

# Apunte de Control Avanzado de Sistemas

## EL4105

**Departamento de Ingeniería Eléctrica**  
*Versión 31/03/14*

**Autor:** Dra. Doris Sáez H.

**Colaborador:** Patricio J. Santis T.

**Agradecimientos:** Dr. Guillermo González R.  
Dr. Marcos Orchard C.

Santiago, Chile

# Índice General

1. Fundamentos del Control Óptimo de Sistemas .....	1
1.1. Clasificación del Control Óptimo de Sistemas .....	1
1.2. Control Óptimo en Equilibrio determinístico .....	1
1.3. Control Óptimo Dinámico determinístico .....	2
1.3.1. Principio del Mínimo (Máximo) de Pontryagin.....	3
1.3.2. Índice J con terminal funcional.....	3
1.3.3. Controlador LQ .....	4
1.3.4. Controlador LQR .....	5
1.3.4.1. LQR tiempo continuo .....	5
1.3.4.2. LQR tiempo discreto.....	6
1.3.4.3. LQR con seguimiento de referencia .....	6
1.4. Control Óptimo Dinámico estocástico.....	7
1.4.1. Controlador LQG .....	7
1.4.2. Controlador LQGR.....	8
1.5. Problemas Resueltos .....	9
2. Introducción al Control Predictivo .....	15
2.1. Control Predictivo basado en modelos (MBPC).....	15
2.1.1. Estrategia el MBPC .....	16
2.1.2. Estructura Básica del Control MBPC.....	17
2.1.3. Elementos del MBPC .....	18
2.1.3.1. Modelo de predicción .....	18
2.1.3.1.1. Modelo de respuesta al impulso.....	18
2.1.3.1.2. Modelo de respuesta al escalón.....	19
2.1.3.1.3. Modelo función de transferencia.....	20
2.1.3.1.4. Modelo en espacio de estados .....	20
2.1.3.1.5. Modelo de perturbaciones .....	20
2.1.4.1.6. Modelos Estocásticos.....	21
2.1.3.2. Función objetivo .....	22
2.1.3.3. Obtención de la ley de control.....	23
2.2. Trayectoria de referencia .....	24
2.3. Predicciones para diferentes tipos de modelos .....	25

2.3.1.	Modelo respuesta al escalón.....	25
2.3.2.	Modelo función de transferencia.....	26
2.3.2.1.	Predicción de la salida.....	28
2.3.3.	Predicción con modelo en variables de estado.....	33
2.4.	Estructura de los predictores.....	35
2.5.	Control de Varianza Mínima.....	35
2.5.1.	Varianza Mínima con Minimización de Energía de Control.....	37
2.6.	Control por Matriz Dinámica (DMC).....	39
2.7.	Control Predictivo Generalizado (GPC.).....	40
2.8.	Control Predictivo con Restricciones.....	46
2.8.1.	Tipos de restricciones.....	47
2.8.2.	Consideraciones: Control predictivo con restricciones.....	50
2.9.	Algoritmo de Programación Cuadrática.....	51
2.10.	Problemas Resueltos.....	53
3.	Fundamentos de la Lógica Difusa.....	61
3.1.	Funciones de Pertenencia.....	62
3.2.	Operaciones básicas de lógica difusa.....	64
3.3.	Modelos Difusos.....	65
3.3.1.	Sistema de Lógica Difusa tipo Mamdani.....	65
3.3.2.	Modelos difusos de Takagi y Sugeno.....	70
3.4.	Método de Identificación de un Modelo Takagi y Sugeno.....	72
3.4.1.	Identificación de los parámetros de las premisas.....	73
3.4.2.	Identificación de los parámetros de las consecuencias.....	76
3.4.3.	Elección de la estructura del modelo.....	77
3.4.3.1.	Método heurístico (Sugeno, 1993).....	77
3.4.3.2.	Análisis de Sensibilidades.....	78
3.5.	Control basado en modelos de Takagi y Sugeno.....	82
3.5.1.	Controladores PI difusos.....	83
3.5.2.	Controlador LQR Difuso.....	86
3.5.2.1.	Controlador LQR Difuso para un péndulo invertido.....	87
3.5.3.	Controlador Difuso por ubicación de Polos.....	91
3.5.4.	Control Predictivo basado en modelos difusos.....	93
4.	Introducción a Redes Neuronales.....	100
4.1.	Representación Matemática de una Neurona.....	101

4.2. Ventajas de las Redes Neuronales.....	103
4.3. Perceptrón Multicapa .....	104
4.4. Redes Recurrentes .....	108
4.5. Redes de Funciones de Base Radiales .....	109
4.6. Otros tipos de Redes.....	111
4.7. Entrenamiento de Redes Neuronales.....	111
4.7.1. La regla delta .....	112
4.7.2. Algoritmo de entrenamiento backpropagation .....	112
4.8. Identificación Basada en Redes Neuronales.....	114
4.9. Estructuras en identificación de redes neuronales .....	117
4.9.1. Modelación directa.....	117
4.9.2. Ecuación del sistema no lineal (Proceso).....	117
4.9.3. Red neuronal (Modelo).....	117
4.9.4. Modelación inversa .....	118
4.9.5. Modelación inversa especializada .....	118
4.10. Estructuras de control con redes neuronales .....	119
4.10.1. Control por linealización instantánea.....	119
4.10.2. Control por modelo de referencia.....	121
4.11. Control Predictivo basado en redes neuronales .....	122
4.12. Principios básicos de sistemas neuro-difusos .....	126
4.12.1. Sistema de inferencia difusa neuronal adaptiva (ANFIS) .....	126
4.12.2. Algoritmo de Aprendizaje ANFIS.....	127
4.13. Problemas Resueltos .....	129
5. Control Adaptativo de sistemas.....	133
5.1. Control Adaptativo por sistema de referencia (MRAC) .....	134
5.2. Regla MIT .....	134
6. Referencias .....	136
7. Anexos.....	137
7.1. Linealización .....	137
7.2. Controlabilidad y observabilidad .....	138

# Capítulo 1

## 1. Fundamentos del Control Óptimo de Sistemas

En esta sección se expone un resumen de los contenidos. Se encuentran disponibles en u-cursos los apuntes de Control Óptimo del Profesor Guillermo González R.

### 1.1. Clasificación del Control Óptimo de Sistemas

El control óptimo se basa en la definición de un criterio de desempeño. Además, se pueden incluir también restricciones. Se genera una acción de control óptima según el criterio establecido.

El Control Óptimo puede clasificarse según

$$\text{Control Óptimo} \left\{ \begin{array}{l} \text{En equilibrio} \left\{ \begin{array}{l} \text{Determinístico} \\ \text{Estocástico} \end{array} \right. \\ \text{Dinámico} \left\{ \begin{array}{l} \text{Determinístico} \\ \text{Estocástico} \end{array} \right. \end{array} \right.$$

- Determinístico: Modelo y señal conocida.
- Estocástico: Modelo conocido y señal estocástica; Modelo incierto y señal estocástica.

### 1.2. Control Óptimo en Equilibrio determinístico

Un sistema dinámico

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= h(x, u)\end{aligned}$$

Está en equilibrio ssi  $\dot{x} = 0$ , lo que genera puntos de equilibrio en función de la entrada  $u$ . Entonces, se generan los estados de equilibrio:

$$x_e = \phi(u)$$

Reemplazando es posible expresar el sistema en equilibrio:

$$y_e = h(x_e, u) = h(\phi(u), u) = h(u)$$

Se desea encontrar el  $u^*$  tal que resuelva el problema para la siguiente función de costo:

$$\begin{aligned} \min J &= J(u, y) \\ \text{s.a. } y &= h(u) \\ g_i(u, y) &\leq 0, \quad i = 1, \dots, v \\ g_i(u, y) &= 0, \quad i = v + 1, \dots, n \end{aligned}$$

Entonces, para resolver el problema anterior se utiliza el teorema de Karush-Kuhn-Tucker, el cual es una generalización de los multiplicadores de Lagrange, el cual dice que existen  $n$  multiplicadores  $\lambda_{i=1, \dots, n}$  tales que:

$$\begin{aligned} \lambda_i &\geq 0, \quad i = 1, \dots, v \quad (\text{restricciones de desigualdad}) \\ \frac{\partial J}{\partial x} + \lambda \frac{\partial g}{\partial x} &= 0 \\ \lambda_i g_i &= 0, \quad i = 1, \dots, v \quad (\text{restricciones de desigualdad}) \\ \lambda_i &\in \mathbb{R}, \quad i = v + 1, \dots, n \quad (\text{restricciones de igualdad}) \end{aligned}$$

Donde  $\lambda = [\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n]$  y  $\frac{\partial g}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_2}{\partial x} & \dots & \frac{\partial g_n}{\partial x} \end{bmatrix}$

### 1.3. Control Óptimo Dinámico determinístico

Se tiene una planta modelada por la ecuación de estado:

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

Con  $x(t_0) = x_0$  y  $x(T) = x_T$  dados. Se desea encontrar el control óptimo  $u[t_0, T]$  entre el periodo  $[t_0, T]$  que minimice el funcional o índice de rendimiento del sistema (o que lo maximice):

$$J = \phi(x(T)) + \int_{t_0}^T L(x, u) dt$$

Sujeto a ciertas restricciones.

Formalmente se considere el siguiente problema:

$$\begin{aligned} \min J &= \int_{t_0}^T L(x, u) dt \\ \text{s.a. } \dot{x} &= f(x, u), \quad u \in \mathfrak{R}^r, x \in \mathfrak{R}^n \\ x(t_0) &= x_0, \quad x(T) = x_T \end{aligned}$$

Para resolver el problema se introduce la función **Hamiltoniana**:

$$H(x, u, t, \lambda) = L(x, u, t) + \lambda^T f(x, u, t)$$

Con  $\lambda = [\lambda_1 \ \lambda_2 \ \dots \ \lambda_n] \in \mathfrak{R}^n, \lambda \in C^1(T)$  multiplicadores de Lagrange. Las ecuaciones que resuelven este problema de optimización son:

$$\dot{x} = f(x, u, t), \quad \frac{d\lambda^T}{dt} = -\frac{\partial H(\lambda, x, u, t)}{\partial x}, \quad \frac{\partial H(\lambda, x, u, t)}{\partial u} = 0, \quad t \in [t_0, T]$$

Donde,  $\frac{d\lambda^T}{dt} = -\frac{\partial H(\lambda, x, u, t)}{\partial x} \Rightarrow \dot{\lambda} = -\frac{\partial f^T}{\partial x} \lambda - \frac{\partial L^T}{\partial x}$  se denomina ecuación de coestado.

