronas adicionales ($\varphi_0, 2\varphi_1, \varphi_2$) con signo positivo y 4 con signo negativo (lo cual asegura la modelación del error en depresiones y en saltos. En este caso los valores iniciales de los pesos como en todos los casos son fijos; estos normalmente son cero pero en esta simulación son fijados en -0.1

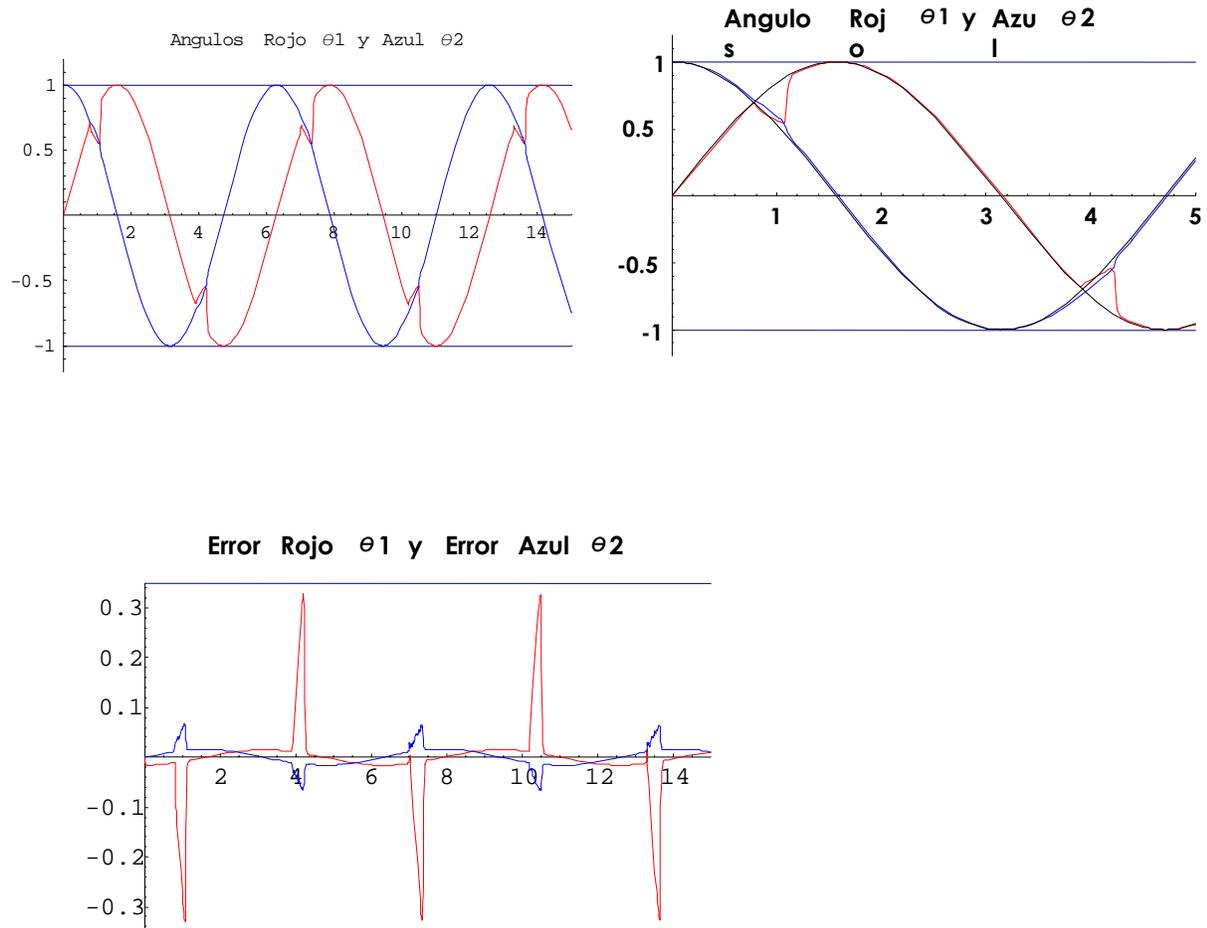
Simulación 10.2

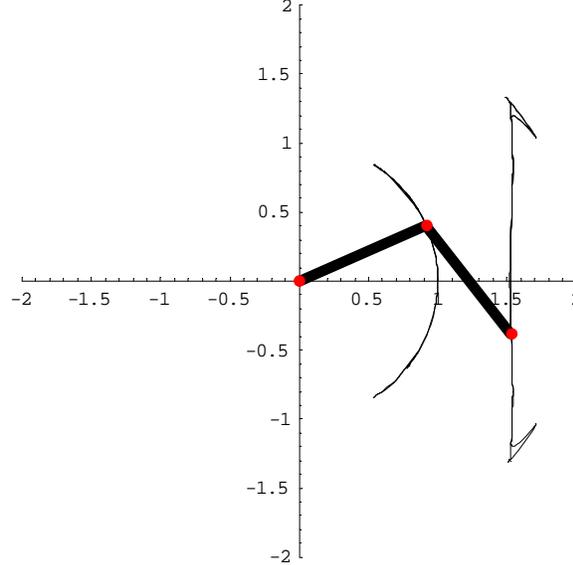
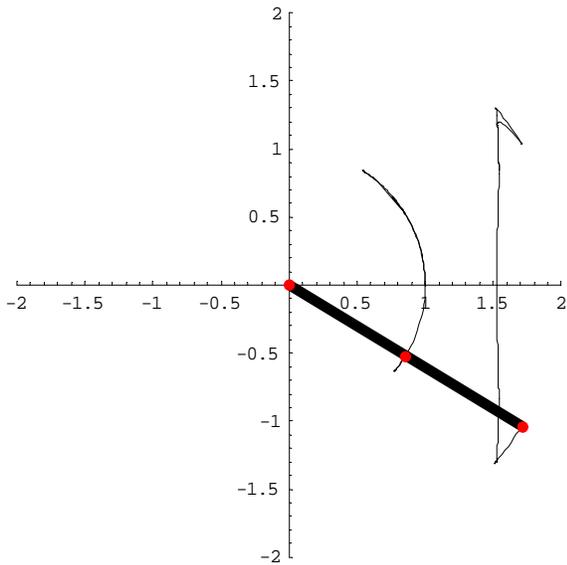
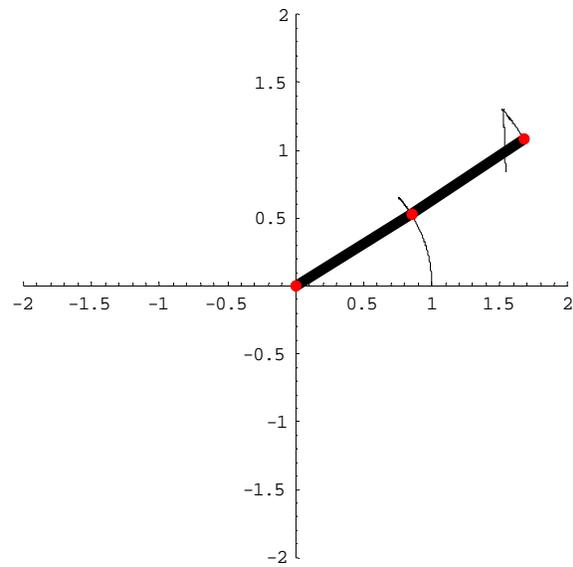
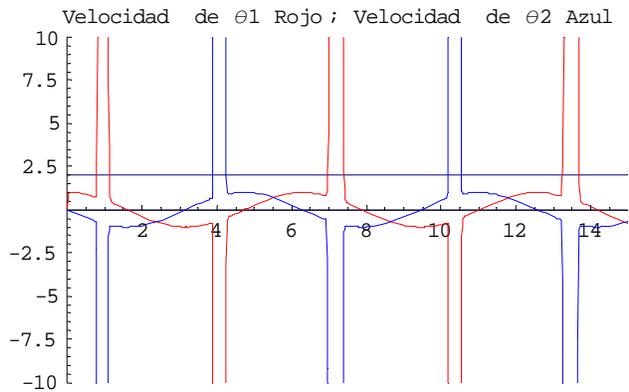
Brazo con dos eslabones, Con fricción, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

Programa de Mathematica **PD+RNA_Bifurcación**

Diagonal K = 379, Diagonal $\Lambda = 63$ Neuronas normales: 10, Diagonal T = 30, Diagonal S = 30,

Neuronas adicionales = 8, Diagonal G = 35, Diagonal $\beta = 35$





Observaciones:

En esta simulación se observa que el sistema comienza a ser inestable cuando se acerca a un valor máximo de la ganancia K que es límite. Aquí en los puntos donde los brazos se alinean comienza a ser inestable por un efecto de bifurcación.

Ahora para alcanzar mayor precisión en el seguimiento de la trayectoria de referencia manteniéndose el sistema estable es necesario aumentar sólo la matriz diagonal de la ganancia Λ .

