UNIVERSIDAD PANAMERICANA

Mecatrónica y Control de Sistemas

Control Moderno

Parte III:
Controladores no-lineales

Introducción

Objetivo:
Dado un sistema físico a controlar y las especificaciones de su respuesta deseada, construir una ley de control para hacer que el sistema en lazo cerrado presente el comportamiento deseado.

Las tareas de los controladores pueden dividirse en 2 categorías:

- Estabilización
- Tracking

Estabilización

En tareas de estabilización, un *estabilizador* o *regulador* se diseña de tal manera que la respuesta del sistema en lazo cerrado se estabilice alrededor de un punto de equilibrio.

Problema en estabilización:
Dado un sistema dinámico no-lineal:

$$x' = f(x, u, t)$$

Encontrar una ley de control **u** para que independientemente del punto donde se empiece en una región, el estado **x** tienda a **0** cuando **t**→∞.

Si la ley de control **u** se basa en la medición de señales, se le considera una *ley de control estática*. Si se basa en mediciones a través de ecuaciones diferenciales, se le considera *ley de control dinámica*.

Tracking

En tareas de tracking, un *seguidor* o *tracker* se diseña de tal manera que la salida del sistema siga la trayectoria de la señal de entrada.

Problema en tracking:
Dado un sistema dinámico no-lineal:

$$x' = f(x, u, t) \qquad y = h(x)$$

y la trayectoria deseada $\mathbf{y_d}$, encontrar una ley de control **u** de tal forma que de un estado inicial el error $\mathbf{y(t)-y_d(t)}$ tienda a cero mientras el estado **x** permanece delimitado. Esta propiedad implica cero error en el tracking (tracking perfecto) todo el tiempo:

$$y(t) \equiv y_d(t)$$

En general, los problemas de tracking son mas difíciles que los de estabilización porque en tracking, el controlador no solo debe conservar al sistema estable sino también llevar la salida del sistema al valor deseado.

Diseño de Controladores

Para el diseño de controladores no-lineales se tiene que considerar:

Estabilidad
La estabilidad debe ser garantizada para el modelo ya sea en sentido local o global. La región de estabilidad y convergencia son también de gran interés.

Precisión y velocidad de respuesta
Ambas pueden ser consideradas para trayectorias de movimiento típicas en la región de operación.

Robustez
Es la sensibilidad a fenómenos no considerados en el diseño como perturbaciones, ruido, parámetros no modelados, etc.

Costo
El costo de un controlador se determina por el número y tipo de sensores, actuadores, circuitería y procesadores necesarios para implementarlo.

Diseño de Controladores

Dado un sistema a controlar, uno típicamente sigue este procedimiento::

1. Especificar el comportamiento deseado y seleccionar los actuadores y sensores.

2. Modelar el proceso por ecuaciones diferenciales.

3. Diseñar la ley de control para el proceso.

4. Analizar y simular el desempeño del controlador

5. Implementar el controlador en hardware/software.

Métodos para el diseño de controladores no-lineales

Linealización por retroalimentación (Feedback linearization)
La idea de base es transformar algebraicamente la dinámica de un sistema no-lineal a uno (total o parcialmente) lineal. Entonces todas las técnicas de control lineal se pueden aplicar.

Control robusto
En control no-lineal clásico, la ley de control se diseña únicamente en base al modelo del sistema. Sin embargo, el comportamiento del controlador en presencia de incertidumbres no está claro. En control robusto, el controlador se diseña basándose en ambos: en el modelo y en la caracterización de sus incertidumbres.

Control adaptativo
Un controlador adaptativo difiere de un ordinario en el sentido que sus parámetros son variables y existen mecanismos para ajustarlos en tiempo real.

Feedback Linearization

Concept:

Feedback linearization techniques can be viewed as ways of transforming original system models into equivalent models of a simpler form.

The idea of feedback linearization, i.e. of canceling the nonlinearities and imposing a desired linear dynamics, can be simply applied to a class of nonlinear systems described by the *controllability canonical form*:

$$x^{(n)} = f(X) + b(X)u$$

u: scalar control input
x: scalar output of interest
X: state vector $X = \begin{bmatrix} x, & x', \ldots, & x^{(n-1)} \end{bmatrix}^T$
f(X), ***b(X)***: nonlinear functions of the states

In state space:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ \cdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \cdots \\ x_n \\ f(X) + b(X)u \end{bmatrix}$$

For systems in canonical form, the control input:

$$u = \frac{1}{b}(v - f) \qquad \text{with} \quad v = x^{(n)}$$

actually cancels the nonlinearities and obtains a simple output relation. For tasks involving the tracking of a desired output $x_d(t)$, the control law:

$$v = x_d^{(n)} - k_1 e - k_2 e' - \ldots - k_n e^{(n-1)} \tag{1}$$

with:

$$e = x(t) - x_d(t)$$

leads to exponentially convergent tracking.

The stable error dynamics is then:

$$e^{(n)} + k_n e^{n-1} + \ldots + k_1 e = 0$$

and yields exponential convergence of ***e*** to zero.