### 1.3.1. Principio del Mínimo (Máximo) de Pontryagin

El principio del mínimo de Pontryagin, explicita que el proceso admisible  $(x^*(t), u^*(t), \lambda^*(t))$  es óptimo si se cumple:

$$H(x^*, u^*, \lambda^*, t) \leq (\geq) H(x^*, u, \lambda^*, t), \quad t \in [t_0, T]$$

es decir,  $u^*$  resuelve el problema de control óptimo si minimiza el Hamiltoniano sobre todo el conjunto de señales de control admisible en  $t \in [t_0, T]$ .

### 1.3.2. Índice $J$ con terminal funcional

En este caso el funcional a minimizar es:

$$J = \phi(x(T)) + \int_{t_0}^T L(x, u) dt$$

La diferencia con el caso anterior es que el vector de estado final  $x(T)$  no está dado, sino que es variable de optimización. Con esta formulación, el coestado al instante  $T$  cumple:

$$\lambda(T) = \frac{\partial \phi(x(T))}{\partial x}$$

También las ecuaciones de Hamiltoniano y la ecuación de coestado.

Como se vio anteriormente, en el caso de que no existan restricciones para  $u$ , se tiene que:

$$\frac{\partial H(\lambda, x, u, t)}{\partial u} = 0, \quad t \in [t_0, T]$$

Si existen las restricciones  $g = [g_1 \ g_2 \ \dots \ g_m]^T$ , la minimización del Hamiltoniano debe ser vista con las ecuaciones:

$$\begin{aligned} \frac{\partial H}{\partial u} + \mu \frac{\partial g}{\partial u} &= 0, \quad t \in [t_0, T] \\ \mu_j &\geq 0, \quad j = 1, \dots, m \\ \mu_j g_j &= 0, \quad j = 1, \dots, m \end{aligned}$$

Donde  $\mu = [\mu_1 \ \mu_2 \ \dots \ \mu_m]$  son los multiplicadores de Lagrange correspondientes.

### 1.3.3. Controlador LQ

El control lineal cuadrático LQ consiste en regular un sistema lineal mediante la minimización de un funcional cuadrático.

Sea el sistema línea:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \\ x &\in \mathbb{R}^n, u \in \mathbb{R}^m \end{aligned}$$

Se desea llevar el sistema de  $x(t_0) = x_0$  a una vecindad del origen, con una señal de control simple. Notar que  $x(T) = x_T$  es libre y por lo tanto es una variable de optimización.

Para lograr lo anterior se debe minimizar

$$\begin{aligned} \min J &= \frac{1}{2} x^T(t_f) S_f x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \\ \text{s.a. } \dot{x} &= Ax + Bu \\ x(t_0) &= x_0 \end{aligned}$$

Donde las matrices  $Q, R, S_f$  son matrices definidas positivas. Las matrices anteriores pueden considerarse simétricas, sin pérdida de generalidad.



Para resolver el problema anterior, se considera el Hamiltoniano:

$$H = \frac{1}{2}(x^T(t)Qx(t) + u^T(t)Ru(t)) + \lambda^T(Ax(t) + Bu(t))$$

Entonces, con las condiciones del Hamiltoniano anteriormente mencionadas se resuelve este problema de control óptimo.

Se asume que el coestado, puede ser escrito como  $\lambda(t) = S(t)x(t)$

$$u(t) = -R^{-1}B^T \lambda(t) \Rightarrow u(t) = -R^{-1}B^T S(t)x(t)$$

Donde la ecuación diferencial de Riccati, viene dada por:

$$\begin{aligned}\dot{S} &= -SA - A^T S + SBR^{-1}B^T S - Q \\ S(t_f) &= S_f\end{aligned}$$

#### 1.3.4. Controlador LQR

El LQR pretende mantener el estado dentro de una vecindad que contenga el origen, por lo que no existe el concepto de tiempo final  $t_f$ , y tampoco el de estado final  $x(t_f)$ .

##### 1.3.4.1. LQR tiempo continuo

Se considera el sistema definido por:

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx \\ x &\in \mathfrak{R}^n, u \in \mathfrak{R}^m\end{aligned}$$

Se desea resolver el problema:

$$\begin{aligned}\min J &= \frac{1}{2} \int_{t_0}^{\infty} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \\ s.a. \quad \dot{x} &= Ax + Bu \\ x(t_0) &= x_0\end{aligned}$$

Donde Q y R son matrices constantes. El término  $\frac{1}{2} x^T(t_f) S_f x(t_f)$  no aparece, pues no existe un estado final.

Resolviendo el problema anterior con el Hamiltoniano:

$$H = \frac{1}{2}(x^T(t)Qx(t) + u^T(t)Ru(t)) + \lambda^T(Ax(t) + Bu(t))$$

Se considera que el coestado es el mismo que en el control LQ. Resolviendo lo anterior, se tiene que el control viene dado por

$$u(t) = -R^{-1}B^T S(t)x(t)$$

Dado que en  $t_f - t_0 \rightarrow \infty$   $S(t)$  alcanza el régimen permanente, se tendrá que la ecuación de Riccati se transforma en una algebraica, es decir:

$$-SA - A^T S + SBR^{-1}B^T S - Q = 0$$

### 1.3.4.2. LQR tiempo discreto

Sea el sistema definido por:

$$x(k+1) = Ax(k) + Bu(k)$$

Se desea resolver el problema:

$$\begin{aligned} \min J &= \frac{1}{2} \sum_{k=0}^{\infty} [x(k)^T Q x(k) + u(k)^T R u(k)] \\ \text{s.a. } x(k+1) &= Ax(k) + Bu(k) \\ x(t_0) &= x_0 \end{aligned}$$

Resolviendo el problema anterior, se llega a que el controlador viene dado por:

$$u(t) = -Kx(k)$$

Donde:

$$K = (R + B^T P B)^{-1} B^T P A$$

La ecuación de Riccati viene dada por:

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A$$

### 1.3.4.3. LQR con seguimiento de referencia

Sea el sistema:

$$\left. \begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \right\} x \in \mathbb{R}^n, u \in \mathbb{R}^m$$

Se desea resolver el problema:

$$\begin{aligned} \min J &= \frac{1}{2} \int_{t_0}^{t_f} ((y^* - Cx)^T Q (y^* - Cx) + u(t)^T R u(t)) dt \\ \text{s.a. } \dot{x} &= Ax + Bu \end{aligned}$$

Donde  $Q$  y  $R$  son matrices definidas positivas (simétricas, sin pérdida de generalidad).

Sea el Hamiltoniano dado por:

$$H = \frac{1}{2}((y^* - Cx)^T Q(y^* - Cx) + u^T(t)Ru(t)) + \lambda(Ax(t) + Bu(t))$$

Resolviendo el problema, se llega al controlador de la forma:

$$\begin{aligned} u(t) &= \underbrace{-R^{-1}B^T Sx}_{-K} - \underbrace{R^{-1}B^T G}_H \\ \Rightarrow u(t) &= -Kx + H \end{aligned}$$

Donde:

$$\begin{cases} \dot{S} + SA + A^T S - SBR^{-1}B^T S + C^T QC = 0 \\ \dot{G} + (A^T - SBR^{-1}B^T)G - C^T Qy^* = 0 \\ S(t_f) = 0, \quad G(t_f) = 0 \end{cases}$$

#### 1.4. Control Óptimo Dinámico estocástico

Cuando el sistema presenta perturbaciones estocásticas se requiere la introducción de un estimador de estado óptimo para sistemas estocásticos.

##### 1.4.1. Controlador LQG

Sea el sistema

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ x &\in \mathfrak{R}^n, u \in \mathfrak{R}^m \end{aligned}$$

Donde los ruidos blancos  $w$  y  $v$  tienen covarianzas:

$$\begin{aligned} \mathbb{E}(w(t)w(t-\tau)) &= W(t)\delta(\tau) \\ \mathbb{E}(v(t)v(t-\tau)) &= V(t)\delta(\tau) \end{aligned}$$

Para realizar el control, se minimiza el siguiente funcional:

$$J = \mathbb{E} \left[ \frac{1}{2} x(t_f)^T S_f x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \right]$$

Debido a que los estados son variables aleatorias.

El problema consiste en encontrar una estimación óptima del estado (a través de un filtro de Kalman) y determinar el control LQ asociado.

Por lo anterior, el control LQ está dado por:

$$u(t) = -K\hat{x}(t)$$

Donde  $K$  es la misma ganancia de un LQ determinístico y  $\hat{x}(t)$  es el estado estimado a través de Kalman.

Las ecuaciones del filtro de Kalman vienen dadas por:

$$\begin{aligned}\frac{d\hat{x}}{dt} &= A\hat{x} + Bu + \Gamma(y - C\hat{x}) & \Gamma &= PC^TV^{-1} \\ &= A\hat{x} - BK\hat{x} + \Gamma(y - C\hat{x}) & \frac{dP}{dt} &= AP + PA^T - PC^TV^{-1}CP + W, \quad P(t_0) = P_0 \\ &= [A - BK - \Gamma C]\hat{x} + \Gamma y\end{aligned}$$

#### 1.4.2. Controlador LQGR

Al igual que en el caso determinístico, es posible implementar un regulador para el estado estimado. Considerar el sistema definido en variables de estado:

$$\begin{aligned}\dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ x &\in \mathfrak{R}^n, u \in \mathfrak{R}^m\end{aligned}$$

Para realizar el control, se minimiza el siguiente funcional:

$$J = \mathbb{E} \left[ \frac{1}{2} \int_{t_0}^{t_f} (x(t)^T Q x(t) + u(t)^T R u(t)) dt \right]$$

Al igual que lo anterior el problema consiste en encontrar una estimación del estado y el controlador LQR  $u(t) = -K\hat{x}(t)$ ,  $K$  es la misma ganancia.

El filtro de Kalman en el proceso estacionario, se convierte en una ecuación de Riccati para  $P$ , se tiene que:

$$\begin{aligned}\frac{d\hat{x}}{dt} &= A\hat{x} + Bu + \Gamma(y - C\hat{x}) & \Gamma &= PC^TV^{-1} \\ &= A\hat{x} - BK\hat{x} + \Gamma(y - C\hat{x}) & 0 &= AP + PA^T - PC^TV^{-1}CP + W \\ &= [A - BK - \Gamma C]\hat{x} + \Gamma y\end{aligned}$$

## 1.5. Problemas Resueltos

### Problema 1.1

Sea el sistema definido por  $x = \begin{pmatrix} y \\ \dot{y} \end{pmatrix}$ , donde  $\ddot{y} = u$ ; dado  $x(0) = x_0 \in \mathbb{R}^2$  y que  $|u(t)| \leq 1$ . Se desea encontrar el mínimo tiempo necesario para que  $x(T) = 0$ .