Simulación 10.3

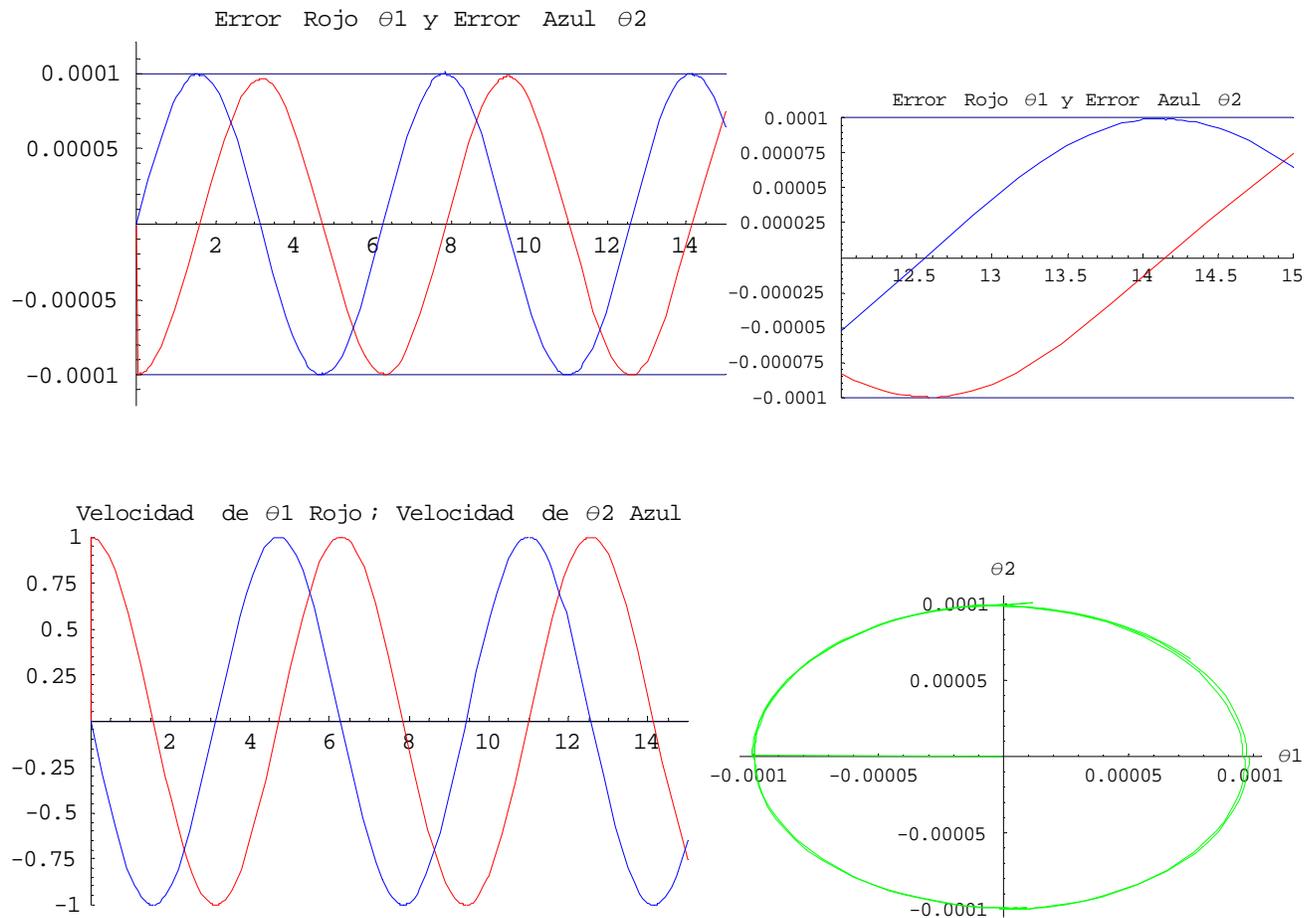
Brazo con dos eslabones, Con fricción, Trayectoria de referencia: $\phi_1 = \text{Sin}[t], \phi_2 = \text{Cos}[t]$

Programa de Mathematica **PD+RNA_CFricción_traySC_Aprox.Final**

Diagonal K = 380; Diagonal $\Lambda = 10000$, Neuronas normales: 10, Diagonal T = 30,

Diagonal S =30,

Neuronas adicionales = 8, Diagonales $\alpha, \beta, \gamma, \eta, \lambda, \mu = 35$



ERROR Gráfica Paramétrica

Observaciones:

En esta simulación la tolerancia del error es de una diezmilésima de radián. En la gráfica del error no aparecen alteraciones porque este Controlador cumple bien su función de modelar bien las perturbaciones y las cancela usando el término de robustez. Esta gráfica del error al igual que la de velocidades muestran que se puede reducir aún mas el tamaño del error con un valor mayor en la matriz diagonal de ganancia Λ conservando estable el sistema. Los valores de los pesos fueron inicializados en -0.1.

11. OBSERVACIONES Y CONCLUSIONES

11.a. Observaciones

Los pesos de las redes neuronales son inicializados todos en cero o algún valor conveniente de tal manera que cada vez que se corra el programa los pesos finales siempre serán los mismos. Los pesos de V_2 para la Red Neuronal Aumentada no son ajustados ya que son fijos y seleccionados en base a la localización de la discontinuidad, la cual ocurre en el origen para la fricción.

Los algoritmos de ajuste de pesos de la red neuronal son una forma aumentada de la retropropagación. Los primeros términos de (10.49), (10.50) y (10.51) son versiones modificadas del algoritmo estandar de retropropagación donde el error de seguimiento es la señal retropropagada.

El último término corresponde a la *modificación-e* es el término robusto, el cual garantiza la estimación de pesos limitada.

Por otro lado (10.43) muestra que los errores de los pesos en la red están principalmente limitados por Z_B . En situaciones prácticas, no se tiene preocupación de hacia donde convergen los pesos de la red para aproximarse a la función desconocida del robot $f(x)$. El teorema muestra que en un sentido práctico el error de seguimiento $r(t)$ es sin embargo limitado a un pequeño valor. El parámetro de ajuste k permite un diseño intercambiante entre las magnitudes relativas de $\|r\|$ y de $\|\tilde{Z}\|_F$.

Es evidente que hay un diseño libre en el grado de complejidad (es decir L). A mayor complejidad de la red (es decir mas unidades en la capa oculta), la estimación del error de la red decrece, mientras la constante C_0 de límite decrecerá, resultando errores pequeños de seguimiento. Por otro lado, una simplificada red con pocas unidades ocultas produce un límite grande del error, esta degradación puede ser compensada con un valor alto en la ganancias K y Λ del *PD*.

El Controlador PD y RNA con compensación de fricción que se presenta aquí no requiere *LIP* (linealidad en parámetros). Aproximaciones estandar para la compensación de fricción requieren comúnmente alguna previa linealización del modelo de fricción para alcanzar *LIP*. El requerimiento *LIP* es una severa restricción para los sistemas prácticos.

Aquí la propiedad de aproximación de las redes es usada para evitar el *LIP*. El modelo de fricción no lineal es lineal con respecto a las funciones de activación de la red no lineales, lo cual es una diferencia fundamental para la condición *LIP*.

11.b. Conclusiones

El Controlador PD funciona bien en los casos en que se busca alcanzar una posición determinada y permanecer ahí por algún tiempo, sin embargo cuando se desea alcanzar una tolerancia del error muy baja, se requiere el utilizar valores de ganancias muy elevados.

El Controlador PD esta restringido, para seguir una trayectoria sinusoidal, a que las funciones que modele sean continuas y con no-linealidades suaves como se vio en las simulaciones 8.3 y 8.4. Al incorporarse al sistema las perturbaciones, en este caso la provocada por la fricción, la cual es no-lineal y pronunciada, el controlador no tiene la capacidad de modelar esta perturbación y tiende a rechazarla; sin embargo, no dejan de estar presentes alteraciones en ciertos puntos críticos donde se acentúa el efecto de la fricción. Lo anterior trae consigo la presencia constante de un riesgo de inestabilidad y movimientos locales erráticos del brazo.

El Controlador PD con Red Neuronal Normal por medio de su capacidad adaptación modela bien la trayectoria sinusoidal y conforme se aumenta paulatinamente el nivel de aproximación a la trayectoria de referencia el sistema adapta y corrige el efecto de la perturbación. Sin embargo como se presentó en el inciso 9 con el sistema de actualización de pesos con retropropagación únicamente, no puede garantizar en todos los casos con perturbaciones que los pesos de la red se mantengan acotados. El controlador PD con Red Neuronal Aumentada modela convenientemente la perturbación y al tener incluido un término de robustez en el controlador, este cancela el efecto indeseable de la perturbación y mantiene los pesos acotados garantizando un seguimiento de la trayectoria siempre estable. Los conceptos de adaptabilidad y robustez son efectos contrarios y al mismo tiempo complementarios, es decir un nivel de robustez alto reduce la capacidad de adaptabilidad de la red, por lo tanto, hay que lograr un equilibrio conveniente de las dos propiedades en la red. Se incorpora el concepto de pasividad para asegurar un conveniente manejo de las energías dentro del sistema y su estabilidad.

En la búsqueda de los valores de las matrices diagonales ganancias K y Λ que permitan aproximar el sistema controlado con redes neuronales a la trayectoria de referencia se encuentra el límite del valor de la matriz de valores de K a través de la ejercitación sobre el programa. El procedimiento es entonces primero partir de los valores de K dados por los coeficientes del término de Coriolis y los valores de Λ conforme la regla empírica del 5 al 20% e ir aumentando poco a poco estos valores hasta que se llegue al valor de K en el cual se presenta el efecto de bifurcación donde el sistema empieza a ser inestable, esto es cerca a su valor límite, ahora sólo es necesario aumentar el valor de Λ y el sistema seguirá la trayectoria de referencia tanto como se incremente este último valor, manteniéndose estable. El diseño del Controlador PD mas la Red Neuronal Aumentada, representa la aplicación de una técnica de modelado y control que combina convenientemente una base teórica suficientemente fundamentada que permite que se afinen y se garanticen soluciones estables para a los requerimientos de problemas de control bajo la acción de efectos reales de perturbaciones, con una forma no complicada y práctica de realizarse.