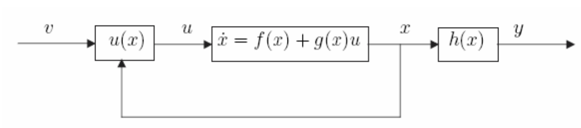
Advantages:

- Simple and fast-to-implement method

Limitations:

- It cannot be used for all non-linear systems
- The full state has to be measured
- No robustness is guaranteed in the presence of parameter uncertainty or un-modeled dynamics

How to perform input-output linearization of SISO systems: x'=f(x)+g(x)u   y=h(x)



- Differentiate the output ***y*** repeatedly until the input ***u*** appears.
- Design ***u*** to cancel the nonlinearity.
- Design a control law for ***v*** according to the canonical form (1).

Note:
- When using $e = x(t) - x_d(t)$, k >0
- When using $e = x_d(t) - x(t)$, k <0

Sliding Control

Concept:

Now we allow models to be imprecise. Model imprecision may come from actual uncertainty about the plant parameters.

A simple approach to deal with uncertainties is sliding control. It is based on the remark that it is easier to control 1st order systems (be they nonlinear or uncertain), than it is to control nth order systems. So the idea is to replace nth order problems by equivalent 1st order problems.

Consider the dynamic system:

$$x^{(n)} = f(X) + b(X)u \qquad (1)$$

u: scalar control input
x: scalar output of interest
X: state vector $X = \left[x, \quad x', \ldots, \quad x^{(n-1)}\right]^T$
f(X), ***b(X)***: nonlinear functions of the states <u>not exactly known (i.e they contain uncertainties)</u>

The control problem is to get the state **X** to track a desired state $\boldsymbol{X_d}$ in the presence of imprecision on ***f(X)*** and ***b(X).***

Notions

Let $\tilde{x} = x - x_d$ be the tracking error and let

$\tilde{x} = x - x_d = \left[\tilde{x} \quad \tilde{x}' \quad \ldots \quad \tilde{x}^{(n-1)}\right]^T$
be the tracking error vector.

Furthermore, let us define a time-varying surface *S(t)* by the scalar *s(x,t)*, where:

$$s(x,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} \qquad (2)$$

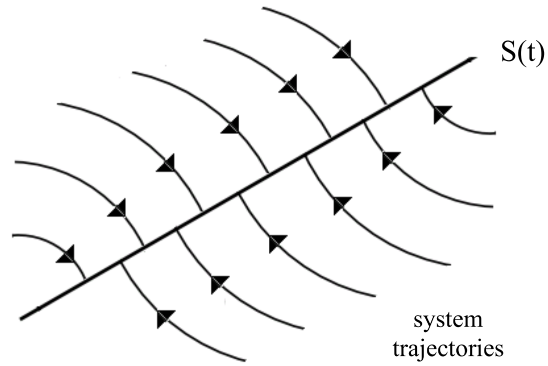
with λ strictly positive constant.

Ex:
n=2 $\quad s = \tilde{x}' + \lambda\tilde{x}$
n=3 $\quad s = \tilde{x}'' + 2\lambda\tilde{x}' + \lambda^2\tilde{x}$

The dynamics while in sliding mode is $s' = 0$ (3)

The problem of tracking $x \equiv x_d$ is equivalent to that of remaining on the surface *S(t)* for all *t>0*. In particular, once on the surface, the system trajectories remain on the surface. This implies that some disturbances or dynamic uncertainties can be tolerated while still keeping on the surface.

Graphically, trajectories off the surface can "move" while still pointing towards the surface. *S(t)* is referred as the <u>sliding surface</u> and the system's behavior once in the surface is called <u>sliding mode</u>.



For systems in form (1), the best approximation $\hat{u}$ of a continuous law would achieve $s' = 0$.

In order to satisfy the sliding condition despite uncertainty on the dynamics *f*, we add to $\hat{u}$ a term discontinuous across the surface s=0:

$$u = \hat{u} - k \ \ \mathrm{sgn}(s) \qquad (4)$$

with:
sgn(s)= +1 s>0
sgn(s)= -1 s<0

and $\quad k = F + \eta \qquad (5)$

Notes:
- η is the time to reach the sliding surface. It is chosen to be small as compared to the average value of *k*.
- ***F*** is some function that bounds the estimation error $\hat{f}$ on f : $\left|\hat{f} - f\right| \le F$
- λ must be chosen as: $\lambda = \frac{1}{5}v$, where *v* is the sampling rate.

How to design a sliding controller

- Relate your system to the general form (1)
- Obtain your sliding surface with (2)
- Obtain the sliding dynamics with (3) and replace your term $\tilde{x}^n$ with $\tilde{x}^n = x^n - x_d{}^n$
- Substitute x^n for $f + bu$ (eq. (1)). Remember input u and all constants are now estimates.
- Replace your estimate input $\hat{u}$ for input *u* using (4). There you have your control input *u*.
- Calculate *k* with (5), λ and choose η small.

Adaptive Control

Concept:

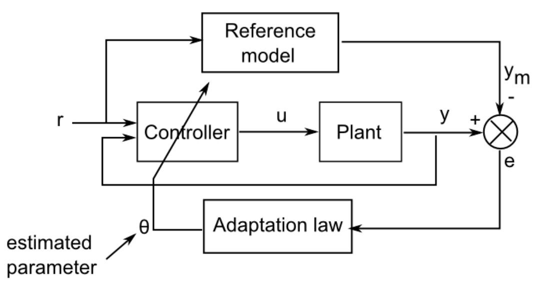
Adaptive control is a method of automatically adjusting the controller parameters in the face of changing/uncertain parameters. It is considered as a control system with on-line parameter estimation.