### Solución

Dado que  $\ddot{y} = u$ , el sistema se puede escribir en variables de estado como:

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

En el problema de tiempo mínimo ( $\min \{T \mid x(T) = 0\}$ ) se resuelve mediante el índice de rendimiento definido por:

$$\begin{aligned} J(t) &= \int_0^T L(x, u) dt = T \Rightarrow L(x, u) = 1 \\ \text{s.a. } &|u(t)| \leq 1 \\ &x(T) = 0 \end{aligned}$$

Se considera el Hamiltoniano ( $\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$ ):

$$\begin{aligned} \min H(x, u, \lambda) &= L(x, u) + \lambda^T f(x, u) = 1 + \lambda^T \left( \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \right) \\ &= 1 + \lambda_1 x_2 + \lambda_2 u \\ \text{s.a. } &|u(t)| \leq 1 \end{aligned}$$

El control minimizará solo  $\lambda_2 u$  (principio del mínimo de Pontryagin), se resuelve:

$$\begin{aligned} \min & \lambda_2 u \\ \text{s.a. } &|u(t)| \leq 1 \end{aligned}$$

Por lo tanto

- Si  $\lambda_2(t) > 0 \Rightarrow u = -1$
- Si  $\lambda_2(t) < 0 \Rightarrow u = 1$

La ecuación del coestado, viene dada por  $\dot{\lambda} = -\frac{\partial f^T}{\partial x} \lambda - \frac{\partial L^T}{\partial x} = 0$ ,  $\lambda(T)$

$$\dot{\lambda} = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} \lambda \Rightarrow \begin{cases} \dot{\lambda}_1 = 0 \Rightarrow \lambda_1 = c_1 \\ \dot{\lambda}_2 = -\lambda_1 \Rightarrow \lambda_2 = -c_1 t + c_2 \end{cases}$$

Graficamente

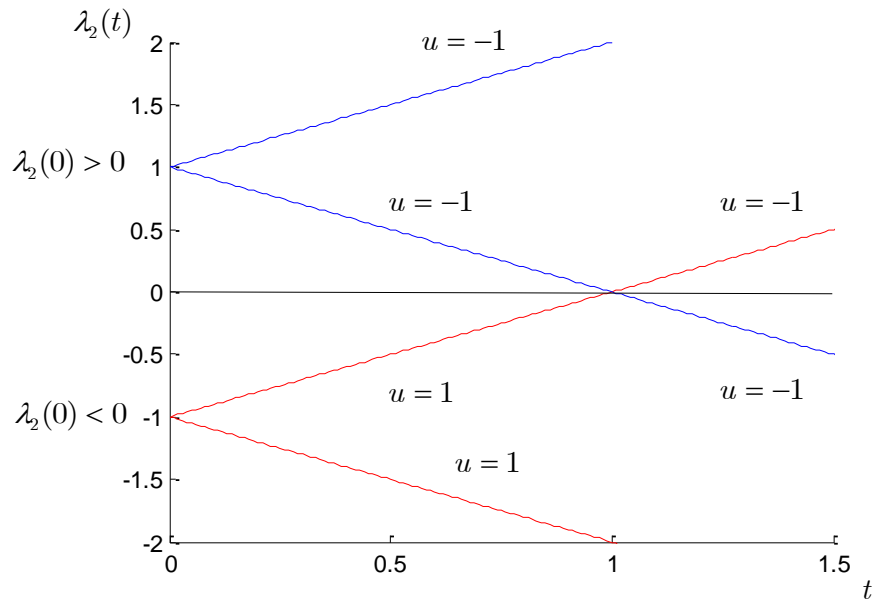


Gráfico 1.1: Posibilidades de realizar el control.

Es decir el control cambiara a lo mas una vez.

Volviendo a las ecuaciones de estado:

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \quad \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u \end{cases}$$

Por simplicidad se supone  $x(0) = 0$

- Si  $u(t) = 1$

$$\dot{x}_2 = 1 \Rightarrow x_2(t) = t \Rightarrow x_1(t) = \frac{1}{2}t^2 \Leftrightarrow x_1(t) = \frac{1}{2}x_2^2$$

- Si  $u(t) = -1$

$$\dot{x}_2 = -1 \Rightarrow x_2(t) = -t \Rightarrow x_1(t) = -\frac{1}{2}t^2 \Leftrightarrow x_1(t) = -\frac{1}{2}x_2^2$$

Para el control hay que cambiar  $u$  (de -1 a 1 o viceversa)

**Propuesto:** Estudiar los diagramas de fases y calcular el tiempo mínimo.

**Problema 1.2**

- a) Un sistema mecánico está modelado por la ecuación de estado:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = u$$

Donde  $x_1$  es la posición y  $x_2$  es la velocidad. Partiendo desde el reposo (posición y velocidad cero) en el instante inicial cero, se desea llegar a la posición 4 en el instante final  $t_f = 6$  a una velocidad cero, minimizando el esfuerzo de control

$$J = \frac{1}{2} \int_{t_0}^{t_f} u^2 dt$$

Encontrar el control y trayectoria óptima y además el valor del funcional.

- b) Considere el sistema continuo dado por (2 ptos):

$$\dot{x} = \begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u, \quad y = \begin{pmatrix} 0 & 1 \end{pmatrix} x$$

Buscar el control óptimo LQR que minimice la función de costos:

$$J = \int_0^{\infty} \left( x^T \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} x + u^2 \right) dt$$

¿Por qué es posible crear un controlador para el sistema? Demostrar.

**Solución**

- a) El sistema puede quedar como:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ u \end{pmatrix} = f(x, u)$$

Y además  $x_1(0) = x_2(0) = 0$  por enunciado, por otro lado se tiene que:

$$J = \frac{1}{2} \int_{t_0}^{t_f} u^2 dt \Rightarrow L = \frac{1}{2} u^2$$

---

<sup>1</sup> También se puede escoger  $L = u^2$

Por lo tanto planteando el Hamiltoniano:

$$H = L + \lambda^T f(x, u) = \frac{1}{2}u^2 + \lambda_1 x_2 + \lambda_2 u$$

Aplicando las condiciones para resolver el problema, como  $u$  no tiene restricciones se cumple que:

$$\frac{\partial H}{\partial u} = 0 \Leftrightarrow u + \lambda_2 = 0 \Rightarrow u^* = -\lambda_2$$

Para el coestado:

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \Leftrightarrow \dot{\lambda}_1 = 0, \quad \dot{\lambda}_2 = -\lambda_1$$

Resolviendo lo anterior se tiene:

$$\lambda_1 = K_1, \quad \lambda_2 = -K_1 t + K_2$$

Volviendo a la ecuación de estado:

$$\dot{x}_2 = u^* = -\lambda_2 = K_1 t - K_2 \Rightarrow x_2(t) = K_1 \frac{t^2}{2} - K_2 t + x_2(0)$$

$$\dot{x}_1 = x_2 = K_1 \frac{t^2}{2} - K_2 t + x_2(0) \Rightarrow x_1(t) = K_1 \frac{t^3}{6} - K_2 \frac{t^2}{2} + x_2(0)t + x_1(0)$$

Reemplazando las condiciones iniciales  $x_1(0) = x_2(0) = 0$ :

$$x_1(t) = K_1 \frac{t^3}{6} - K_2 \frac{t^2}{2}$$

$$x_2(t) = K_1 \frac{t^2}{2} - K_2 t$$

Ahora reemplazando  $x_1(6) = 4$ ,  $x_2(6) = 0$

$$x_1(6) = K_1 \frac{6^3}{6} - K_2 \frac{6^2}{2} = K_1 36 - K_2 18 = 4$$

$$x_2(6) = K_1 \frac{6^2}{2} - K_2 6 = K_1 18 - K_2 6 = 0 \Rightarrow 3K_1 = K_2$$



Entonces se tiene:

$$36K_1 - 54K_1 = -18K_1 = 4 \Rightarrow K_1 = -\frac{4}{18} = -\frac{2}{9} \Rightarrow K_2 = -\frac{2}{3}$$

Por lo tanto la trayectoria óptima de la posición y velocidad respectivamente:

$$x_1(t) = -\frac{1}{27}t^3 + \frac{1}{3}t^2$$

$$x_2(t) = -\frac{1}{9}t^2 + \frac{2}{3}t$$

El control óptimo es,

$$u^*(t) = -\lambda_2 = K_1 t - K_2 = -\frac{2}{9}t + \frac{2}{3}$$

El funcional,

$$J = \frac{1}{2} \int_0^6 \left( -\frac{2}{9}t + \frac{2}{3} \right)^2 dt = \dots$$

c) El sistema continuo:

$$\dot{x} = \underbrace{\begin{pmatrix} 1 & 0 \\ 2 & 0 \end{pmatrix}}_A x + \underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_B u$$

Minimizar,

$$J = \int_0^\infty \left( x^T \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} x + u^2 \right) dt \Leftrightarrow J = \frac{1}{2} \int_0^\infty \left( x^T \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix} x + 2u^2 \right) dt$$

Por lo cual  $Q = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}$ ,  $R = 2$

El control LQR se calcula como:

$$0 = A^T S + SA - SBR^{-1}B^T S + Q$$

$$u^*(t) = -R^{-1}B^T S \cdot x(t)$$

Resolviendo se llega a:

$$S = \begin{pmatrix} 6.47 & 2 \\ 2 & 2.2361 \end{pmatrix}$$

$$K = (3.23 \quad 1)$$

**¿Por qué es posible crear un controlador para el sistema? Demostrar.**

Calculando la matriz de controlabilidad,

$$C = (B \quad AB) = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix} \Rightarrow \det C = 2 \neq 0$$

Por lo tanto el sistema es controlable.

## Capítulo 2

### 2. Introducción al Control Predictivo

El control predictivo tiene por objetivo resolver de forma efectiva, problemas de control y automatización de procesos que se caracterizan por presentar comportamientos dinámicos complejos, multivariable y/o inestable. El control predictivo permite abordar problemas estocásticos, procesos con retardo de tiempo, multivariados e incluir restricciones.