REFERENCIAS

- R.1 Barrón A.R. Universal approximation bounds for superposition function , IEEE Trans. Inf. Theory , 39 pp. 930-945 (1993).
- R.2 Blanchard, P. Devaney, R.L. y Hall, G. R. . "Ecuaciones Diferenciales". Thompson (1999).
- R.3 Bose N.K. and Liang P. " Neural Network Fundamentals " , McGraw-Hill , New York, 1996.
- R.4 Canudas de Wit H. Olsson , Astrom K.J. A new model for control of systems with friction , IEEE Trans. Automatic Control , 40 419-425 , 1995.
- R.5 Cervantes, I y Alvarez-Ramírez, J.. On the PID tracking control of robot manipulators. Systems and Control Letters, 42, N. 1, pp. 37-46(10) (2001).
- R.6 Craig, J.J.. "Robótica" (3a. ed.). Pearson Prentice Hall (2006).
- R.7 Desoer, C.A. , Liu,R. and Auth, Jr.,L.V. Linearity vs nonlinearity and asymptotic stability in the large , IRE Transactions on Circuit Theory 12:117-118 March 1965.
- R.8 Domínguez, S., Campoy, P. Sebastián, J.M y Jiménez, A. "Control en el Espacio de Estado". Pearson Educación, España (2006).
- R.9 Duarte, M.A. , and Narendra ,K.S. Combined direct and indirect approach to adaptive control , Technical Report No. 8711 Center for Systems Science, Yale University, 1987.
- R.10 Hahn, W. " Stability of motion " New York-Springer-Verlag, 1957.
- R.11 Haykin S. " Neural networks " , IEEE Press and Macmillan , New York , 1994.
- R.12 Jagannathan S. Robust backstepping control of robotic systems using neural networks in Proceedings of the 37th IEEE Conference on Decision and Control , IEEE Computer Society Press, Los Alamitos CA 943-948 , 1998.
- R.13 Kalman and Bertram , "Control systems analysis and desing via second method of Lyapunov" Journal of Basic Engineering 82: 371-392 June 1960.
- R.14 Karakasoglu A. , Sudharsanan S.I. Identification and decentralized adaptive control using dynamical neural networks with application to robots manipulators, IEEE Trans. Neural Networks 4: 919-930 , 1993.
- R.15 Kim Y.H. and Lewis F.L. High-Level Feedback Control with Neural Networks , World Scientific , River Edge. NJ., 1998.
- R.16 Kosmatopoulos M. Polycarpou M. High-order neural network structures for identification of dynamical systems IEEE Trans Neural Networks , 6 : 422-431 , 1995.
- R.17 Kuo, B.C y Golnaraghi, F. "Automatic Control Systems". Wiley (2002).
- R.18 Lewis, Campos y Selmic. "Neuro-Fuzzy Control of Industrial Systems with Actuator Nonlinearities". Frontiers in Applied Mathematics SIAM (2002).
- R.19 Lewis F.L. , Jagannathan S. and Yesildirek A. Neural Networks Control of Robot Manipulators and Nonlinear Systems , Taylor and Francis , London 1999.

- R.20 Lewis, F.L., Yesildirek, A. y Liu, K. Multilayer Neural-Net Controller with Guaranteed Tracking Performance. *IEEE Trans. on Neural Networks*, vo. 7, No. 2, pp. 388-399 (1996).
- R.21 Miller W.T. Sutton R.S. and Werbos P.J. *Neural Networks for Control* , MIT Press , Cambridge MA. 1991.
- R.22 Moreno, L., Garrido-Valero, M.S. y Balaguer, C. "Ingeniería de control: Modelado y Control de Sistemas Dinámicos". Ariel, España (2003).
- R.23 Narendra K.S. and Annaswamy A.M. . A new adaptive law for robust adaptation without persistent excitation *IEEE Trans. Automatic Control* 32:134-145 , 1987.
- R.24 Narendra, K.S. y Annaswamy, A.M. "Stable Adaptive Systems". Dover (2005).
- R.25 Narendra, K.S. and Valavani L.S. Stable adaptive observers and controllers. *Proceedings of the IEE* 64: 1198-1208 August 1976.
- R.26 Ogata, K. "Ingeniería de Control Moderna" (4ª. ed.). Pearson Universitario (2003).
- R.27 Osburn , P.V. , Whitaker, H.P. and Kezer A. , New developments in the design of model reference adaptive control systems. In *Proceedings of the IAS 29th Annual Meeting* , New York, NY ,1961.
- R.28 Polycarpou M. M. Stable adaptive neural control scheme for nonlinear systems . *IEEE Trans Automatic Control*. 41: 447-451 , 1996.
- R.29 Selmic R.R. *Neurocontrol of Industrial Motion Systems with Actuator Nonlinearities* , Ph.D thesis . University of Texas at Arlington , TX, 2000.
- R.30 Spong, M.W. y Vidyasagar, M. "Robot Dynamics and Control". Wiley (1989).
- R.31 Yang B. and Asada H. A new approach of adaptive reinforcement learning control, in *Proceedings of the International Joint Conference on Neural Networks* vol.1 627-630 , 1993.

A N E X O

ESCRITURA DE PROGRAMAS EN MATHEMATICA

Simulación 8.2

PD_2e_Sfricción_posición2

Controlador PD;2 eslabones Sin fricción;posición deseada 2

```
qd = {1, 1};  
q = {θ1[t], θ2[t]};  
e = qd - q;  
de = D[e, t];  
Λ = {{0.35, 0}, {0, 0.35}};  
r = e + Λ.de;  
K = {{900, 0}, {0, 200}};  
τ = K.r;  
robot1 =  
{8.09325 * Cos[θ1[t]] + 0.0675 * Sin[θ1[t] - θ2[t]] θ2'[t]2 + 0.35 * θ1''[t] +  
0.0675 * Cos[θ1[t] - θ2[t]] θ2''[t] == τ[[1]]}  
robot2 =  
{1.32435 * Cos[θ2[t]] - 0.0675 * Sin[θ1[t] - θ2[t]] θ1'[t]2 +  
0.0675 * Cos[θ1[t] - θ2[t]] θ1''[t] + 0.027 * θ2''[t] == τ[[2]]}
```

Solucion

```
init1 = {θ1[0] == 3 π / 2, θ1'[0] == 0};
init2 = {θ2[0] == 3 π / 2, θ2'[0] == 0};
robots = Join[robot1, robot2, init1, init2]

Tmax = 8;
sol = Flatten[NDSolve[robots, {θ1[t], θ2[t]}, {t, 0, Tmax}]]
dte = D[e /. sol, t]
dθ = D[{θ1[t], θ2[t]} /. sol, t];

Plot[Evaluate[{θ1[t], θ2[t]} /. sol], {t, 0, Tmax}, PlotRange → All,
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, PlotPoints → 350,
  AxesLabel → {t, θ}, PlotLabel → "Angulos θ1(rojo) y θ2(azul)"]

Plot[Evaluate[e /. sol], {t, 0, Tmax}, PlotRange → All,
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, PlotPoints → 350,
  AxesLabel → {t, eθ}, PlotLabel → "Errores eθ1(rojo) y eθ2(azul)"]

Show[%, PlotRange → .01 {-1, 1}]

Plot[Evaluate[dθ /. sol], {t, 0, Tmax}, PlotRange → All,
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, PlotPoints → 350,
  AxesLabel → {t, θ'}, PlotLabel → "Velocidades θ1'(rojo) y θ2'(azul)"]

Show[%, PlotRange → {{1, 3}, {-1, 0}}]

armDibuja[dt_] :=
  Block[{tracel},
    tracel = ParametricPlot[{Cos[θ1[t]], Sin[θ1[t]]} /. sol, {t, 0, dt},
      PlotRange → {{-2, 2}, {-2, 2}}, PlotPoints → 125, AspectRatio → 1,
      DisplayFunction → Identity];
    Block[{trace2},
      trace2 = ParametricPlot[{Cos[θ1[t]] + Cos[θ2[t]], Sin[θ1[t]] + Sin[θ2[t]]} /.
        sol, {t, 0, dt}, PlotRange → {{-2, 2}, {-2, 2}}, PlotPoints → 125,
        AspectRatio → 1, DisplayFunction → Identity];
      Show[tracel, trace2,
        Graphics[
          {{Thickness[.02], Line[{{0, 0}, {Cos[θ1[t]], Sin[θ1[t]}]}]},
           {Thickness[.02], Line[{{Cos[θ1[t]], Sin[θ1[t]],
             Cos[θ1[t]] + Cos[θ2[t]], Sin[θ1[t]] + Sin[θ2[t]}]}]},
           {RGBColor[1, 0, 0], PointSize[.02], Point[{0, 0}]},
           {RGBColor[1, 0, 0], PointSize[.02], Point[{Cos[θ1[t]], Sin[θ1[t]}]}]},
           {RGBColor[1, 0, 0], PointSize[.02],
            Point[{Cos[θ1[t]] + Cos[θ2[t]], Sin[θ1[t]] + Sin[θ2[t]}]} /. sol /.
            t → dt}, PlotRange → {{-2, 2}, {-2, 2}}
          ], AspectRatio → 1, DisplayFunction → $DisplayFunction]]]

For[dt = .1, dt ≤ Tmax, dt += .2, armDibuja[dt]]
```

Simulación 8.7

PD_Cfricción_tray.Seno-Coseno_Aprox.Final

Control PD 2 eslabones

Con Fricción ; Trayectoria de referencia II Seno-Coseno

Tolerancia del error a una diezmilésima de radian

```
qd = {Sin[t], Cos[t]};

q = {θ1[t], θ2[t]};

e = qd - q;

de = D[e, t];

Λ = {{0.0003, 0}, {0, 0.0003}};

r = e + Λ.de;

K = {{450000, 0}, {0, 450000}};

τ = K.r;

robot1 =
{8.09325 * Cos[θ1[t]] + 0.0675 * Sin[θ1[t] - θ2[t]] θ2'[t]^2 + 0.35 * θ1''[t] +
0.0675 * Cos[θ1[t] - θ2[t]] θ2''[t] +
(35 + 1.1 * Exp[-50 * Abs[θ1'[t]]) + 0.9 * (1 - Exp[-65 * Abs[θ1'[t]])}) *
Sign[θ1'[t]] == τ[[1]]};

robot2 =
{1.32435 * Cos[θ2[t]] - 0.0675 * Sin[θ1[t] - θ2[t]] θ1'[t]^2 +
0.0675 * Cos[θ1[t] - θ2[t]] θ1''[t] + 0.027 * θ2''[t] +
(38 + 1 * Exp[-55 * Abs[θ2'[t]]) + 0.95 * (1 - Exp[-60 * Abs[θ2'[t]])}) *
Sign[θ2'[t]] == τ[[2]]};
```