Advantages & Drawbacks:

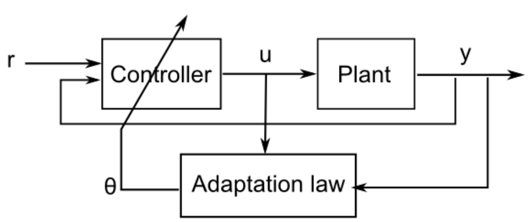
1) An adaptive controller improves its performance as adaptation goes on.
2) An adaptive controller requires little or no a priori information about the unknown parameters.
3) Not good dealing with quickly varying parameters, disturbances and unmodeled dynamics.

Two main approaches for adaptive controllers:

1) Model-Reference Adaptive Control (MRAC)



2) Self-tuning controller



Specifications for MRAC:

a) One adjustable parameter θ (adaptation of a feedforward gain)

a.1. Adaptation laws

MIT Rule: $$\frac{d\theta}{dt} = -\gamma y_m e$$

Lyapunov rule: $$\frac{d\theta}{dt} = -\gamma r e$$

γ is the adaptation gain.

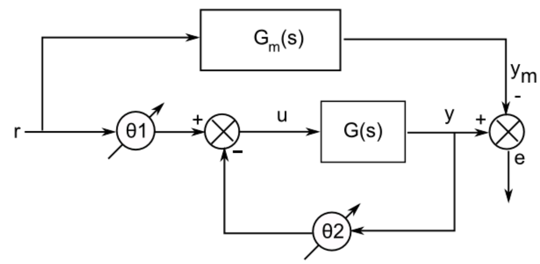
a.2. Controller

$$u = \theta r$$

b) Two adjustable parameters θ1, θ2 (adaptive control of First-Order Systems)

Plant: $y' = -ay + bu$

Model: $y_m' = -a_m y_m + b_m r$



b.1. Adaptation laws

$$\frac{d\theta_1}{dt} = -\gamma\left(\frac{a_m}{s+a_m} r\right)e$$

$$\frac{d\theta_2}{dt} = \gamma\left(\frac{a_m}{s+a_m} y\right)e$$

b.2. Controller

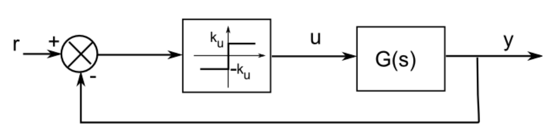
$$u = \theta_1 r - \theta_2 y$$

Self-tuning controllers:

Automatic tuning is not all autonomous; it requires prior information about time scale (the "pre-tuned mode"). Some self-tuning techniques:

-Zieger-Nichols method (PID)
-Relay feedback
-Describing functions

Specifications for relay feedback:



1) Determine the frequency ω_r of your input **r**. (It can be done by observation: $\omega_r = 2\pi/T_r$)
2) Obtain the magnitude of $\mathbf{G(j\omega_r)}$
3) Determine k_u with:

$$k_u = \frac{A}{|G(j\omega_r)|}$$

where A is the amplitude of your input signal.

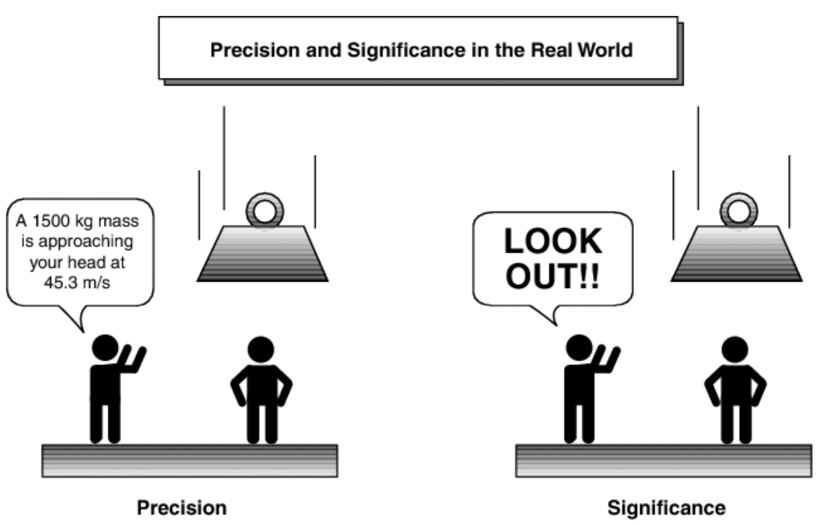
Lógica Difusa (Fuzzy logic)

Concepto:
Todos estamos limitados en nuestra habilidad de percibir, razonar y expresarnos acerca de nuestro entorno por lo que estamos llenos de imprecisiones. Todos quizás entendemos el contenido central de una plática y somos capaces de comunicarnos con un grado aceptable de precisión pero no siempre concordamos en el significado y contexto de las palabras. En resumen nuestro lenguaje es vago.

Ejemplo: "mucho", "alto", "guapo", "joven", "grande"… conceptos *verdaderos* hasta cierto punto y *falsos* hasta cierto punto. Son conceptos imprecisos, ambiguos o *difusos*.

Lógica Difusa

La lógica difusa es acerca de la importancia relativa de la precisión: ¿Qué tan importante es ser extremadamente preciso cuando una simple respuesta funciona?