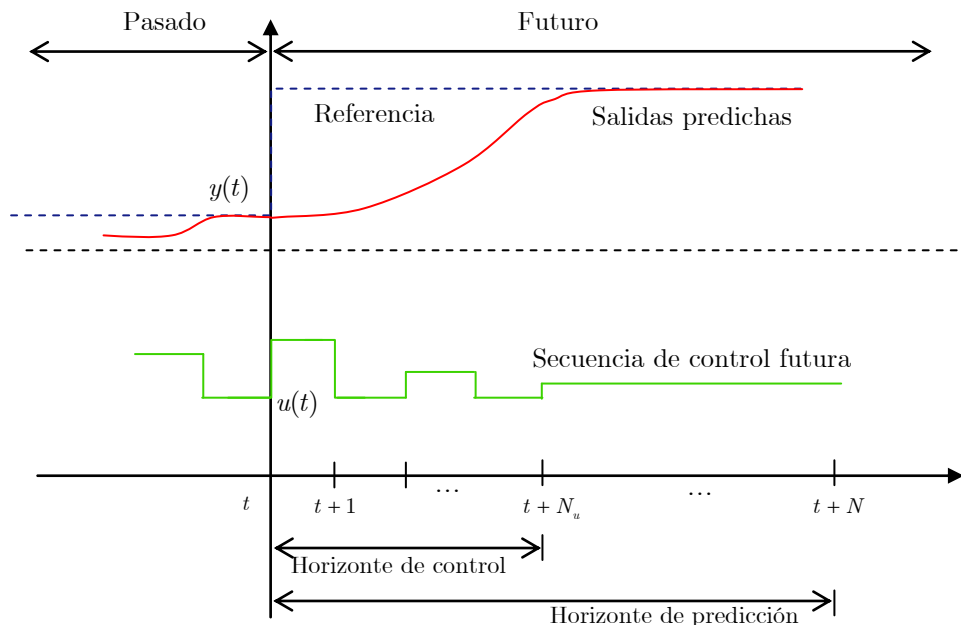
#### 2.1. Control Predictivo basado en modelos (MBPC)

El control predictivo basado en modelos (MBPC o MPC, por su nombre en inglés “Model (Based) Predictive Control”) se presenta actualmente como una atractiva herramienta de control que permite incorporar criterios operacionales a través de la utilización de una función objetivo y restricciones para el cálculo de las acciones de control. Además, estas estrategias de control han alcanzado un nivel muy significativo de aceptabilidad industrial en aplicaciones prácticas de control de procesos. MBPC se basa principalmente en los siguientes elementos:

- El uso de un modelo matemático del proceso que se utiliza para predecir la evolución futura de las variables controladas sobre un horizonte de predicción.
- El establecimiento de una trayectoria deseada futura, o referencia, para las variables controladas.
- El cálculo de las variables manipuladas optimizando una cierta función objetivo o función de costos.
- La aplicación del control siguiendo una política de horizonte móvil.

### 2.1.1. Estrategia el MBPC

La metodología de los controladores MBPC consiste en los siguientes pasos:



**Figura 2.1: Metodología del control predictivo.**

1. Las salidas futuras para un horizonte de predicción  $N$  son predichas en cada instante  $t$ , usando un modelo del proceso. Estas salidas predichas  $\hat{y}(t+j|t)$  dependen de los valores conocidos hasta  $t$  (entradas y salidas pasadas) y además pueden depender de las señales de control futuras  $u(t+j-1|t)$  que se quieren calcular (ver figura 2.1).
2. Las acciones de control futuras  $u(t+j-1|t)$  son calculadas optimizando una función objetivo de manera de llevar la salida del proceso lo más cerca posible de una trayectoria de referencia dada. Este criterio, generalmente es una función cuadrática de los errores entre la salida predicha y la trayectoria de referencia deseada, incluyendo el esfuerzo de control.
3. Sólo se aplica  $u(t|t)$  al proceso, debido a que en el instante siguiente  $t+1$  se tienen dos valores de la variable controlada hasta  $t+1$  y variable manipulada hasta  $t$ .

En síntesis, esta estrategia de control, utiliza el modelo matemático del proceso a controlar para predecir el comportamiento futuro de dicho sistema, y en base a este criterio puede calcular la señal de control futura.

### 2.1.2. Estructura Básica del Control MBPC

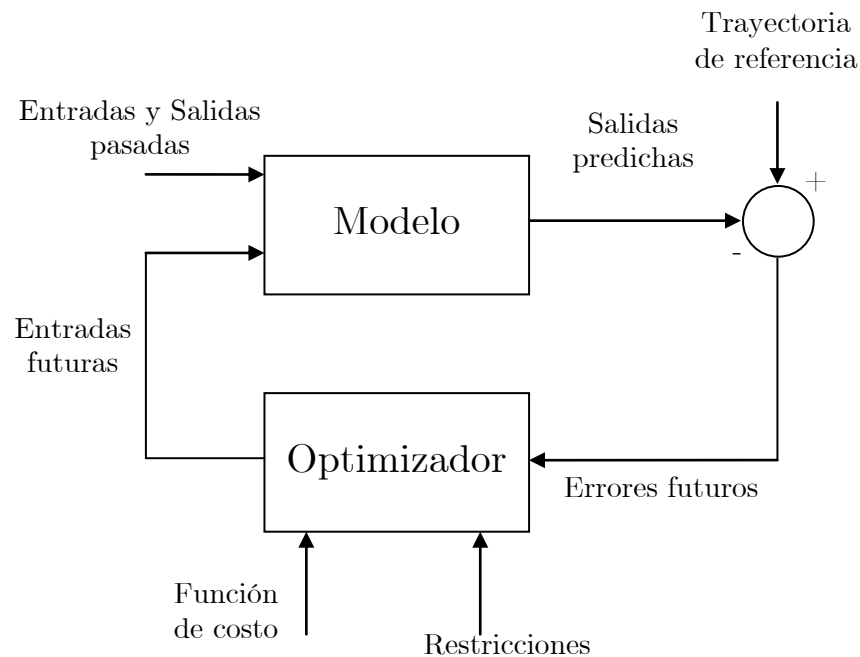


Figura 2.2: Diagrama de bloques de un MBPC

La figura 2.2 muestra la estructura básica de las estrategias de control predictivo basado en modelos. En este caso, se hace uso de un modelo para predecir las salidas futuras del proceso, basándose además en los controles futuros o entradas futuras propuestas. Estas señales son calculadas por un optimizador teniendo en cuenta una función de costo y puede incluir restricciones del proceso.

### 2.1.3. Elementos del MBPC

Los elementos principales del control predictivo son:

- Modelos de predicción
- Función objetivo
- Obtención de la ley de control

#### 2.1.3.1. Modelo de predicción

El modelo de predicción debe ser capaz de capturar la dinámica del proceso para poder predecir las salidas futuras, al mismo tiempo es recomendable que sea sencillo de usar y comprender, y además permitir un análisis teórico.

Las estrategias de MBPC utilizan diferentes modelos del proceso para representar la relación de las salidas con las entradas medibles (variables manipuladas y perturbaciones medibles). Además, se considera un modelo de perturbaciones que intenta incluir las entradas no medibles, el ruido y los errores de modelación.

A continuación se presentarán los principales modelos de proceso y modelos de perturbaciones utilizados para las formulaciones del control predictivo.

##### 2.1.3.1.1. Modelo de respuesta al impulso

La salida del modelo de respuesta al impulso a  $j$ -pasos, está dada por:

$$y(t) = \sum_{i=1}^{\infty} h_i \cdot u(t-i)$$

Donde  $h_i$  son los valores muestreados cuando el proceso es excitado con un impulso unitario (ver figura 2.3).

Entonces, la predicción del modelo está dada por:

$$\hat{y}(t+j|t) = \sum_{i=1}^{\infty} h_i \cdot u(t+j-i|t)$$

Una gran ventaja de este método es que no requiere información previa sobre el proceso, con lo cual el proceso de identificación se simplifica. Además, permite modelar sistemas complejos como fase no mínima y sistemas con retardos. Sin embargo, esta representación sólo es válida para procesos estables.

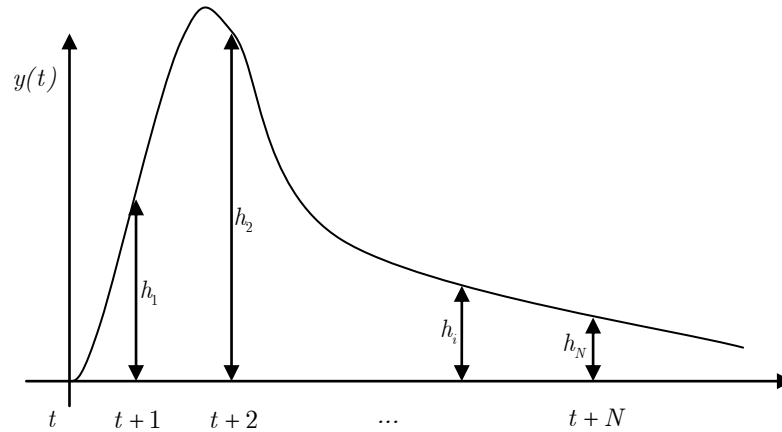


Figura 2.3: Respuesta al impulso de cierto proceso.

#### 2.1.3.1.2. Modelo de respuesta al escalón.

La salida del modelo respuesta al escalón está dada por:

$$y(t) = \sum_{i=1}^{\infty} g_i \cdot \Delta u(t-i)$$

Donde  $g_i$  son los valores muestrados cuando el proceso es excitado con un escalón (ver figura 2.4).

Entonces, la predicción del modelo a  $j$ -pasos, está dada por:

$$\hat{y}(t+j|t) = \sum_{i=1}^{\infty} g_i \cdot \Delta u(t+j-i|t)$$

Este modelo presenta las mismas ventajas e inconvenientes del modelo de respuesta al impulso.

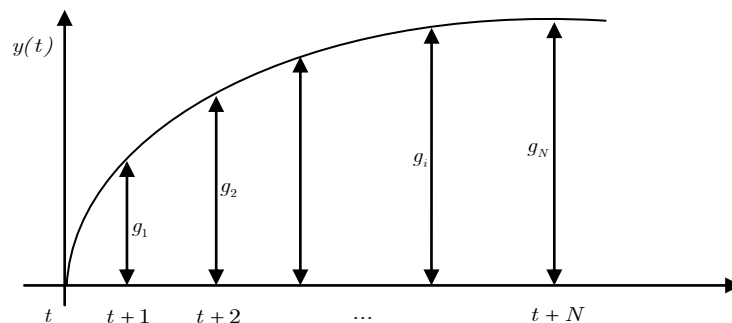


Figura 2.4: Respuesta al escalón de cierto proceso.