Solucion

```
init1 = {θ1[0] == 0, θ1'[0] == 0};
init2 = {θ2[0] == 1, θ2'[0] == 0};
robots = Join[robot1, robot2, init1, init2];
Tmax = 15;
sol = Flatten[NDSolve[robots, {θ1[t], θ2[t]}, {t, 0, Tmax},
  StartingStepSize → 1/10000,
  Method → {"FixedStep", Method → "ExplicitEuler"}, MaxSteps → 1000000]];
dte = D[e /. sol, t];
dθ = D[{θ1[t], θ2[t]} /. sol, t];
G1 = Plot[Evaluate[{θ1[t], θ2[t]} /. sol], {t, 0, Tmax},
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},
  GridLines → {None, {-1, 1}}, PlotLabel → "Angulos θ1(rojo) y θ2(azul)"];
G2 = Plot[{Sin[t], Cos[t]}, {t, 0, 15}]
Show[G1, G2, PlotRange → {-1, 1}]
Show[%, PlotRange → {{11, 15}, {-1, 1}}]
Plot[Evaluate[e /. sol], {t, 0, Tmax}, PlotRange → {{0, 15}, {-1×10-4, 1×10-4}},
  GridLines → {None, {-0.0001, 0.0001}},
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, PlotPoints → 350,
  PlotLabel → "error θ1(rojo) y error θ2(azul)"
Show[%, PlotRange → {{8, 15}, {-1×10-4, 1×10-4}},
  GridLines → {None, {-0.0001, 0.0001}}]
Show[%, PlotRange → {{12, 15}, {-0.0002, 0.0002}}]
Plot[Evaluate[dθ /. sol], {t, 0, Tmax}, PlotRange → All,
  GridLines → {None, {-1, 1}},
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, PlotPoints → 350,
  PlotLabel → "Velocidades θ1(rojo) y θ2(azul)"
Show[%, PlotRange → {{9, 15}, {-1, 1}}]
```

Simulación 9.2

PD+RN_Cfricción_posición_2

■ CONTROLADOR PD + RED NEURONAL Con fricción ; posición deseada 2

Dimensión de entrada ; neuronas de la capa oculta y neuronas de capa de salida

```
d = 2;
nInput = 5 d (*q, dq, qd, dqd, d2qd*);
nHidden = 6;
nOut = d;
```

■ Función de Activación

```
x1 = {θ1, θ2, dθ1, dθ2, eθ1, eθ2, dθd1, dθd2, d2θd1, d2θd2};

Extend[x_List] := Prepend[x, 1];

σ[z_] = 1 / (1 + Exp[-z]);
SetAttributes[σ, Listable]

NN[x_List, VT_List, WT_List] := WT.Extend[σ[VT.Extend[x]]]

NN[x1, V1T, W1T];

(* ordena una matriz m × n por columnas *)
J[{i_, j_}, m_, n_] := (j - 1) * m + i
```

■ Estructura de la Red Neuronal

```
x1 = {θ1, θ2, dθ1, dθ2, eθ1, eθ2, dθd1, dθd2, d2θd1, d2θd2};

Length[x1];

V1T = Table[v[J[{i, j}, nHidden, nInput + 1]], {i, 1, nHidden}, {j, 1, nInput + 1}];

W1T = Table[w[J[{i, j}, nOut, nHidden + 1]], {i, 1, nOut}, {j, 1, nHidden + 1}];
```

■ Valores de las matrices de ganancia y tasa de aprendizaje

```
T = DiagonalMatrix[Table[10, {nInput + 1}]];
S = DiagonalMatrix[Table[10, {nHidden + 1}]];
Λ = DiagonalMatrix[Table[0.6, {nOut}]];
K = DiagonalMatrix[Table[18, {nOut}]];
k = 0.01;
```

■ Error de seguimiento , su derivada y el error filtrado de seguimiento

```
e[x_List] := {Part[x, 5] - Part[x, 1], Part[x, 6] - Part[x, 2]}
de[x_List] := {Part[x, 7] - Part[x, 3], Part[x, 8] - Part[x, 4]}
rT[x_List] := de[x] + Λ.e[x];
```

■ Derivada de la función de activación y matriz derivada

$$d\sigma[z_]=\frac{\text{Exp}[-z]}{(1+\text{Exp}[-z])^2};$$

```
(*Matriz derivada de  $\sigma[z_1, z_2, \dots, z_L]=\{1, \sigma[z_1], \sigma[z_2], \dots, \sigma[z_L]\}$ *)
D $\sigma$ [z_List] := Block[{M = Table[0, {nHidden}, {nHidden}]},
  For[i = 1, i ≤ nHidden, i++, M[[i, i]] = d $\sigma$ [z[[i]]]];
  M = Prepend[M, Table[0, {nHidden}]];
  Return[M];]

norm[z_List] := Sqrt[Inner[Times, z, z]]
(* Campos vectoriales que definen el lado derecho del tunning *)
```

■ Reglas de Ajuste de Pesos de la Red Neuronal

```
dW1[x_List, VT_List, WT_List] := Outer[Times, S.Extend[ $\sigma$ [VT.Extend[x]]], rT[x]];

dV1[x_List, VT_List, WT_List] :=
  T.Outer[Times, Extend[x], rT[x]].WT.D $\sigma$ [VT.Extend[x]];

dW1[x1, V1T, W1T] // Dimensions;
dV1[x1, V1T, W1T] // Dimensions;
```

■ Reglas para sustituir variables por variables dependientes de "t" y para sustituir variables por derivadas dependientes de "t"

```
rule1 = {}; oderule1 = {};
For[i = 1, i ≤ Length[x1], i++,
  rule1 = Append[rule1, ToExpression[ToString[x1[[i]]] <> "[t]"];
  oderule1 = Append[oderule1, ToExpression[ToString[x1[[i]]] <> "" <> "[t]"]];]

(* Trayectoria deseada; en general se usa una función de t *)
```

■ Trayectoria de referencia

```
track = { $\partial_1 \rightarrow 1$ ,  $\partial_2 \rightarrow 1$ ,  $d\partial_1 \rightarrow 0$ ,  $d\partial_2 \rightarrow 0$ ,  $d^2\partial_1 \rightarrow 0$ ,  $d^2\partial_2 \rightarrow 0$ };
along1 = Thread[x1 → rule1];

(* Regla de sustitución *)
odealong1 = Thread[x1 → oderule1];
```

■ Reglas para generar la ecuación vectorial equivalente a la ecuación matricial $V1'=dV1$

```

V1TFlat = Flatten[V1T];

Length[V1TFlat];

rule2 = {}; oderule2 = {}
For[i = 1, i ≤ Length[V1TFlat], i++,
  rule2 = Append[rule2, ToExpression[ToString[V1TFlat[[i]]] <> "[t]"];
  oderule2 =
    Append[oderule2, ToExpression[ToString[V1TFlat[[i]]] <> "' <> "[t]"];];

along2 = Thread[V1TFlat → rule2];

odealong2 = Thread[V1TFlat → oderule2];

W1TFlat = Flatten[W1T];

Length[W1TFlat];

rule3 = {}; oderule3 = {};
For[i = 1, i ≤ Length[W1TFlat], i++,
  rule3 = Append[rule3, ToExpression[ToString[W1TFlat[[i]]] <> "[t]"];
  oderule3 = Append[oderule3,
    ToExpression[ToString[W1TFlat[[i]]] <> "' <> "[t]"];];

along3 = Thread[W1TFlat → rule3];

odealong3 = Thread[W1TFlat → oderule3];

along = Join[along1, along2, along3];

odealong = Join[odealong1, odealong2, odealong3];

```

■ Generar las ecuaciones vectoriales equivalentes a las ecuaciones matriciales $W1'=dW1$ y $V1'=dV1$

```

ode2 = Thread[(Flatten[Transpose[W1T]] /. odealong) ==
  (Flatten[dW1[x1, V1T, W1T] /. track] /. along)];

ode3 = Thread[(Flatten[Transpose[V1T]] /. odealong) ==
  (Flatten[dV1[x1, V1T, W1T] /. track] /. along)];

force =
  {(0.027 τ1 - 0.21851774999999998 Cos[θ1] - 0.0675 τ2 Cos[θ1 - 1. θ2] +
    0.08939362499999999 Cos[θ1 - 1. θ2] Cos[θ2] -
    0.00455625 Cos[θ1 - 1. θ2] Sin[θ1 - 1. θ2] dθ12 -
    0.0018225000000000001 Sin[θ1 - 1. θ2] dθ22) /
    (0.00945 - 0.00455625 Cos[θ1 - 1. θ2]2) +
    (35 + 1.1 * Exp[-50 * Abs[dθ1]] + 0.9 * (1 - Exp[-65 * Abs[dθ1]])) * Sign[dθ1],
  - (1. (-0.35 τ2 + 0.46352249999999995 Cos[θ2] - 0.023625 Sin[θ1 - 1. θ2] dθ12 -
    0.0675 Cos[θ1 - 1. θ2]
    (-1. τ1 + 8.09325 Cos[θ1] + 0.0675 Sin[θ1 - 1. θ2] dθ22)) /
    (0.00945 - 0.00455625 Cos[θ1 - 1. θ2]2) +
    (38 + 1 * Exp[-55 * Abs[dθ2]] + 0.95 * (1 - Exp[-60 * Abs[dθ2]])) * Sign[dθ2]};