Lógica Difusa

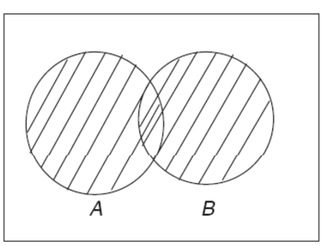
Sistema difuso:

¿Porque usar Lógica Difusa?

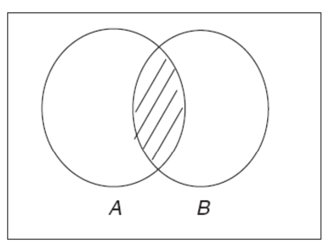
1. Es muy fácil de entender conceptualmente y es una manera muy intuitiva de abordar cualquier sistema
2. Es muy flexible
3. Es tolerante a imprecisiones
4. Puede modelar funciones no-lineales de alta complejidad
5. Se puede mezclar con otras técnicas de control
6. Esta basado en lenguaje natural: las estructuras cualitativas de descripción que todos entendemos y usamos.

Lógica Difusa

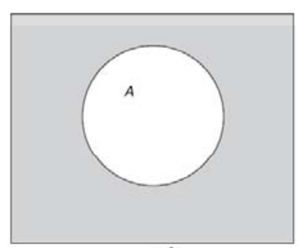
Bases



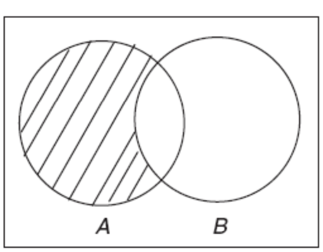
$A \cup B$



$A \cap B$



A^c



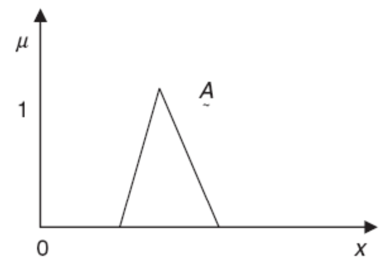
$A|B$

Ley de De Morgan:

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$
$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

Funciones de membrecía

Función que asigna valores (grados de membrecía) a cada elemento de un conjunto. El grado de membrecía μ va de 0 a 1. Los valores altos de μ denotan mayor grado de membrecía.

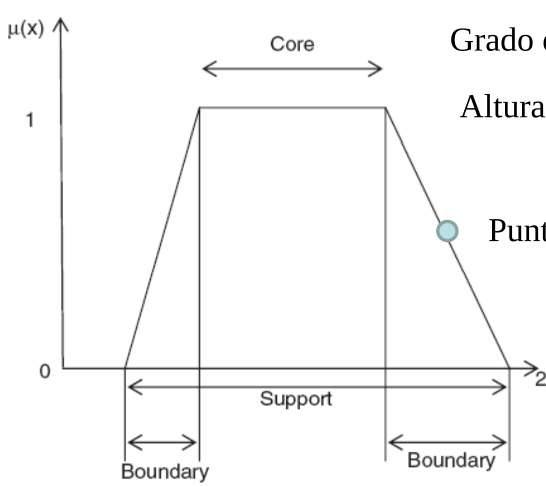


Un conjunto difuso es un conjunto de elementos con diferentes grados de membrecía. Los conjuntos difusos se denotan por una tilde debajo del conjunto (Ex: $\underset{\sim}{A}$) y siguen todas las propiedades de los conjuntos (unión, intersección, complemento, De Morgan …). Otras:

$$\underset{\sim}{A} / \underset{\sim}{B} = \underset{\sim}{A} \cap \overline{\underset{\sim}{B}}$$

$$\underset{\sim}{B} / \underset{\sim}{A} = \underset{\sim}{B} \cap \overline{\underset{\sim}{A}}$$

Propiedades - Funciones de membrecía



Grado de membrecía completa μ=1

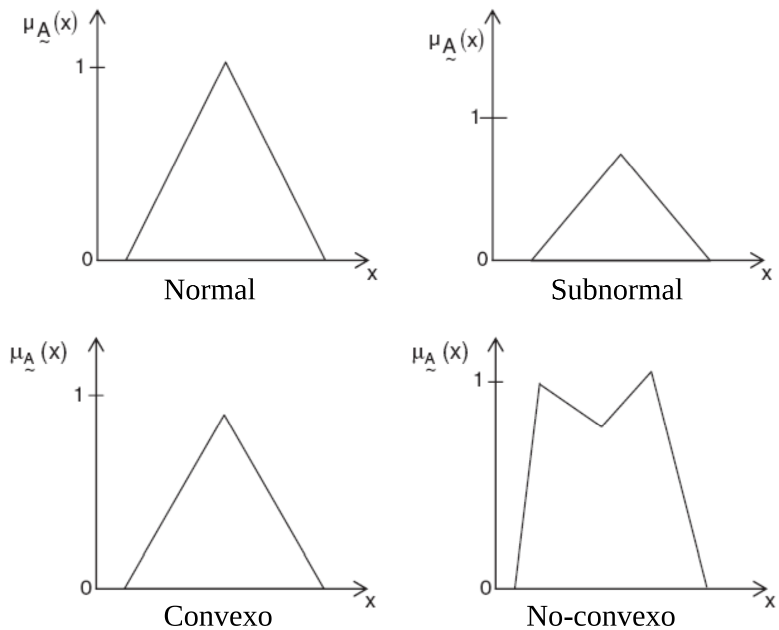
Altura: max(μ)

Punto de cruce μ=0.5

Grado de membrecía μ>0

0<μ<1

Clasificación de conjuntos difusos



Fuzzificación

Proceso en el que las funciones de membrecía se convierten en valores difusos. Existen varios métodos para asignar valores de membrecía a variables difusas:

-Intuición (basado en la inteligencia y propio entendimiento)
-Inferencia (basado en el razonamiento deductivo)
-Ranking (basado en preferencias o importancia)
-Redes neuronales (inspirado en como funciona nuestro cerebro)
-Algoritmos genéticos (basado en la teoría de evolución de Darwin)

Sistemas difusos y reglas

Las reglas son la base del sistema difuso para obtener la salida.