### 2.1.3.1.3. Modelo función de transferencia

Este modelo está dado por la siguiente ecuación:

$$A(z^{-1})y(t) = B(z^{-1})u(t)$$

Con los polinomios  $A$  y  $B$  dados por:

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{na} z^{-na}$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb}$$

Entonces, la predicción del modelo es:

$$\hat{y}(t + j | t) = \frac{B(z^{-1})w(t + j | t)}{A(z^{-1})}$$

Esta representación es también válida para procesos inestables.

### 2.1.3.1.4. Modelo en espacio de estados

Las ecuaciones en variables de estados están dadas por:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

Donde  $x(t)$  es un vector de estado, y  $A$ ,  $B$  y  $C$  son las matrices del sistema. Para este modelo la predicción es:

$$\hat{y}(t + j | t) = C \hat{x}(t + j | t) = C \left[ A^j x(t) + \sum_{i=1}^j A^{j-i} B u(t + j - i | t) \right]$$

Esta representación sirve para procesos multivariables. Sin embargo, se hace necesario disponer de un observador cuando los estados no son medibles.

### 2.1.3.1.5. Modelo de perturbaciones

Uno de los modelos más utilizados para representar las perturbaciones, está dado por:

$$n(t) = \frac{C(z^{-1})e(t)}{D(z^{-1})}$$

Donde  $D(z^{-1})$  es un polinomio que incluye al integrador  $\Delta = 1 - z^{-1}$ ,  $e(t)$  es un ruido blanco y generalmente el polinomio  $C$  es igual a 1.



### 2.1.4.1.6. Modelos Estocásticos

Con la representación del modelo de función de transferencia y molde de perturbaciones es posible definir:

**AR** (Auto Regressive)

$$A(z^{-1})y(t) = e(t)$$

**ARX** (Auto Regressive with eXogenous Input)

$$A(z^{-1})y(t) = B(z^{-1})u(t) + e(t)$$

**ARMA** (Auto Regressive Moving Average)

$$A(z^{-1})y(t) = C(z^{-1})e(t)$$

**ARMAX** (Auto Regressive with Moving Average and eXogenous Input)

$$A(z^{-1})y(t) = B(z^{-1})u(t) + C(z^{-1})e(t)$$

**ARIX** (Auto Regressive Integrated with eXogenous Input)

$$A(z^{-1})y(t) = B(z^{-1})u(t) + \frac{e(t)}{\Delta}$$

**ARIMAX** (Auto Regressive Integrated with Moving Average and eXogenous Input)

$$A(z^{-1})y(t) = B(z^{-1})u(t) + \frac{C(z^{-1})e(t)}{\Delta}$$

**ARARX** (Auto Regressive Auto Regressive eXternal Input)

$$A(z^{-1})y(t) = B(z^{-1})u(t) + \frac{e(t)}{D(z^{-1})}$$

**ARARMAX** (Auto Regressive Auto Regressive Moving Average eXternal Input)

$$A(z^{-1})y(t) = B(z^{-1})u(t) + \frac{C(z^{-1})}{D(z^{-1})}e(t)$$

**Output error**

$$y(t) = \frac{B(z^{-1})}{F(z^{-1})}u(t) + e(t)$$

## Box – Jenkins

$$y(t) = \frac{B(z^{-1})}{F(z^{-1})} u(t) + \frac{C(z^{-1})}{D(z^{-1})} e(t)$$

## Caso General

$$A(z^{-1})y(t) = \frac{B(z^{-1})}{F(z^{-1})} u(t) + \frac{C(z^{-1})}{D(z^{-1})} e(t)$$

## NARMAX (Nonlinear ARMAX)

$$y(t) = f(y(t-1), \dots, y(t-n_a), u(t), \dots, u(t-n_b+1), e(t-1), \dots, e(t-n_c)) + e(t)$$

Con  $f$  una función no lineal.

Es importante destacar que los modelos ARIX y ARIMAX son los más utilizados en el control predictivo. En cambio los modelos ARX y ARMAX son más utilizados en diagnóstico de fallas.

### 2.1.3.2. Función objetivo

Los diferentes algoritmos de control predictivo utilizan diferentes funciones objetivo o de costo para la obtención de la ley de control.

En primer lugar se considera la función objetivo está dada por:

$$J = [w(t+1) - \hat{y}(t+1|t)]^2$$

Donde  $w(t+1)$  es la referencia o salida deseada en el instante  $t+1$  y  $\hat{y}(t+1|t)$  es la salida predicha en el instante  $t+1$ . Nótese que utilizando esta función objetivo y un modelo ARIX, se obtiene un controlador de varianza mínima.

Para reducir las variaciones en la variable manipulada o sobreactuaciones se puede utilizar la siguiente función objetivo, que además incluye acción integral.

$$J = [w(t+1) - \hat{y}(t+1|t)]^2 + \lambda [\Delta u(t)]^2$$

A continuación, para considerar sistemas de fase no mínima y sistemas inestables se utiliza la siguiente función objetivo que incluye horizontes de predicción y control mayores:

$$J = \sum_{j=N_1}^{N_2} \delta(j) [w(t+j) - \hat{y}(t+j|t)]^2 + \sum_{i=1}^{N_u} \lambda(i) [\Delta u(t+i-1)]^2$$

Donde

- $\delta(j)$  y  $\lambda(i)$  son coeficientes que ponderan el comportamiento futuro,
- $\hat{y}(t+j|t)$  es la salida predicha en el instante  $t+j$ ,
- $\Delta u(t+i-1)$  es la variación de control en el instante  $t+i-1$ ,
- $w(t+j)$  representa la trayectoria de referencia deseada,
- $N_1$  y  $N_2$  son los horizontes mínimo y máximo de predicción,
- $N_u$  es el horizonte de control

### 2.1.3.3. Obtención de la ley de control

Para obtener los valores  $u(t+i-1|t)$  es necesario minimizar la función de objetivo planteada anteriormente. Para ello se calculan los valores de las salidas predichas  $\hat{y}(t+j|t)$  en función de los valores pasados de las entradas y salidas, y de las señales de control futuras, haciendo uso de un modelo de predicción.

Además se ha encontrado que una estructuración de la ley de control produce una mejora en la robustez del sistema. Esta estructura de la ley de control se basa en el uso del concepto de horizonte de control ( $N_u$ ), que consiste en considerar que tras un cierto intervalo  $N_u < N_2$  no hay variación en las señales de control propuestas, es decir,  $\Delta u(t+i-1) = 0$ , para  $i > N_u$ .

## 2.2. Trayectoria de referencia

Una de las ventajas principales del control predictivo es que la evolución futura de la referencia es conocida a priori. El sistema reaccionará antes que el cambio haya sido efectivamente aplicado, evitando de este modo retardos en la respuesta del proceso.

La evolución futura de la referencia es conocida previamente en muchas aplicaciones, por ejemplo, en robótica.

$$1.- \quad w(t+j) = r(t+j) = cte$$

$$2.- \quad w(t) = y(t)$$

$$w(t+j) = \alpha w(t+j-1) + (1-\alpha)r(t+j)$$

Con  $j = 1, \dots, N$ ;  $0 < \alpha < 1$

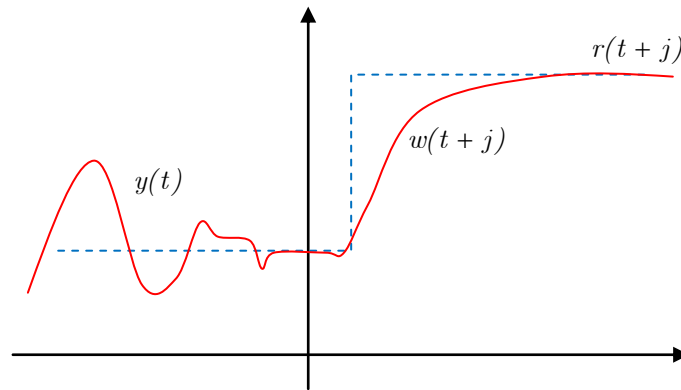


Figura 2.5: Control predictivo con seguimiento de referencia

Cabe destacar que  $\alpha$  es un parámetro ajustable. Dependiendo de éste, la dinámica de la trayectoria es más rápida o más lenta (ver figura 2.5).

Además la trayectoria de  $w(t)$  constituye un filtro de primer orden para  $r(t)$ , este filtro se puede representar mediante la siguiente función de transferencia:

$$\frac{w(z)}{r(z)} = \frac{1-\alpha}{1-\alpha z^{-1}}$$

El filtro para  $r(t)$  se puede implementar fuera del lazo de control, no afectando de esta manera la estabilidad del sistema, pero si contribuyendo al rechazo del ruido.

## 2.3. Predicciones para diferentes tipos de modelos

Para cada modelo se puede establecer una predicción distinta, tal como se explica:

### 2.3.1. Modelo respuesta al escalón

Las perturbaciones se consideran constantes a lo largo del horizonte de predicción e iguales al valor en el instante  $t$ , es decir:

$$\hat{n}(t+j|t) = \hat{n}(t|t) = y_m(t) - \hat{y}(t|t)$$

Donde  $y_m(t)$  son los valores medidos de la salida e  $\hat{y}(t|t)$  es el valor estimado para el modelo en el instante  $t$ .