```

■ Control PD

```
τ0 = K.rT[x1];
```

■ Control Neuronal

```
 $\tau = \text{NN}[x1, V1T, W1T] + \tau0 /. \text{track};$   
(*  $\tau = \tau0 /. \text{track};$  *)
```

■ Retroalimentación

```
 $\tau1 = \tau[[1]];$   
 $\tau2 = \tau[[2]];$   
  
ForceAlong = force /. along;  
  
ode1 = { $\theta1'[t] == d\theta1[t]$ ,  $\theta2'[t] == d\theta2[t]$ ,  $d\theta1'[t] == \text{ForceAlong}[[1]]$ ,  
   $d\theta2'[t] == \text{ForceAlong}[[2]]$ };  
  
ode = Join[ode1, ode2, ode3];  
  
odeA = Join[ode2, ode3];  
  
L = Length[odeA];  
  
vars = Join[{ $\theta1[t]$ ,  $\theta2[t]$ ,  $d\theta1[t]$ ,  $d\theta2[t]$ },  
  Table[v[i][t], {i, 1, Length[Flatten[V1T]]}],  
  Table[w[i][t], {i, 1, Length[Flatten[W1T]]}]];  
  
varsini = Join[{ $\theta1[0]$ ,  $\theta2[0]$ ,  $d\theta1[0]$ ,  $d\theta2[0]$ },  
  Table[v[i][0], {i, 1, Length[Flatten[V1T]]}],  
  Table[w[i][0], {i, 1, Length[Flatten[W1T]]}]];  
  
odeini =  
  Thread[varsini == Join[{ $3\pi/2$ ,  $3\pi/2$ , 0, 0}, Table[Random[Real, {0, 0}], {L}]]];  
  
Tmax = 15;  
  
sol = NDSolve[Join[ode, odeini], vars, {t, 0, Tmax}, StartingStepSize  $\rightarrow 1/10000$ ,  
  Method  $\rightarrow$  {"FixedStep", Method  $\rightarrow$  "ExplicitRungeKutta"}, MaxSteps  $\rightarrow 1000000$ ];  
  
Plot[Evaluate[{ $\theta1[t]$ ,  $\theta2[t]$ } /. sol], {t, 0, Tmax},  
  PlotStyle  $\rightarrow$  {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, GridLines  $\rightarrow$  {None, {0, 1}},  
  PlotRange  $\rightarrow$  {{0, 15}, {0, 5}}, PlotLabel  $\rightarrow$  "Angulos Rojo  $\theta1$  y Azul  $\theta2$ ";  
  
Plot[Evaluate[{ $\theta1[t] - 1$ ,  $\theta2[t] - 1$ } /. sol], {t, 0, Tmax},  
  PlotStyle  $\rightarrow$  {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},  
  PlotRange  $\rightarrow$  {{0, 15}, {-1, 5}},  
  PlotLabel  $\rightarrow$  "Error Rojo  $\theta1$  y Error Azul  $\theta2$ ";  
  
Show[%, PlotRange  $\rightarrow$  {{11, 15}, {-1  $\times 10^{-3}$ , 1  $\times 10^{-3}$ }}]  
  
Plot[Evaluate[{ $d\theta1[t]$ ,  $d\theta2[t]$ } /. sol], {t, 0, Tmax},  
  PlotStyle  $\rightarrow$  {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},  
  GridLines  $\rightarrow$  {None, {0, 2}}, PlotRange  $\rightarrow$  {{0, 15}, {-5, 1}},  
  PlotLabel  $\rightarrow$  "Velocidad de  $\theta1$  Rojo; Velocidad de  $\theta2$  Azul ";  
  
Show[%, PlotRange  $\rightarrow$  {{0, 15}, {-1, 1}}]
```

Simulación 9.5

PD+RN_Cfricción_SC_Aprox._1

■ CONTROLADOR PD + RED NEURONAL

Con fricción ; seguimiento de trayectoria Seno-Coseno
para Comparativo con RNA

Dimensión de entrada ; neuronas de la capa oculta y neuronas de capa de salida

```
d = 2;  
nInput = 5 d (*q, dq, qd, dqd, d2qd*);  
nHidden = 10;  
nOut = d;
```

■ Función de Activación

```
x1 = {θ1, θ2, dθ1, dθ2, eθ1, eθ2, dθd1, dθd2, d2θd1, d2θd2};  
  
Extend[x_List] := Prepend[x, 1];  
  
σ[z_] = 1 / (1 + Exp[-z]);  
SetAttributes[σ, Listable]  
  
NN[x_List, VT_List, WT_List] := WT.Extend[σ[VT.Extend[x]]]  
  
NN[x1, V1T, W1T];  
  
(* ordena una matriz m x n por columnas *)  
J[{i_, j_}, m_, n_] := (j - 1) * m + i
```

■ Estructura de la Red Neuronal

```
x1 = {θ1, θ2, dθ1, dθ2, eθ1, eθ2, dθd1, dθd2, d2θd1, d2θd2};  
  
Length[x1];  
  
V1T = Table[v[J[{i, j}, nHidden, nInput + 1]], {i, 1, nHidden}, {j, 1, nInput + 1}];  
  
W1T = Table[w[J[{i, j}, nOut, nHidden + 1]], {i, 1, nOut}, {j, 1, nHidden + 1}];
```

■ Valores de las matrices de ganancia y tasa de aprendizaje

```
T = DiagonalMatrix[Table[30, {nInput + 1}]];  
S = DiagonalMatrix[Table[30, {nHidden + 1}]];  
Λ = DiagonalMatrix[Table[5, {nOut}]];  
K = DiagonalMatrix[Table[30, {nOut}]];  
k = 0.05;
```

■ Error de seguimiento , su derivada y el error filtrado de seguimiento

```
e[x_List] := {Part[x, 5] - Part[x, 1], Part[x, 6] - Part[x, 2]}  
de[x_List] := {Part[x, 7] - Part[x, 3], Part[x, 8] - Part[x, 4]}  
rT[x_List] := de[x] + Λ.e[x];
```

■ Derivada de la función de activación y matriz derivada

$$d\sigma[z_]=\frac{\text{Exp}[-z]}{(1+\text{Exp}[-z])^2};$$

```
(*Matriz derivada de  $\sigma[z_1, z_2, \dots, z_L]=\{1, \sigma[z_1], \sigma[z_2], \dots, \sigma[z_L]\}$ *)
D $\sigma$ [z_List] := Block[{M = Table[0, {nHidden}, {nHidden}]},
  For[i = 1, i ≤ nHidden, i++, M[[i, i]] = d $\sigma$ [z[[i]]]];
  M = Prepend[M, Table[0, {nHidden}]];
  Return[M];]

norm[z_List] := Sqrt[Inner[Times, z, z]]
(* Campos vectoriales que definen el lado derecho del tunning *)
```

■ Reglas de Ajuste de Pesos de la Red Neuronal

```
dW1[x_List, VT_List, WT_List] :=
  Outer[Times, S.Extend[ $\sigma$ [VT.Extend[x]]], rT[x]] -
  (Outer[Times, S.D $\sigma$ [VT.Extend[x]].VT.Extend[x], rT[x]]) -
  (k norm[rT[x]] S.Transpose[WT]);

dV1[x_List, VT_List, WT_List] :=
  T.Outer[Times, Extend[x], rT[x]].WT.D $\sigma$ [VT.Extend[x]] -
  (k norm[rT[x]] T.Transpose[VT]);

dW1[x1, v1T, w1T] // Dimensions;

dV1[x1, v1T, w1T] // Dimensions;
```

■ Reglas para sustituir variables por variables dependientes de "t" y para sustituir variables por derivadas dependientes de "t"

```
rule1 = {}; oderule1 = {};
For[i = 1, i ≤ Length[x1], i++,
  rule1 = Append[rule1, ToExpression[ToString[x1[[i]]] <> "[t]"];
  oderule1 = Append[oderule1, ToExpression[ToString[x1[[i]]] <> "" <> "[t]"]];]

(* Trayectoria deseada; en general se usa una función de t *)
```

■ Trayectoria de referencia

```
track = { $\theta$ d1 → Sin[t],  $\theta$ d2 → Cos[t], d $\theta$ d1 → 0, d $\theta$ d2 → 0, d2 $\theta$ d1 → 0, d2 $\theta$ d2 → 0};
along1 = Thread[x1 → rule1];

(* Regla de sustitución *)

odealong1 = Thread[x1 → oderule1];
```

■ Reglas para generar la ecuación vectorial equivalente a la ecuación matricial $V1'=dV1$

```
V1TFlat = Flatten[V1T];
Length[V1TFlat];
rule2 = {}; oderule2 = {}
For[i = 1, i ≤ Length[V1TFlat], i++,
  rule2 = Append[rule2, ToExpression[ToString[V1TFlat[[i]]] <> "[t]"];
  oderule2 =
  Append[oderule2, ToExpression[ToString[V1TFlat[[i]]] <> "' <> "[t]"];];
{}
along2 = Thread[V1TFlat → rule2];
odealong2 = Thread[V1TFlat → oderule2];
W1TFlat = Flatten[W1T];
Length[W1TFlat];
rule3 = {}; oderule3 = {};
For[i = 1, i ≤ Length[W1TFlat], i++,
  rule3 = Append[rule3, ToExpression[ToString[W1TFlat[[i]]] <> "[t]"];
  oderule3 = Append[oderule3,
  ToExpression[ToString[W1TFlat[[i]]] <> "' <> "[t]"];];
along3 = Thread[W1TFlat → rule3];
odealong3 = Thread[W1TFlat → oderule3];
along = Join[along1, along2, along3];
odealong = Join[odealong1, odealong2, odealong3];
```