Formación de reglas
Existen 3 maneras para formar reglas:

1. Asignaciones
Asignación de un valor a una variable (operador =).

2. Condicionales
Condiciones y restricciones (operadores IF, THEN)

3. Incondicionales
Cuando no hay condición especifica a satisfacer. Ejemplo: transferir el control a otro lado, conexión de condiciones, etc. (operadores AND, GOTO, ELSE)

Practica 4 – Feedback Linearization

1.- Diseñe un control para el nivel ***h*** de un tanque de agua de radio 0.5 m y una salida de descarga de 0.1 m. La entrada del sistema es ***u*** y se desea tener una altura final de 0.6 m.

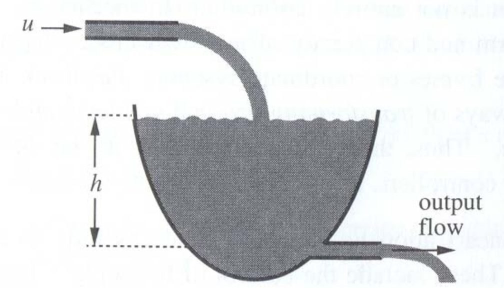


Figure 6.1 : Fluid level control in a tank

<u>Input-Output Linearization</u>

2.- Diseñe un controlador para siguientes sistemas no-lineales:

(a)

$$x_1' = senx_2$$
$$x_2' = x_1^4 \cos x_2 + u$$
$$y = x_1$$

(b)

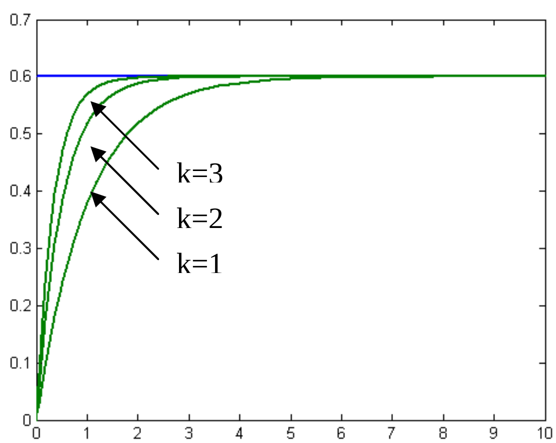
$$x_1' = x_2^3 + u$$
$$x_2' = u$$
$$y = x_1$$

(c)

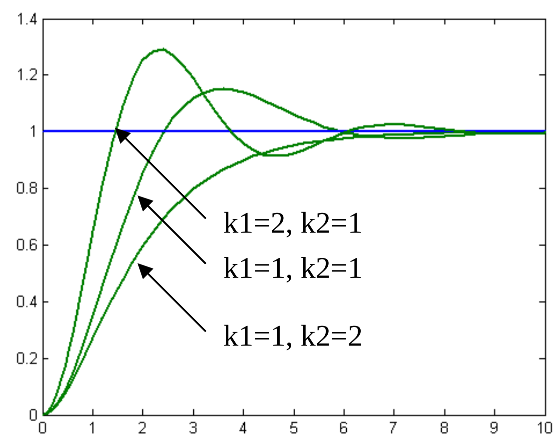
$$x_1' = -x_1 + e^{2x_2} u$$
$$x_2' = 2x_1 x_2 + senx_2 + \frac{1}{2} u$$
$$x_3' = 2x_2$$
$$y = x_3$$

<u>Soluciones:</u>

1)



2(a)



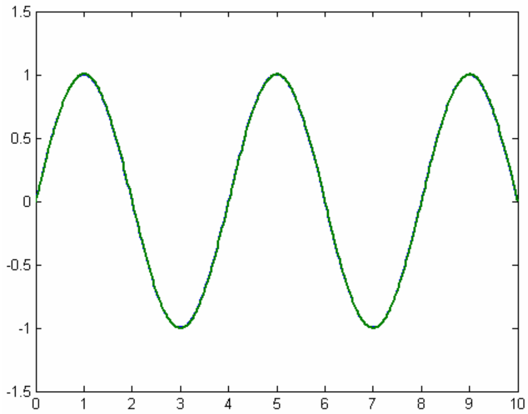
Practica 5 – Sliding Control

1.- Considere el sistema: $x''+a(t)x'^2\cos 3x = u$ donde *a(t)* es un parámetro incierto que satisface: $1 \le a(t) \le 2$.