La predicción está dada por:

$$\hat{y}(t+j|t) = \sum_{i=1}^{\infty} g_i \cdot \Delta u(t+j-i|t) + \hat{n}(t+j|t)$$

Entonces, la predicción está dada por

$$\hat{y}(t+j|t) = \underbrace{\sum_{i=1}^j g_i \Delta u(t+j-i|t)}_{\text{acciones de control futuras}} + \underbrace{\sum_{i=j+1}^{\infty} g_i \Delta u(t+j-i)}_{\text{acciones de control pasadas}} + \hat{n}(t+j|t)$$

Reemplazando la predicción de la perturbación en esta ecuación y el modelo para  $\hat{y}(t|t)$  en ella, se tiene:

$$\begin{aligned} \hat{y}(t+j|t) &= \sum_{i=1}^j g_i \Delta u(t+j-i|t) + \sum_{i=j+1}^{\infty} g_i \Delta u(t+j-i) \\ &\quad + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i) \end{aligned} \quad (**)$$

Entonces, se define la respuesta libre del sistema como los valores conocidos hasta el instante  $t$ , es decir:

$$\begin{aligned} p_j &= \sum_{i=j+1}^{\infty} g_i \Delta u(t+j-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i) \\ p_j &= \sum_{i=1}^{\infty} [g_{j+i} - g_i] \Delta u(t-i) + y_m(t) \end{aligned}$$

Suponiendo un proceso asintóticamente estable,  $g_{j+i} - g_i \approx 0$  para  $i > N$ , se

$$\text{tiene que } p_j = \sum_{i=1}^N [g_{j+i} - g_i] \Delta u(t-i) + y_m(t)$$

Por lo tanto, la ecuación (\*\*) se puede reescribir como:

$$\hat{y}(t+j|t) = \sum_{i=1}^j g_i \Delta u(t+j-i|t) + p_j$$

Entonces, las predicciones en el intervalo  $N_1$  y  $N_2$  están dadas por:

$$\begin{aligned} \hat{y}(t+N_1|t) &= g_1 \Delta u(t+N_1-1|t) + \dots + g_{N_1} \Delta u(t|t) + p_{N_1} \\ \hat{y}(t+N_1+1|t) &= g_1 \Delta u(t+N_1|t) + \dots + g_{N_1+1} \Delta u(t|t) + p_{N_1+1} \\ &\vdots \\ \hat{y}(t+N_2|t) &= g_1 \Delta u(t+N_2-1|t) + \dots + g_{N_2} \Delta u(t|t) + p_{N_2} \end{aligned}$$

Matricialmente, y considerando que  $\Delta u(t+j-1) = 0$  para  $j > N_u$ , se tiene:

$$\hat{y} = G \Delta u + p$$

Donde

$$\hat{y} = \begin{bmatrix} \hat{y}(t+N_1|t) & \hat{y}(t+N_1+1|t) & \dots & \hat{y}(t+N_2|t) \end{bmatrix}^T$$

$$G = \begin{bmatrix} g_{N_1} & \dots & g_1 & 0 & \dots & \dots & 0 \\ g_{N_1+1} & \dots & \dots & g_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ g_{N_2} & \dots & \dots & \dots & \dots & \dots & g_{N_2-N_u+1} \end{bmatrix}$$

$$\Delta u = \begin{bmatrix} \Delta u(t|t) & \Delta u(t+1|t) & \dots & \Delta u(t+N_u-1|t) \end{bmatrix}^T$$

$$p = \begin{bmatrix} p_{N_1}, p_{N_1+1}, \dots, p_{N_2} \end{bmatrix}$$

### 2.3.2. Modelo función de transferencia

- **Modelo ARMAX**

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + n(t)$$

Donde  $n(t) = C(z^{-1})w(t)$ , con  $w_t$  un Ruido Blanco Discreto ( $\bar{w}(t) = 0$  y  $\mathbb{E}(w(t)^2) = \sigma_w^2$ ) y  $d$  es el retardo del sistema.

Sin embargo, esta modelación no considera procesos no estacionarios.

- **Modelo ARIMAX**

Para representar procesos no estacionarios, se considera el siguiente modelo ARIMAX:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + n(t)$$

Donde  $n(t) = \frac{C(z^{-1})w(t)}{\Delta}$ ,  $\Delta = 1 - z^{-1}$

Entonces, se tiene que:

$$\begin{aligned}\Delta n(t) &= C(z^{-1})w(t) = n(t) - n(t-1) \\ n(t) &= n(t-1) + C(z^{-1})w(t)\end{aligned}$$

Sea  $x(t) = C(z^{-1})w(t)$ , entonces,

$$\begin{aligned}n(t) &= n(t-1) + x(t) \\ n(1) &= n_0 + x(1) \\ n(2) &= n_0 + x(1) + x(2) \\ &\vdots \\ n(t) &= n_0 + \sum_{i=1}^t x(i)\end{aligned}$$

Por lo tanto,  $n(t)$  es un sumador y  $n_0$  es la condición inicial.

A continuación, se calcula el valor medio de  $n(t)$ , es decir:

$$\begin{aligned}\bar{n}(t) &= \bar{n}(t-1) + \bar{x}(t) \\ &= \bar{n}_0 + \sum_{i=1}^t \bar{x}(i)\end{aligned}$$

Entonces, se tiene que a menos que  $\bar{x}(i) = \bar{x} = 0$ ,  $n(t)$  no es estacionario aunque  $x$  lo sea.

Así, si  $\bar{x}(i) = \bar{x}$ , entonces  $\bar{n}(t) = \bar{n}_0 + t\bar{x}$ , además si  $C = 1$ ,  $\bar{x}(t) = \bar{w}(t)$  y como  $w(t)$  es Ruido Blanco Discreto se tiene que  $\bar{w}(t) = 0$ , entonces  $\bar{x}(t) = 0$ . Por lo tanto  $\bar{n}(t) = \bar{n}_0$

Sin embargo, el proceso  $n(t)$  no es estacionario en sentido amplio porque  $\sigma_{n(t)}^2$  no es constante como se demuestra a continuación.

$$\begin{aligned}
\sigma_{n(t)}^2 &= \mathbb{E}[n(t) - \bar{n}(t)]^2 \\
&= \mathbb{E}[(n(t-1) + x(t)) - (\bar{n}(t-1) + \bar{x}(t))]^2 \\
&= \mathbb{E}[n(t-1) - \bar{n}(t-1)]^2 + E[x(t) - \bar{x}(t)]^2 - 2\mathbb{E}[(n(t-1) - \bar{n}(t-1))(x(t) - \bar{x}(t))]
\end{aligned}$$

Además,  $x(t) = C(z^{-1})w(t)$

Si  $C(z^{-1}) = 1$  entonces  $C(z^{-1})w_t = x_t = w_t$ ,  $\bar{x}_t = \bar{w}_t = 0$

De esta manera,

$$\mathbb{E}[(n_{t-1} - \bar{n}_{t-1})w_t] = 0$$

pues  $n(t) = n(t-1) + w(t)$  y  $w(t)$  es Ruido Blanco Discreto con  $\bar{w}(t) = 0$

Por lo tanto,

$$\begin{aligned}
\sigma_{n_t}^2 &= \sigma_{n_{t-1}}^2 + \sigma_w^2 \Rightarrow \sigma_{n_t}^2 = \sigma_{n_0}^2 + t\sigma_w^2 \\
\bar{n}_t &= \bar{n}_0
\end{aligned}$$

### 2.3.2.1. Predicción de la salida

Para calcular la predicción a  $j$ -pasos de  $y(t)$ , se considera un modelo ARIMAX, con  $C = 1$  por simplicidad.

$$\begin{aligned}
Ay(t) &= Bu(t-d) + n(t) = Bu(t-d) + \frac{w(t)}{\Delta} \cdot \frac{1}{A} \\
y(t) &= \frac{B}{A} u(t-d) + \frac{w(t)}{A\Delta} \\
\hat{y}(t+j|t) &= \frac{B}{A} u(t-d+j) + \frac{w(t+j)}{A\Delta}
\end{aligned}$$

donde

$$\frac{1}{A\Delta} = \underbrace{E_j}_{\text{Cuociente}} + \underbrace{\frac{F_j}{A\Delta} z^{-j}}_{\text{Resto}}$$

( $F_j$  y  $G_j$  notación en ARMAX)

La división  $\frac{1}{A\Delta}$  se realiza hasta que  $E_j$  sea un polinomio de grado  $j-1$ , de modo que  $E_j w(t+j)$  tenga los valores futuros de  $w(t)$ , es decir,  $w(t+1), w(t+2), \dots, w(t+j)$



Además,

$$A\Delta = 1 - z^{-1} + a_1 z^{-1} - a_1 z^{-2} + \dots$$

entonces  $A\Delta$  es mónico y por ende  $E_j$  también.

En consecuencia,

$$y(t+j) = \frac{B}{A} u(t-d+j) + \underbrace{E_j w(t+j)}_{(w(t+1), w(t+2), \dots)} + \underbrace{\frac{F_j}{A\Delta} w(t)}_{(w(t), w(t-1), \dots)} \quad (\#)$$

Para la determinación de  $w(t), w(t-1), \dots$ , se utiliza el modelo ARIMAX, es decir

$$Ay(t) = Bu(t-d) + \frac{w(t)}{\Delta} \Rightarrow w(t) = A\Delta y(t) - B\Delta u(t-d)$$

Entonces, reemplazando  $w(t)$  en la ecuación (#), se obtiene:

$$\begin{aligned} y(t+j) &= \frac{B}{A} u(t-d+j) + E_j w(t+j) + \frac{F_j}{A\Delta} A\Delta y(t) - \frac{F_j}{A\Delta} B\Delta u(t-d) \\ &= \frac{B}{A} u(t-d+j) + F_j y(t) - F_j \frac{B}{A} u(t-d) + E_j w(t+j) \end{aligned} \quad (j \geq 1)$$

$F_j$  es de grado  $n-1$  ( $n$  orden del sistema) considerando que  $E_j$  tiene grado  $j-1$ .

A continuación,

$$\begin{aligned} y(t+j) &= \frac{B}{A} u(t-d+j) + E_j w(t+j) + F_j y(t) - F_j \frac{B}{A} \underbrace{u(t-d+j)z^{-j}}_{u(t-d)} \\ &= \frac{B}{A} u(t-d+j) [1 - F_j z^{-j}] + E_j w(t+j) + F_j y(t) \end{aligned}$$

donde según la ecuación diofántica  $F_j z^{-j} = 1 - E_j A\Delta$

Entonces,

$$\begin{aligned} y(t+j) &= \frac{B}{A} u(t-d+j) E_j A\Delta + E_j w(t+j) + F_j y(t) \\ &= B E_j \Delta u(t-d+j) + E_j w(t+j) + F_j y(t) \end{aligned}$$

Se define  $G_j \cong B E_j$

$$y(t+j) = G_j \Delta u(t-d+j) + E_j w(t+j) + F_j y(t)$$

Por lo tanto, la predicción a  $j$  pasos de  $y(t)$  está dada por:

$$\boxed{\mathbb{E}[y(t+j)] \equiv \hat{y}(t+j|t) = G_j \Delta u(t-d+j) + F_j y(t)} \quad (*)$$

pues  $E_j w(t+j)$  genera sólo valores futuros de ruido blanco  $(w(t+1), w(t+2), \dots)$ .