■ Generar las ecuaciones vectoriales equivalentes a las ecuaciones matriciales $W1'=dW1$ y $V1'=dV1$

```
ode2 = Thread[(Flatten[Transpose[W1T]] /. odealong) ==
  (Flatten[dW1[x1, V1T, W1T] /. track] /. along)];
ode3 = Thread[(Flatten[Transpose[V1T]] /. odealong) ==
  (Flatten[dV1[x1, V1T, W1T] /. track] /. along)];
force =
{(0.027 τ1 - 0.21851774999999998 Cos[θ1] - 0.0675 τ2 Cos[θ1 - 1. θ2] +
  0.08939362499999999 Cos[θ1 - 1. θ2] Cos[θ2] -
  0.00455625 Cos[θ1 - 1. θ2] Sin[θ1 - 1. θ2] dθ12 -
  0.0018225000000000001 Sin[θ1 - 1. θ2] dθ22) /
(0.00945 - 0.00455625 Cos[θ1 - 1. θ2]2) +
(35 + 1.1 * Exp[-50 * Abs[dθ1]] + 0.9 * (1 - Exp[-65 * Abs[dθ1]])) * Sign[dθ1],
- (1. (-0.35 τ2 + 0.46352249999999995 Cos[θ2] - 0.023625 Sin[θ1 - 1. θ2] dθ12 -
  0.0675 Cos[θ1 - 1. θ2]
  (-1. τ1 + 8.09325 Cos[θ1] + 0.0675 Sin[θ1 - 1. θ2] dθ22)) /
(0.00945 - 0.00455625 Cos[θ1 - 1. θ2]2) +
(38 + 1 * Exp[-55 * Abs[dθ2]] + 0.95 * (1 - Exp[-60 * Abs[dθ2]])) * Sign[dθ2]};
```

■ Control PD

```
τ0 = K.rT[x1];
```

■ Control Neuronal

```
 $\tau = \text{NN}[x1, v1T, w1T] + \tau0 /. \text{track};$   
(*  $\tau = \tau0 /. \text{track};$  *)
```

■ Retroalimentación

```
 $\tau1 = \tau[[1]];$   
 $\tau2 = \tau[[2]];$   
ForceAlong = force /. along;  
ode1 = { $\theta1'[t] == d\theta1[t]$ ,  $\theta2'[t] == d\theta2[t]$ ,  $d\theta1'[t] == \text{ForceAlong}[[1]]$ ,  
   $d\theta2'[t] == \text{ForceAlong}[[2]]$ };  
ode = Join[ode1, ode2, ode3];  
odeA = Join[ode2, ode3];  
L = Length[odeA];  
vars = Join[{ $\theta1[t]$ ,  $\theta2[t]$ ,  $d\theta1[t]$ ,  $d\theta2[t]$ },  
  Table[v[i][t], {i, 1, Length[Flatten[v1T]]}],  
  Table[w[i][t], {i, 1, Length[Flatten[w1T]]}]];  
varsini = Join[{ $\theta1[0]$ ,  $\theta2[0]$ ,  $d\theta1[0]$ ,  $d\theta2[0]$ },  
  Table[v[i][0], {i, 1, Length[Flatten[v1T]]}],  
  Table[w[i][0], {i, 1, Length[Flatten[w1T]]}]];  
odeini =  
  Thread[varsini == Join[{0, 1, 0, 0}, Table[Random[Real, {0.1, 0.1}], {L}]]];  
Tmax = 10;  
sol = NDSolve[Join[ode, odeini], vars, {t, 0, Tmax}, StartingStepSize → 1/10000,  
  Method → {"FixedStep", Method → "ExplicitRungeKutta"}, MaxSteps → 1000000];  
G1 = Plot[Evaluate[{ $\theta1[t]$ ,  $\theta2[t]$ } /. sol], {t, 0, Tmax},  
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]}, GridLines → {None, {-1, 1}},  
  PlotRange → {{0, 10}, {-1.2, 1.2}}, PlotLabel → "Angulos Rojo  $\theta1$  y Azul  $\theta2$ ";  
G2 = Plot[{Sin[t], Cos[t]}, {t, 0, 10}]  
Show[G1, G2, PlotRange → {{0, 10}, {-1.2, 1.2}}]  
Show[G1, G2, PlotRange → {{4.5, 10}, {-1.2, 1.2}}]  
Plot[Evaluate[{ $\theta1[t] - \text{Sin}[t]$ ,  $\theta2[t] - \text{Cos}[t]$ } /. sol], {t, 0, Tmax},  
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},  
  PlotRange → {{0, 10}, {-0.2, 0.2}}, GridLines → {None, {-0.2, 0.2}},  
  PlotLabel → "Error Rojo  $\theta1$  y Error Azul  $\theta2$ ";  
Show[%, PlotRange → {{2.5, 7.5}, {-0.2, 0.2}}]  
Plot[Evaluate[{ $d\theta1[t]$ ,  $d\theta2[t]$ } /. sol], {t, 0, Tmax},  
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},  
  PlotRange → {{0, 10}, {-1, 1}},  
  PlotLabel → "Velocidad de  $\theta1$  Rojo; Velocidad de  $\theta2$  Azul ";  
Show[%, PlotRange → {{0, 2}, {-1.2, 1}}]
```

Simulación 10.3.

PD+RNA_Cfricción_traySC_Aprox.Final

- **CONTROLADOR PD + RED NEURONAL + RED AUMENTADA**
Con fricción ; Trayectoria de referencia seno-coseno
8 neuronas adicionales 4 con función de activación positiva y 4 con negativa

Dimensión de entrada ; neuronas de la capa oculta y neuronas de capa de salida

```
nInput = 10;
nHidden = 10;
nHiddenA1 = 1;
nHiddenA2 = 2;
nHiddenA3 = 1;
nHiddenB1 = 1;
nHiddenB2 = 2;
nHiddenB3 = 1;
nOut = 2;
```

- **Función de Activación**

```
x1 = {θ1, θ2, dθ1, dθ2, eθ1, eθ2, dθd1, dθd2, d2θd1, d2θd2};

Extend[x_List] := Prepend[x, 1];

σ[z_] = 1 / (1 + Exp[-z])
SetAttributes[σ, Listable];

φ1[z_] = Piecewise[{{0, z < 0}, {(1 - e-z) / (1 + e-z)0, z ≥ 0}}]
SetAttributes[φ1, Listable];

φ2[z_] = Piecewise[{{0, z < 0}, {(1 - e-z) / (1 + e-z)1, z ≥ 0}}]
SetAttributes[φ2, Listable];

φ3[z_] = Piecewise[{{0, z < 0}, {(1 - e-z) / (1 + e-z)2, z ≥ 0}}]
SetAttributes[φ3, Listable];

ω1[z_] = Piecewise[{{0, z < 0}, {-((1 - e-z) / (1 + e-z)0, z ≥ 0}}]
SetAttributes[ω1, Listable];

ω2[z_] = Piecewise[{{0, z < 0}, {-((1 - e-z) / (1 + e-z)1, z ≥ 0}}]
SetAttributes[ω2, Listable];

ω3[z_] = Piecewise[{{0, z < 0}, {-((1 - e-z) / (1 + e-z)2, z ≥ 0}}]
SetAttributes[ω3, Listable];

NN[x_List, VT_List, WT_List, AT_List, BT_List, CT_List, DT_List, ET_List,
  FT_List, GT_List, HT_List, LT_List, MT_List, NT_List, OT_List,
  PT_List, QT_List, RT_List, UT_List,] :=
  WT.Extend[σ[VT.Extend[x]]] + BT.φ1[AT.Extend[x]] + DT.φ2[CT.Extend[x]] +
  FT.φ3[ET.Extend[x]] + MT.ω1[LT.Extend[x]] + OT.ω2[NT.Extend[x]] +
  QT.ω3[RT.Extend[x]];

NN[x1, V1T, W1T, A1T, B1T, C1T, D1T, E1T, F1T, L1T, M1T, N1T, O1T, R1T, Q1T];

(* ordena una matriz m × n por columnas *)
J[{i_, j_}, m_, n_] := (j - 1) * m + i;
```

■ Estructura de la Red Neuronal

```
Length[x1];

V1T = Table[v[J[{i, j}, nHidden, nInput + 1]], {i, 1, nHidden}, {j, 1, nInput + 1}];
V1T // Dimensions;

A1T = Table[J[{1, 1}, nHiddenA1, nInput + 1], {i, 1, nHiddenA1}, {j, 1, nInput + 1}];
A1T // Dimensions;

C1T = Table[J[{1, 1}, nHiddenA2, nInput + 1], {i, 1, nHiddenA2}, {j, 1, nInput + 1}];
C1T // Dimensions;

E1T = Table[J[{1, 1}, nHiddenA3, nInput + 1], {i, 1, nHiddenA3}, {j, 1, nInput + 1}];
E1T // Dimensions;

L1T = Table[J[{1, 1}, nHiddenB1, nInput + 1], {i, 1, nHiddenB1}, {j, 1, nInput + 1}];
L1T // Dimensions;

N1T = Table[J[{1, 1}, nHiddenB2, nInput + 1], {i, 1, nHiddenB2}, {j, 1, nInput + 1}];
N1T // Dimensions;

R1T = Table[J[{1, 1}, nHiddenB3, nInput + 1], {i, 1, nHiddenB3}, {j, 1, nInput + 1}];
R1T // Dimensions;

W1T = Table[w[J[{i, j}, nOut, nHidden + 1]], {i, 1, nOut}, {j, 1, nHidden + 1}];
W1T // Dimensions;

B1T = Table[w1[J[{i, j}, nOut, nHiddenA1]], {i, 1, nOut}, {j, 1, nHiddenA1}];
B1T // Dimensions;

D1T = Table[w2[J[{i, j}, nOut, nHiddenA2]], {i, 1, nOut}, {j, 1, nHiddenA2}];
D1T // Dimensions;

F1T = Table[w3[J[{i, j}, nOut, nHiddenA3]], {i, 1, nOut}, {j, 1, nHiddenA3}];
F1T // Dimensions;

M1T = Table[w5[J[{i, j}, nOut, nHiddenB1]], {i, 1, nOut}, {j, 1, nHiddenB1}];
M1T // Dimensions;

O1T = Table[w6[J[{i, j}, nOut, nHiddenB2]], {i, 1, nOut}, {j, 1, nHiddenB2}];
O1T // Dimensions;

Q1T = Table[w7[J[{i, j}, nOut, nHiddenB3]], {i, 1, nOut}, {j, 1, nHiddenB3}];
Q1T // Dimensions;
```