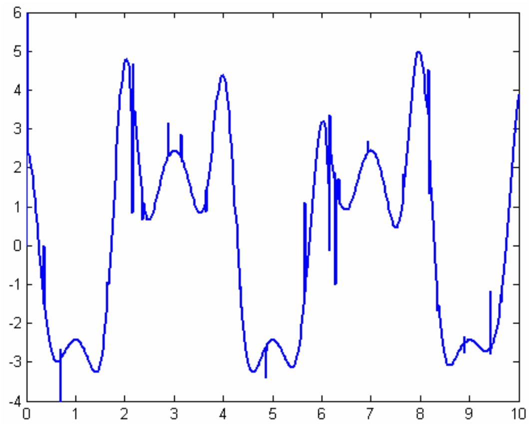
Diseñe un controlador en modo deslizante para este sistema asumiendo que la trayectoria deseada es: $x_d = sen(\pi t/2)$, una frecuencia de muestreo de 1 Khz y un comportamiento dinámico de a(t): $a(t) = |sen(t)| + 1$ (que satisface la condición anterior).

Sol:

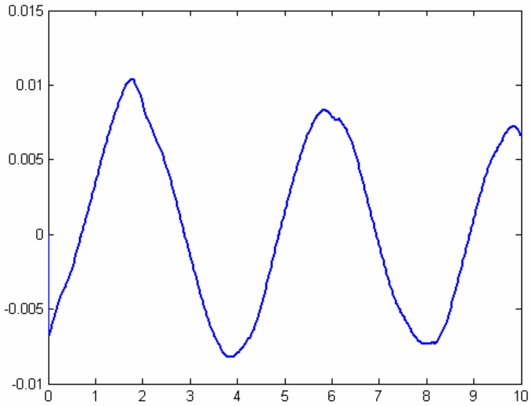
¡Tracking perfecto!



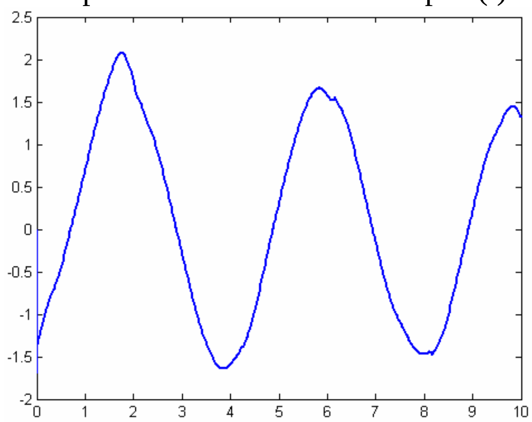
Señal de control ***u***



Tracking error $\tilde{x}$ (note la amplitud)



Superficie variante en el tiempo s(t)



2.- Considere el sistema: $mx''+c|x'|x' = u$ donde *m, c* son parámetros inciertos que satisfacen: $1 \le m \le 5$, $0.5 \le c \le 1.5$

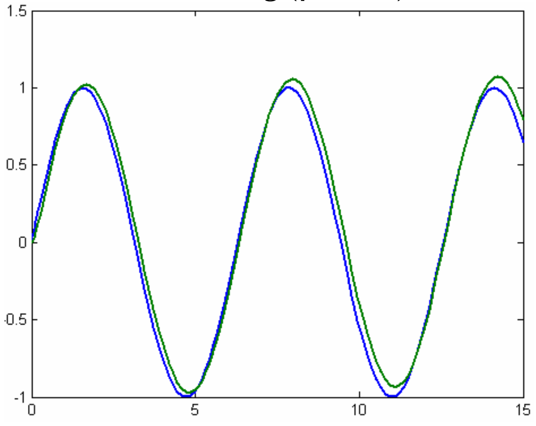
Diseñe un controlador en modo deslizante para este sistema asumiendo que la trayectoria deseada es: $x_d = sen(t)$, una frecuencia de muestreo de 50 Hz y un comportamiento dinámico de los parámetros *m, c*:

$$m = 3 + 1.5 sen(|x'|t) \qquad c = 1.2 + 0.2 sen(|x'|t)$$

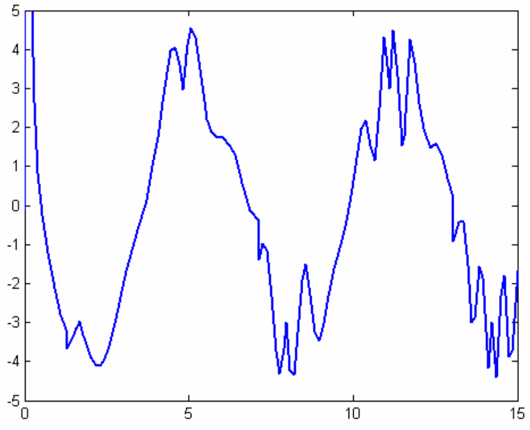
(que satisfacen la condición anterior)

Sol:

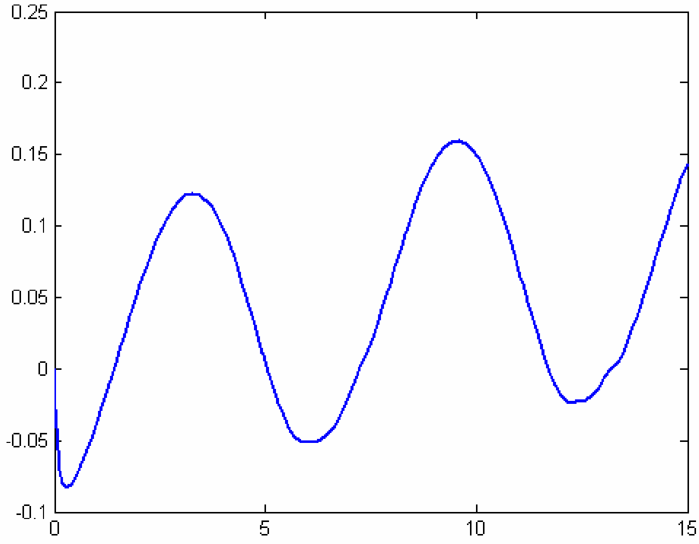
Tracking (¡bueno!)