### Casos especiales

- $d = 1$

$$\hat{y}(t+j|t) = G_j \Delta u(t+j-1) + F_j y(t)$$

- $j = d$

$$\hat{y}(t+d|t) = G_d \Delta u(t) + F_d y(t)$$

### Valores futuros de $u(t)$ ( $u(t+1), u(t+2), \dots$ )

$$\begin{aligned} G_j \Delta u(t-d+j) &= B E_j (1 - z^{-1}) u(t-d+j) \\ &= b_0 u(t-d+j) + \beta_1 u(t-d+j-1) + \dots \end{aligned}$$

Para  $j \leq d$ , la predicción depende de términos hasta  $t$ , es decir:

$$\begin{aligned} F_j y(t) &= \{y(t), y(t-1), \dots\} \\ G_j \Delta u(t-d+j) &= \{u(t), u(t-1), \dots\} \end{aligned}$$

Para  $j < d$ , la predicción además depende de los valores de control por aplicarse (no conocidos), dados por:

$$G_j \Delta u(t-d+j) = \{u(t+1), u(t+2), \dots, u(t-d+j)\}$$

Entonces,  $G_j \Delta u(t-d+j)$  contiene términos conocidos en  $t$   $\{u(t-1), u(t-2), \dots\}$  y términos desconocidos (o por determinarse)  $\{u(t), u(t+1), u(t+2), \dots\}$

Además, como  $G_j = B E_j$ , entonces

$$G_j(z^{-1}) = g_{j0} + g_{j1} z^{-1} + g_{j2} z^{-2} + \dots$$

### Predicción con modelo ARIMAX con $C \neq 1$

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})\frac{e(t)}{\Delta}$$

donde  $e(t)$  es ruido blanco.

En este caso, la ecuación diofántica está dada por:

$$C(z^{-1}) = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1})$$

donde  $e(t)$  es de grado  $j - 1$  y  $F_j$  es de grado  $na$ .

$$\frac{C(z^{-1})}{A(z^{-1})} = E_j(z^{-1}) + \frac{z^{-j}F_j(z^{-1})}{A(z^{-1})\Delta}$$

A partir del modelo ARIMAX, se tiene:

$$\begin{aligned} y(t+j) &= \frac{B(z^{-1})}{A(z^{-1})} z^{-d} u(t+j-1) + \frac{C(z^{-1})}{A(z^{-1})\Delta} e(t+j) \\ y(t+j) &= \frac{B}{A} u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{A\Delta} e(t) \end{aligned}$$

El ruido  $e(t)$  en instantes previos está dado por:

$$e(t) = \frac{A\Delta y(t) - B\Delta u(t-1-d)}{C(z^{-1})}$$

Entonces, la salida a  $j$  pasos es:

$$\begin{aligned} y(t+j) &= \frac{B}{A} u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{A\Delta} \left( \frac{A\Delta y(t) - B\Delta u(t-1-d)}{C} \right) \\ &= \frac{B}{A} u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{C} y(t) - \frac{B}{AC} F_j u(t-1-d) \\ &= \frac{B}{A} \left[ 1 - \frac{F_j}{C} z^{-j} \right] u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{C} y(t) \\ &= \frac{B}{A} \frac{E_j A \Delta}{C} u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{C} y(t) \\ &= \frac{BE_j}{C} \Delta u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{C} y(t) \\ &= \frac{BE_j}{C} \Delta u(t+j-1-d) + E_j e(t+j) + \frac{F_j}{C} y(t) \end{aligned}$$

Por lo tanto, la predicción de la salida es:

$$\begin{aligned} \hat{y}(t+j|t) &= \mathbb{E}\{y(t+j)\} \\ \Rightarrow \hat{y}(t+j|t) &= \frac{BE_j}{C} \Delta u(t+j-1-d) + \frac{F_j}{C} y(t) \end{aligned}$$

Esta ecuación representa una expresión recursiva para la predicción.

Para separar las acciones de control futuras de las pasadas, se considera la siguiente ecuación diofántica:

$$E_j(z^{-1})B(z^{-1}) = C(z^{-1})H_j(z^{-1}) + z^{-j}K_j(z^{-1})$$

con  $H_j$  de grado  $j - 1$ .

Por lo tanto, la predicción está dada por:

$$\begin{aligned}\hat{y}(t+j|t) &= \frac{CH_j + z^{-j}K_j}{C} \Delta u(t+j-1-d) + \frac{F_j}{C} y(t) \\ \hat{y}(t+j|t) &= H_j \Delta u(t+j-1-d) + \frac{K_j}{C} \Delta u(t-1-d) + \frac{F_j}{C} y(t)\end{aligned}$$

Donde  $H_j \Delta u(t+j-1-d)$  incluye las acciones de control futuras y  $\frac{K_j}{C} \Delta u(t-1-d)$  las acciones de control pasadas.

### Predicción con modelo ARIMAX con $C \neq 1$ (Método 2)

Considerar las siguientes señales filtradas de la planta:

$$y^f(t) = \frac{y(t)}{C(z^{-1})}, \quad u^f(t) = \frac{u(t)}{C(z^{-1})}$$

Se sigue que el nuevo modelo queda dado por:

$$A(z^{-1})y^f(t) = B(z^{-1})z^{-d}u^f(t-1) + \frac{e(t)}{\Delta}$$

La predicción se calcula utilizando el procedimiento para ruido blanco. Luego, la señal obtenida  $\hat{y}^f(t+j)$  es filtrada por  $C(z^{-1})$  para obtener la predicción de la salida original  $\hat{y}(t+j)$ .

### Predicción con modelo ARIMAX con $C \neq 1$ (Método 3)

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \quad (*)$$

donde  $e(t)$  es ruido blanco y  $d = 0$  por simplicidad.

En este caso, la ecuación diofántica está dada por:

$$C(z^{-1}) = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \quad (**)$$

donde  $E_j$  es de grado  $j - 1$  y  $F_j$  es de grado  $na$ .

Multiplicando la ecuación (\*) por  $\Delta E_j(z^{-1})z^j$  y usando la ecuación (\*\*), se tiene:

$$C(z^{-1})(y(t+j) - E_j(z^{-1})e(t+j)) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y(t)$$

Considerando el valor esperado, se tiene:

$$\mathbb{E}\left[C(z^{-1})(y(t+j) - E_j(z^{-1})e(t+j))\right] = C(z^{-1})\hat{y}(t+j|t)$$

Entonces,

$$C(z^{-1})\hat{y}(t+j|t) = E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) + F_j(z^{-1})y(t) \quad (***)$$

Esta predicción es recursiva y para obtener una expresión explícita se utiliza la siguiente ecuación diofántica:

$$1 = C(z^{-1})M_j(z^{-1}) + z^{-j}N_j(z^{-1})$$

Donde el grado de  $M_j$  es  $j-1$  y grado de  $N_j$  es el grado de  $C-1$ .

Multiplicando la ecuación (\*\*\*) por  $M_j(z^{-1})$ , se tiene:

$$\hat{y}(t+j|t) = M_j E_j(z^{-1})B(z^{-1})\Delta u(t+j-1) + M_j(z^{-1})F_j(z^{-1})y(t) + N_j(z^{-1})y(t)$$

Entonces,

$$\begin{aligned} \hat{y}(t+j|t) &= G(z^{-1})\Delta u(t+j-1) + G_p(z^{-1})\Delta u(t+j-1) \\ &\quad + (M_j(z^{-1})F_j(z^{-1}) + N_j(z^{-1}))y(t) \end{aligned}$$

con grado de  $G(z^{-1}) < j$

### 2.3.3. Predicción con modelo en variables de estado

Se considera el siguiente modelo en variables de estado:

$$\begin{aligned} x(t+1) &= Ax(t) + B\Delta u(t) + Gw(t) \\ y(t) &= Dx(t) + w(t) \end{aligned}$$

donde  $w(t)$  es ruido blanco.

El estado a  $j$  -pasos, está dado por:

$$\begin{aligned}
 x(t+2) &= Ax(t+1) + B\Delta u(t+1) + Gw(t+1) \\
 &= A^2x(t) + AB\Delta u(t) + AGw(t) + B\Delta u(t+1) + Gw(t+1) \\
 x(t+3) &= A^3x(t) + A^2B\Delta u(t) + A^2Gw(t) \\
 &\quad + AB\Delta u(t+1) + AGw(t+1) + B\Delta u(t+2) + Gw(t+2) \\
 &\quad \vdots \\
 x(t+j) &= A^jx(t) + \sum_{i=1}^j A^{j-i}B\Delta u(t+i-1) \\
 &\quad + \sum_{i=2}^j A^{j-i}Gw(t+i-1) + A^{j-1}Gw(t)
 \end{aligned}$$

Donde  $\sum_{i=2}^j A^{j-i}Gw(t+i-1)$  agrupa las perturbaciones futuras.

El ruido en instantes siguientes, desde  $t$  hacia atrás es conocido y está dado por:

$$w(t) = y(t) - Dx(t)$$

Por lo tanto, el estado a  $j$  -pasos es:

$$\begin{aligned}
 x(t+j) &= A^jx(t) + \sum_{i=1}^j A^{j-i}B\Delta u(t+i-1) \\
 &\quad + \sum_{i=2}^j A^{j-i}Gw(t+i-1) + A^{j-1}G(y(t) - Dx(t))
 \end{aligned}$$

Entonces, la predicción de la salida es:

$$\hat{y}(t+j|t) = D\hat{x}(t+j|t) + \mathbb{E}(w(t+j|t))$$

donde  $\mathbb{E}(w(t+j|t)) = 0$  pues  $w(t)$  es ruido blanco.

$$\hat{y}(t+j|t) = D \cdot \mathbb{E} \left( A^jx(t) + \sum_{i=1}^j A^{j-i}B\Delta u(t+i-1) + \sum_{i=2}^j A^{j-i}Gw(t+i-1) + A^{j-1}G(y(t) - Dx(t)) \right).$$

Además, se tiene  $\mathbb{E}(x(t|t)) = \hat{x}(t|t)$

Si el estado no es medible se requerirá de un observador de estado. También,

$$\mathbb{E}(w(t+i-1)) = 0 \quad i \geq 2$$

Finalmente, la predicción de la salida es:

$$\hat{y}(t+j|t) = DA^j \hat{x}(t|t) + \sum_{i=1}^j A^{j-i} BD \Delta u(t+i-1) + DA^{j-1} G (y(t) - D \hat{x}(t|t))$$

## 2.4. Estructura de los predictores

Tipo de modelo	Futuro	Pasado
Modelo respuesta al escalón	$\sum_{i=1}^j g_i \cdot \Delta u(t+j-i)$	$\sum_{i=1}^N [g_{j+i} - g_i] \cdot \Delta u(t-i) + y(t)$
Modelo ARIMAX C≠1	$H_j \cdot \Delta u(t+j-i)$	$\frac{K_j}{C} \Delta u(t-1) + \frac{F_j}{C} y(t)$
Modelo en variables de estado	$\sum_{i=1}^j A^{j-i} BD \cdot \Delta u(t+i-1)$	$DA^{j-1} (A - GD) \hat{x}(t t) + DA^{j-1} G y(t)$
	Respuesta forzada $y_f(t+j t)$	Respuesta libre $y_l(t+j t)$

Tabla 2.1: Estructura de los predictores.