■ Valores de las matrices de ganancia y tasa de aprendizaje

```
T = DiagonalMatrix[Table[30, {nInput + 1}]];
S = DiagonalMatrix[Table[30, {nHidden + 1}]];
α = DiagonalMatrix[Table[35, {nHiddenA1}]];
β = DiagonalMatrix[Table[35, {nHiddenA2}]];
γ = DiagonalMatrix[Table[35, {nHiddenA3}]];
η = DiagonalMatrix[Table[35, {nHiddenB1}]];
λ = DiagonalMatrix[Table[35, {nHiddenB2}]];
μ = DiagonalMatrix[Table[35, {nHiddenB3}]];
Λ = DiagonalMatrix[Table[10000, {nOut}]];
K = DiagonalMatrix[Table[380, {nOut}]];
k = 0.05;
```

■ Error de seguimiento, su derivada y el error filtrado de seguimiento

```
e[x_List] := {Part[x, 5] - Part[x, 1], Part[x, 6] - Part[x, 2]}
de[x_List] := {Part[x, 7] - Part[x, 3], Part[x, 8] - Part[x, 4]}
rT[x_List] := de[x] + Λ.e[x];
```

■ Derivada de la función de activación y matriz derivada

$$d\sigma[z] = \frac{\text{Exp}[-z]}{(1 + \text{Exp}[-z])^2};$$

```
(*Matriz derivada de  $\sigma[z_1, z_2, \dots, z_L] = \{1, \sigma[z_1], \sigma[z_2], \dots, \sigma[z_L]\}$ *)
Dσ[z_List] := Block[{M = Table[0, {nHidden}, {nHidden}]},
  For[i = 1, i ≤ nHidden, i++, M[[i, i]] = dσ[z[[i]]]];
  M = Prepend[M, Table[0, {nHidden}]];
  Return[M];];

norm[z_List] := Sqrt[Inner[Times, z, z]];
(* Campos vectoriales que definen el lado derecho del tunning *)
```

■ Reglas de Ajuste de Pesos de la Red Neuronal

```
dW1[x_List, VT_List, WT_List] :=
  Outer[Times, S.Extend[σ[VT.Extend[x]]], rT[x]] -
  (Outer[Times, S.Dσ[VT.Extend[x]].VT.Extend[x], rT[x]] -
  (k norm[rT[x]] S.Transpose[WT]))

dW1[x1, v1T, w1T] // Dimensions;

dV1[x_List, VT_List, WT_List] :=
  T.Outer[Times, Extend[x], rT[x]].WT.Dσ[VT.Extend[x]] -
  (k norm[rT[x]] T.Transpose[VT])

dV1[x1, v1T, w1T] // Dimensions;

dB1[x_List, AT_List, BT_List] :=
  Outer[Times, α.φ1[AT.Extend[x]], rT[x]] - (k norm[rT[x]] α.Transpose[BT])

dB1[x1, a1T, b1T] // Dimensions;

dD1[x_List, CT_List, DT_List] :=
  Outer[Times, β.φ2[CT.Extend[x]], rT[x]] - (k norm[rT[x]] β.Transpose[DT])

dD1[x1, c1T, d1T] // Dimensions;

dF1[x_List, ET_List, FT_List] :=
  Outer[Times, γ.φ3[ET.Extend[x]], rT[x]] - (k norm[rT[x]] γ.Transpose[FT])

dF1[x1, e1T, f1T] // Dimensions;

dM1[x_List, LT_List, MT_List] :=
  Outer[Times, η.ω1[LT.Extend[x]], rT[x]] - (k norm[rT[x]] η.Transpose[MT])

dM1[x1, l1T, m1T] // Dimensions;

dO1[x_List, NT_List, OT_List] :=
  Outer[Times, λ.ω2[NT.Extend[x]], rT[x]] - (k norm[rT[x]] λ.Transpose[OT])

dO1[x1, n1T, o1T] // Dimensions;

dQ1[x_List, RT_List, QT_List] :=
  Outer[Times, μ.ω3[RT.Extend[x]], rT[x]] - (k norm[rT[x]] μ.Transpose[QT]);

dQ1[x1, r1T, q1T] // Dimensions;
```

- Reglas para sustituir variables por variables dependientes de "t" y para sustituir variables por derivadas dependientes de "t"

```
rule1 = {}; oderule1 = {};
For[i = 1, i < Length[x1], i++,
  rule1 = Append[rule1, ToExpression[ToString[x1[[i]]] <> "[t]"]];
  oderule1 = Append[oderule1, ToExpression[ToString[x1[[i]]] <> "' ' <> "[t]"]];]

(* Trayectoria deseada; en general se usa una función de t *)
```

- Trayectoria de referencia

```
track = {θd1 → Sin[t], θd2 → Cos[t], dθd1 → 0, dθd2 → 0, d2θd1 → 0, d2θd2 → 0};
along1 = Thread[x1 → rule1];

(* Regla de sustitución *)

odealong1 = Thread[x1 → oderule1];
```

- Reglas para generar la ecuación vectorial equivalente a la ecuación matricial $V1'=dV1$

```
V1TFlat = Flatten[V1T];
A1TFlat = Flatten[A1T];
C1TFlat = Flatten[C1T];
E1TFlat = Flatten[E1T];
L1TFlat = Flatten[L1T];
N1TFlat = Flatten[N1T];
R1TFlat = Flatten[R1T];

Length[V1TFlat];
Length[A1TFlat];
Length[C1TFlat];
Length[E1TFlat];
Length[L1TFlat];
Length[N1TFlat];
Length[R1TFlat];
```

```

rule2 = {}; oderule2 = {}
For[i = 1, i <= Length[V1TFlat], i++,
  rule2 = Append[rule2, ToExpression[ToString[V1TFlat[[i]]] <> "[t]"]];
  oderule2 =
    Append[oderule2, ToExpression[ToString[V1TFlat[[i]]] <> "' ' <> "[t]"]];];

along2 = Thread[V1TFlat -> rule2];

odealong2 = Thread[V1TFlat -> oderule2];

W1TFlat = Flatten[W1T];

Length[W1TFlat];

rule3 = {}; oderule3 = {};
For[i = 1, i <= Length[W1TFlat], i++,
  rule3 = Append[rule3, ToExpression[ToString[W1TFlat[[i]]] <> "[t]"]];
  oderule3 = Append[oderule3,
    ToExpression[ToString[W1TFlat[[i]]] <> "' ' <> "[t]"]];];

along3 = Thread[W1TFlat -> rule3];

odealong3 = Thread[W1TFlat -> oderule3];

B1TFlat = Flatten[B1T];

Length[B1TFlat];

rule3A = {}; oderule3A = {}
For[i = 1, i <= Length[B1TFlat], i++,
  rule3A = Append[rule3A, ToExpression[ToString[B1TFlat[[i]]] <> "[t]"]];
  oderule3A =
    Append[oderule3A, ToExpression[ToString[B1TFlat[[i]]] <> "' ' <> "[t]"]];];

along3A = Thread[B1TFlat -> rule3A];

odealong3A = Thread[B1TFlat -> oderule3A];

D1TFlat = Flatten[D1T];

Length[D1TFlat];

rule3B = {}; oderule3B = {};
For[i = 1, i <= Length[D1TFlat], i++,
  rule3B = Append[rule3B, ToExpression[ToString[D1TFlat[[i]]] <> "[t]"]];
  oderule3B = Append[oderule3B,
    ToExpression[ToString[D1TFlat[[i]]] <> "' ' <> "[t]"]];];

along3B = Thread[D1TFlat -> rule3B];

odealong3B = Thread[D1TFlat -> oderule3B];

F1TFlat = Flatten[F1T];

Length[F1TFlat];

```

```

rule3C = {}; oderule3C = {};
For[i = 1, i ≤ Length[F1TFlat], i++,
  rule3C = Append[rule3C, ToExpression[ToString[F1TFlat[[i]]] <> "[t]"];
  oderule3C = Append[oderule3C,
    ToExpression[ToString[F1TFlat[[i]]] <> "" <> "[t]"];];
along3C = Thread[F1TFlat → rule3C];
odealong3C = Thread[F1TFlat → oderule3C];
M1TFlat = Flatten[M1T];
Length[M1TFlat];
rule3E = {}; oderule3E = {};
For[i = 1, i ≤ Length[M1TFlat], i++,
  rule3E = Append[rule3E, ToExpression[ToString[M1TFlat[[i]]] <> "[t]"];
  oderule3E = Append[oderule3E,
    ToExpression[ToString[M1TFlat[[i]]] <> "" <> "[t]"];];
along3E = Thread[M1TFlat → rule3E];
odealong3E = Thread[M1TFlat → oderule3E];
O1TFlat = Flatten[O1T];
Length[O1TFlat];
rule3F = {}; oderule3F = {};
For[i = 1, i ≤ Length[O1TFlat], i++,
  rule3F = Append[rule3F, ToExpression[ToString[O1TFlat[[i]]] <> "[t]"];
  oderule3F = Append[oderule3F,
    ToExpression[ToString[O1TFlat[[i]]] <> "" <> "[t]"];];
along3F = Thread[O1TFlat → rule3F];
odealong3F = Thread[O1TFlat → oderule3F];
Q1TFlat = Flatten[Q1T];
Length[Q1TFlat];
rule3G = {}; oderule3G = {};
For[i = 1, i ≤ Length[Q1TFlat], i++,
  rule3G = Append[rule3G, ToExpression[ToString[Q1TFlat[[i]]] <> "[t]"];
  oderule3G = Append[oderule3G,
    ToExpression[ToString[Q1TFlat[[i]]] <> "" <> "[t]"];];
along3G = Thread[Q1TFlat → rule3G];
odealong3G = Thread[Q1TFlat → oderule3G];
along = Join[along1, along2, along3, along3A, along3B, along3C, along3E,
  along3F, along3G];
Length[along];
odealong = Join[odealong1, odealong2, odealong3, odealong3A, odealong3B,
  odealong3C, odealong3E, odealong3F, odealong3G];
Length[odealong];