Señal de control ***u***



Tracking error $\tilde{x}$ (¡incrementando! Ajustar λ, intente λ=5)



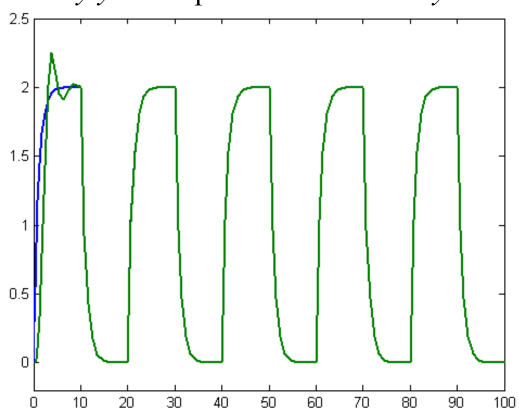
<u>En conclusión:</u> El control en modo deslizante (sliding control) sirve para controlar sistemas en cuyo modelo hay parámetros que son inciertos y además variantes en el tiempo o dependientes de otra variable (ya no es un valor constante como siempre lo habíamos considerado).

Practica 6 – Adaptive Control
(MRAC systems)

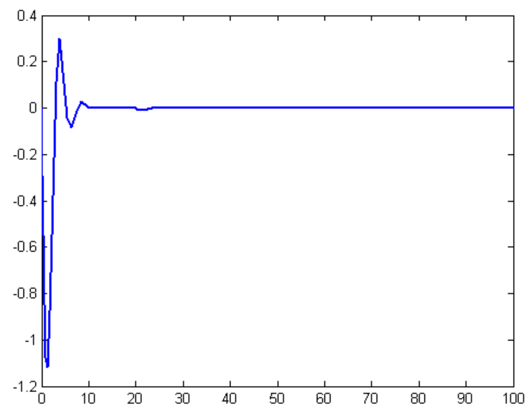
1.- Considere el control del proceso $G(s)=\dfrac{1}{s+1}$ al cual se le aplica un comando *r(t)*. Una manera de especificar la respuesta ideal de *G(s)* es usar el modelo de referencia $G_m(s)=\dfrac{2}{s+1}$. Diseñe un control adaptativo para este sistema para varios valores de γ. Use una señal cuadrada (A=1, f=0.05 Hz) como la entrada *r(t)*.

Sol (γ =1):

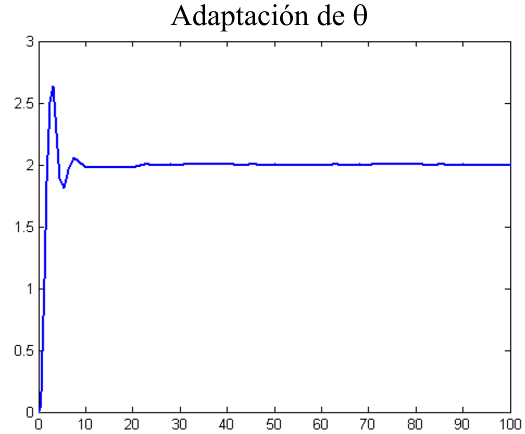
y/ym -Adaptación de la salida *y*



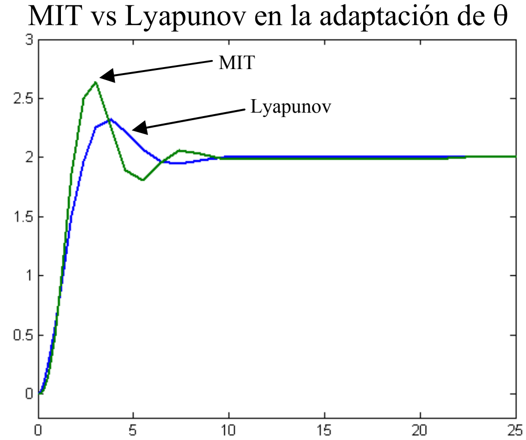
Error *e*



Adaptación de θ



MIT vs Lyapunov en la adaptación de θ



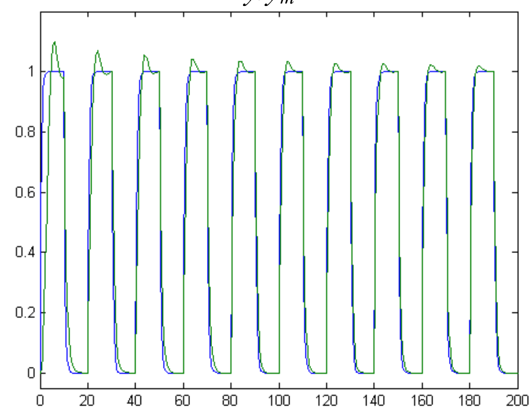
2. Diseñe un control adaptativo para el siguiente sistema MRAC