```

■ Generar las ecuaciones vectoriales equivalentes a las ecuaciones matriciales

$W1'=dW1$ y $V1'=dV1$

```

ode2 = Thread[(Flatten[Transpose[W1T]] /. odealong) ==
  (Flatten[dW1[x1, V1T, W1T] /. track] /. along)];

ode3 = Thread[(Flatten[Transpose[V1T]] /. odealong) ==
  (Flatten[dV1[x1, V1T, W1T] /. track] /. along)];

ode3A = Thread[(Flatten[Transpose[B1T]] /. odealong) ==
  (Flatten[dB1[x1, A1T, B1T] /. track] /. along)];

Length[ode3A];

ode3B = Thread[(Flatten[Transpose[D1T]] /. odealong) ==
  (Flatten[dD1[x1, C1T, D1T] /. track] /. along)];

Length[ode3B];

ode3C = Thread[(Flatten[Transpose[F1T]] /. odealong) ==
  (Flatten[dF1[x1, E1T, F1T] /. track] /. along)];

Length[ode3C];

ode3E = Thread[(Flatten[Transpose[M1T]] /. odealong) ==
  (Flatten[dM1[x1, L1T, M1T] /. track] /. along)];

Length[ode3E];

ode3F = Thread[(Flatten[Transpose[O1T]] /. odealong) ==
  (Flatten[dO1[x1, N1T, O1T] /. track] /. along)];

Length[ode3F];

ode3G = Thread[(Flatten[Transpose[Q1T]] /. odealong) ==
  (Flatten[dQ1[x1, R1T, Q1T] /. track] /. along)];

Length[ode3G];

force =
  {(0.027  $\tau_1$  - 0.21851774999999998 Cos[ $\theta_1$ ] - 0.0675  $\tau_2$  Cos[ $\theta_1 - 1. \theta_2$ ] +
    0.08939362499999999 Cos[ $\theta_1 - 1. \theta_2$ ] Cos[ $\theta_2$ ] -
    0.00455625 Cos[ $\theta_1 - 1. \theta_2$ ] Sin[ $\theta_1 - 1. \theta_2$ ]  $d\theta_1^2$  -
    0.0018225000000000001 Sin[ $\theta_1 - 1. \theta_2$ ]  $d\theta_2^2$ ) /
    (0.00945 - 0.00455625 Cos[ $\theta_1 - 1. \theta_2$ ]2) +
    (35 + 1.1 * Exp[-50 * Abs[d $\theta_1$ ]] + 0.9 * (1 - Exp[-65 * Abs[d $\theta_1$ ]])) * Sign[d $\theta_1$ ],
  - (1. (-0.35  $\tau_2$  + 0.46352249999999995 Cos[ $\theta_2$ ] - 0.023625 Sin[ $\theta_1 - 1. \theta_2$ ]  $d\theta_1^2$  -
    0.0675 Cos[ $\theta_1 - 1. \theta_2$ ]
    (-1.  $\tau_1$  + 8.09325 Cos[ $\theta_1$ ] + 0.0675 Sin[ $\theta_1 - 1. \theta_2$ ]  $d\theta_2^2$ )) /
    (0.00945 - 0.00455625 Cos[ $\theta_1 - 1. \theta_2$ ]2) +
    (38 + 1 * Exp[-55 * Abs[d $\theta_2$ ]] + 0.95 * (1 - Exp[-60 * Abs[d $\theta_2$ ]])) * Sign[d $\theta_2$ ];

```

■ Control PD

$\tau_0 = K.rT[x1];$

■ Control Neuronal

V1T;

x1;

W1T;

A1T;

B1T;

C1T;

D1T;

E1T;

F1T;

L1T;

M1T;

N1T;

O1T;

R1T;

Q1T;

```

NN[x_List, VT_List, WT_List, AT_List, BT_List, CT_List, DT_List, ET_List,
  FT_List, LT_List, MT_List, NT_List, OT_List, RT_List, QT_List] :=
  WT.Extend[σ[VT.Extend[x]]] + BT.φ1[AT.Extend[x]] + DT.φ2[CT.Extend[x]] +
  FT.φ3[ET.Extend[x]] + MT.ω1[LT.Extend[x]] + OT.ω2[NT.Extend[x]] +
  QT.ω3[RT.Extend[x]];

τ = NN[x1, v1T, w1T, A1T, B1T, C1T, D1T, E1T, F1T, L1T, M1T, N1T, O1T, R1T, Q1T] +
  τ0 /. track;
(* τ=τ0/.track; *)

τ // Dimensions;

```

■ Retroalimentación

```

τ1 = τ[[1]];
τ2 = τ[[2]];

ForceAlong = force /. along;

ode1 = {θ1'[t] == dθ1[t], θ2'[t] == dθ2[t], dθ1'[t] == ForceAlong[[1]],
  dθ2'[t] == ForceAlong[[2]]};

ode = Join[ode1, ode2, ode3, ode3A, ode3B, ode3C, ode3E, ode3F, ode3G];
Length[ode];

odeA = Join[ode2, ode3, ode3A, ode3B, ode3C, ode3E, ode3F, ode3G];
L = Length[odeA];

vars = Join[{θ1[t], θ2[t], dθ1[t], dθ2[t]},
  Table[v[i][t], {i, 1, Length[Flatten[v1T]}]],
  Table[w[i][t], {i, 1, Length[Flatten[w1T]}]],
  Table[w1[i][t], {i, 1, Length[Flatten[B1T]}]],
  Table[w2[i][t], {i, 1, Length[Flatten[D1T]}]],
  Table[w3[i][t], {i, 1, Length[Flatten[F1T]}]],
  Table[w5[i][t], {i, 1, Length[Flatten[M1T]}]],
  Table[w6[i][t], {i, 1, Length[Flatten[O1T]}]],
  Table[w7[i][t], {i, 1, Length[Flatten[Q1T]}]]];

Length[vars];

varsini = Join[{θ1[0], θ2[0], dθ1[0], dθ2[0]},
  Table[v[i][0], {i, 1, Length[Flatten[v1T]}]],
  Table[w[i][0], {i, 1, Length[Flatten[w1T]}]],
  Table[w1[i][0], {i, 1, Length[Flatten[B1T]}]],
  Table[w2[i][0], {i, 1, Length[Flatten[D1T]}]],
  Table[w3[i][0], {i, 1, Length[Flatten[F1T]}]],
  Table[w5[i][0], {i, 1, Length[Flatten[M1T]}]],
  Table[w6[i][0], {i, 1, Length[Flatten[O1T]}]],
  Table[w7[i][0], {i, 1, Length[Flatten[Q1T]}]]];

Lv = Length[varsini];

odeini =
  Thread[varsini == Join[{0, 1, 0, 0}, Table[Random[Real, {-0.1, -0.1}], {L}]]];

Length[odeini];

Tmax = 15;

```

```

sol = NDSolve[Join[ode, odeini], vars, {t, 0, Tmax}, StartingStepSize → 1/10000,
  Method → {"FixedStep", Method → "ExplicitRungeKutta"}, MaxSteps → 1000000];

G1 = Plot[Evaluate[{θ1[t], θ2[t]} /. sol], {t, 0, Tmax},
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},
  PlotRange → {{0, 15}, {-1.2, 1.2}},
  GridLines → {None, {-1, 1}}, PlotLabel → "Angulos Rojo θ1 y Azul θ2"];

G2 = Plot[{Sin[t], Cos[t]}, {t, 0, 15}]

Show[G1, G2, PlotRange → {{0, 15}, {-1.2, 1.2}}]

Show[G1, G2, PlotRange → {{11, 15}, {-1.2, 1.2}}]

Plot[Evaluate[{θ1[t] - Sin[t], θ2[t] - Cos[t]} /. sol], {t, 0, Tmax},
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},
  PlotRange → {{0, 15}, {-.00012, .00012}},
  GridLines → {None, {-.0001, .0001}}, PlotLabel → "Error Rojo θ1 y Error Azul θ2"]

Show[%, PlotRange → {{12, 15}, {-0.0001, .0001}}]

Plot[Evaluate[{dθ1[t], dθ2[t]} /. sol], {t, 0, Tmax},
  PlotStyle → {RGBColor[1, 0, 0], RGBColor[0, 0, 1]},
  PlotRange → {{0, 15}, {-1, 1}},
  GridLines → {None, {0, 2}},
  PlotLabel → "Velocidad de θ1 Rojo; Velocidad de θ2 Azul "];

Show[%, PlotRange → {{6, 10}, {-1, 1}}]

ParametricPlot[{θ1[t] - Sin[t], θ2[t] - Cos[t]} /. sol, {t, 0, Tmax},
  AxesLabel → {θ1, θ2}, PlotStyle → {RGBColor[0, 1, 0]}]

```