Modelo: $y'_m=-2y_m+2r$
Planta: $y'=-y+0.5u$

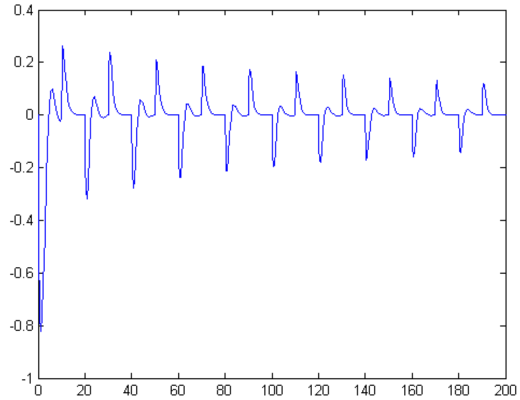
Use una señal cuadrada (A=1, f=0.05 Hz) como la entrada *r(t)*.

Sol (γ =1):

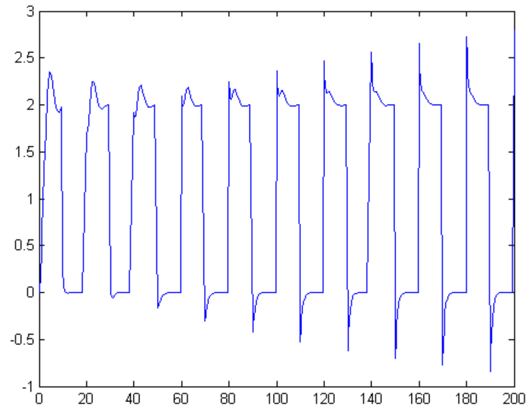
y/y_m



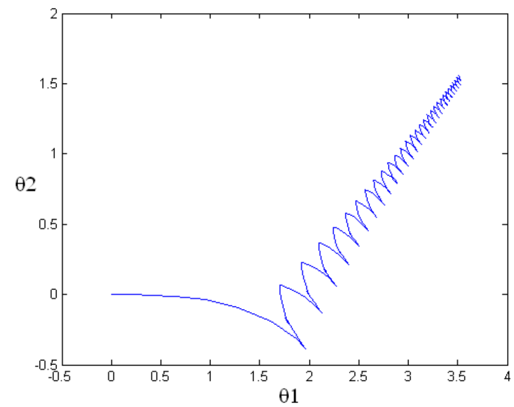
Error e



Señal de control u



$\theta 1/\theta 2$ – Convergencia en t=600s



(Self-tuning systems)

3.- Diseñe un control adaptativo autoajustable con relays para $G(s)=\dfrac{1}{s+1}$. Use una señal senoidal (A=1, f=1 rad/s) como la entrada *r(t)*.

4.- Considere el proceso $G(s)=\dfrac{1}{s+2}$. Observe el efecto de variar tanto la amplitud como la frecuencia de la señal de entrada (use una senoidal) en un controlador adaptativo con relays.

5.- Diseñe un control adaptativo con relays para $G(s)=\dfrac{1}{(s+2)(s+4)}$. Use una señal senoidal (A=5, f=3 rad/s) como la entrada *r(t)*.

Practica 7 – Lógica Difusa

1.- Se busca implementar un controlador difuso para un tanque de agua de capacidad 1 L. En particular, se busca controlar la válvula de agua que permite el llenado del tanque. Considere el nivel de agua del tanque como variable principal. Sean los estados:

nivel bajo: 0 - 0.4 L
nivel medio: 0.3 - 0.7 L
nivel alto: 0.6 – 1 L

Y las acciones a tomar:

- Si el nivel es *bajo* entonces *abrir* la válvula
- Si el nivel es *medio* entonces *no hay cambio*
- Si el nivel es *alto* entonces *cerrar* la válvula

a) Implemente en *Simulink* el controlador difuso y verifique su funcionamiento
b) Acople su funcionamiento al tanque descrito por:

$$Ah' = u - a\sqrt{2gh}$$

Donde **h** es el nivel de agua, **u** es el flujo de agua de la válvula y **A**, **a**, **g** son contantes (A=0.7853, a=0.0314, g=9.81).

2.- Se busca implementar un controlador difuso para determinar la propina que debe dejarse en un restaurante a partir de la calidad del servicio y la comida. Las premisas son:

-Si el servicio es *pobre* y la comida *mala* entonces la propina será *poca*.
- Si el servicio es *regular* independientemente de la calidad de la comida entonces la propina será *promedio*.
- Si el servicio es *excelente* y la comida *buena* entonces la propina será *generosa*.

3.- Considere el sistema en *motordc.mdl*. Este bloque representa un motor de DC donde **U** es el voltaje de entrada (12V), **T** es un torque resistivo, **th1** es la posición angular del motor y **th'** es la velocidad angular del motor.

Se busca mantener siempre la velocidad en el motor independientemente del torque resistivo.

a) Obtenga la tabla de relaciones (T-th'-U) para este motor
b) Diseñe un controlador difuso de tal forma que:

Si el motor va más lento por T=1, entonces se suministre más voltaje
Si el motor va mucho más lento por T=2, entonces se suministre mucho más voltaje
Si el motor va a velocidad normal (T=0), entonces no hay cambio

c) Integre el controlador al sistema y verifique su correcto funcionamiento.