Por lo tanto, la predicción se puede escribir de la forma:

$$\hat{y}(t+j|t) = \hat{y}_f(t+j|t) + \hat{y}_l(t+j|t)$$

$\hat{y}_f(t+j|t)$  es la respuesta forzada que depende de las futuras acciones de control  $\hat{y}_l(t+j|t)$  es la respuesta libre, suponiendo que no hay futuras acciones de control y por lo tanto, el control permanece en el valor que tiene en el instante  $t$ ,  $\Delta u(t+i-1) = 0$ ,  $i \geq 1$ .

## 2.5. Control de Varianza Mínima

Se considera el siguiente criterio para determinar la acción de control:

$$\min J = \mathbb{E} \{y(t+d)^2\}$$

Donde  $d$  es el tiempo muerto; el modelo utilizado está dado por:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + C(z^{-1})e(t)$$

Para determinar la predicción  $y(t+d)$ , se utiliza la siguiente ecuación diofántica:

$$C(z^{-1}) = F(z^{-1})A(z^{-1}) + G(z^{-1})z^{-d}$$

$$\text{Entonces } y(t+d) = \frac{B}{A}u(t) + \frac{C}{A}e(t+d)$$

Utilizando la ecuación diofántica, se tiene:

$$\begin{aligned} y(t+d) &= \frac{B}{A}u(t) + Fe(t+d) + \frac{G}{A}z^{-d}e(t+d) \\ &= \frac{B}{A}u(t) + Fe(t+d) + \frac{G}{A}e(t) \end{aligned}$$

Además, se tiene que el ruido  $e(t)$  en instantes anteriores  $t-1, t-2, \dots$  se puede calcular como  $e(t) = \frac{Ay(t) - Bu(t-d)}{C}$

En efecto,

$$\begin{aligned} y(t+d) &= \frac{B}{A}u(t) + Fe(t+d) + \frac{G}{A} \left( \frac{Ay(t) - Bu(t-d)}{C} \right) \\ &= \frac{B}{A}u(t) + Fe(t+d) + \frac{G}{C}y(t) - \frac{GB}{AC}u(t-d) \\ &= \frac{B}{A} \left( 1 - \frac{G}{C}z^{-d} \right) u(t) + Fe(t+d) + \frac{G}{C}y(t) \\ &= \frac{B}{A} \frac{FA}{C} u(t) + Fe(t+d) + \frac{G}{C}y(t) \\ &\Rightarrow y(t+d) = \frac{G}{C}y(t) + \frac{BF}{C}u(t) + Fe(t+d) \end{aligned}$$

Continuando con el problema de varianza mínima, se tiene:

$$\begin{aligned} \min J = \mathbb{E} \{ y(t+d)^2 \} &= \mathbb{E} \left\{ \left( \frac{G}{C}y(t) + \frac{BF}{C}u(t) + Fe(t+d) \right)^2 \right\} \\ &= \mathbb{E} \left\{ \left( \frac{G}{C}y(t) + \frac{BF}{C}u(t) \right)^2 + (Fe(t+d))^2 \right\} \end{aligned}$$



Se sabe que  $y(t), y(t-1), \dots, u(t), u(t-1), \dots$  y  $e(t+d), \dots, e(t+1)$  son independientes entre sí, luego al calcular la esperanza del término cruzado:

$$\begin{aligned} & \mathbb{E} \left\{ 2 \left( \frac{G}{C} y(t) + \frac{BF}{C} u(t) \right) \cdot (Fe(t+d)) \right\} \\ &= \mathbb{E} \left\{ 2 \left( \frac{G}{C} y(t) + \frac{BF}{C} u(t) \right) \cdot (F) \right\} \cdot \mathbb{E} \{ e(t+d) \} = 0 \end{aligned}$$

Pues  $e(t)$  es ruido blanco y su esperanza es cero.

Por lo tanto, la condición necesaria es:

$$\frac{\partial J}{\partial u(t)} = 0 \Rightarrow \mathbb{E} \left\{ 2 \left( \frac{G}{C} y(t) + \frac{BF}{C} u(t) \right) \frac{\partial \left( \frac{BF}{C} u(t) \right)}{\partial u(t)} \right\} = 0$$

donde

$$\frac{B \cdot F}{C} u(t) = h_0 u(t) + h_1 u(t-1) + \dots$$

$$\mathbb{E} \left\{ 2 \left( \frac{G}{C} y(t) + \frac{BF}{C} u(t) \right) h_0 \right\} = 0$$

De donde se llega al controlador optimo, dado por:

$$\boxed{u^*(t) = -\frac{G}{BF} y(t)}$$

### 2.5.1. Varianza Mínima con Minimización de Energía de Control

Se considera el siguiente criterio:

$$\min J = \mathbb{E} \{ (w(t+d) - y(t+d))^2 + \lambda u^2(t) \}$$

con  $w(t+d)$  es la trayectoria de referencia y  $\lambda$  es un factor de ponderación.

El modelo utilizado está dado por:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + e(t)$$

Para determinar la predicción  $y(t+d)$ , se utiliza lo de la sección 2.5 y se tiene que:

$$y(t+d) = BFu(t) + Gy(t) + Fe(t+d)$$

Por lo tanto,

$$\begin{aligned}\min J &= \mathbb{E} \left\{ (w(t+d) - BFu(t) - Gy(t) - Fe(t+d))^2 + \lambda u^2(t) \right\} \\ &= \mathbb{E} \left\{ (w(t+d) - BFu(t) - Gy(t))^2 + (Fe(t+d))^2 + \lambda u^2(t) \right\}\end{aligned}$$

Los términos cruzados se anulan por lo mismo que en la sección 2.5.

La condición necesaria está dada por:

$$\frac{\partial J}{\partial u(t)} = -2\mathbb{E} \left\{ (w(t+d) - B \cdot Fu(t) - Gy(t)) \frac{\partial(B \cdot Fu(t))}{\partial u(t)} - 2\lambda u(t) \right\} = 0$$

Además,

$$B \cdot Fu(t) = (b_0 + b_1 z^{-1} + \dots + b_{nb} z^{-nb}) (1 + f_1 z^{-1} + \dots + f_{d-1} z^{-d+1}) u(t)$$

Entonces,

$$\frac{\partial B \cdot Fu(t)}{\partial u(t)} = b_0$$

Por lo tanto,

$$\mathbb{E} \{ (w(t+d) - B \cdot Fu(t) - Gy(t)) b_0 - \lambda u(t) \} = 0$$

La acción de control óptima está dada por:

$$u^*(t) = \frac{b_0 (w(t+d) - Gy(t))}{BFb_0 + \lambda}$$

## 2.6. Control por Matriz Dinámica (DMC)

Este algoritmo (“Dynamic Matrix Control”) utiliza el modelo repuesta al escalón para la predicción. Asumiendo que el proceso es estable.

$$y(t) = \sum_{i=1}^N g_i \Delta u(t-i)$$

$$\hat{y}(t+j | t) = \sum_{i=1}^N g_i \Delta u(t+j-i) + \hat{n}(t+j | t)$$

Además,

$$\hat{n}(t+j | t) = \hat{n}(t | t) = y_m(t) - \hat{y}(t | t)$$

De la sección 2.3.1, la predicción está dada por:

$$\hat{y}(t+j | t) = \sum_{i=1}^j g_i \Delta u(t+j-i | t) + p_j$$

Con  $p_j = \sum_{i=1}^N [g_{j+i} - g_i] \Delta u(t-i) + y_m(t)$

Entonces, las predicciones en el intervalo  $N_1$  y  $N_2$  están dadas por:

$$\begin{aligned} \hat{y}(t+N_1 | t) &= g_1 \Delta u(t+N_1-1 | t) + \dots + g_{N_1} \Delta u(t | t) + p_{N_1} \\ \hat{y}(t+N_1+1 | t) &= g_1 \Delta u(t+N_1 | t) + \dots + g_{N_1+1} \Delta u(t | t) + p_{N_1+1} \\ &\vdots \\ \hat{y}(t+N_2 | t) &= g_1 \Delta u(t+N_2-1 | t) + \dots + g_{N_2} \Delta u(t | t) + p_{N_2} \end{aligned}$$

Matricialmente, y considerando que  $\Delta u(t+j-1) = 0$  para  $j > N_u$ , se tiene:

$$\hat{y} = G \Delta u + p$$

Donde

$$\hat{y} = [\hat{y}(t+N_1 | t) \quad \hat{y}(t+N_1+1 | t) \quad \dots \quad \hat{y}(t+N_2 | t)]^T$$

$$G = \begin{bmatrix} g_{N_1} & \dots & g_1 & 0 & \dots & \dots & 0 \\ g_{N_1+1} & \dots & \dots & g_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ g_{N_2} & \dots & \dots & \dots & \dots & \dots & g_{N_2-N_u+1} \end{bmatrix}$$

$$\Delta u = [\Delta u(t | t) \quad \Delta u(t+1 | t) \quad \dots \quad \Delta u(t+N_u-1 | t)]^T$$

$$p = [p_{N_1}, p_{N_1+1}, \dots, p_{N_2}]$$

Por otra parte, el controlador por matriz dinámica utiliza la función objetivo general del MBPC con  $\delta = 1$ , la cual matricialmente está dada por:

$$J = [w - \hat{y}]^T [w - \hat{y}] + \lambda \Delta u^T \Delta u$$

Reemplazando en esta ecuación la predicción  $\hat{y}$ , se tiene

$$J = [w - G\Delta u - p]^T [w - G\Delta u - p] + \lambda \Delta u^T \Delta u$$

Minimizando esta función objetivo se obtiene:

$$\Delta u = [G^T G + \lambda I]^{-1} G^T (w - p)$$

donde sólo se aplica  $\Delta u(t|t)$  de acuerdo a la estrategia de control de horizonte móvil definida. Esto es,

$$\Delta u(t|t) = K(w - p)$$

Donde  $K$  es la primera fila de matriz  $[G^T G + \lambda I]^{-1} G$ . (ver problema resuelto 2.4).

## 2.7. Control Predictivo Generalizado (GPC.)

Para el diseño de Controladores Predictivos Generalizados GPC. (“Generalized Predictive Control”) se utiliza el siguiente modelo ARIMAX:

$$A(z^{-1})y(t) = B(z^{-1})u(t-d) + \frac{C(z^{-1})w(t)}{\Delta}$$

Donde  $d$  es el retardo del sistema. Además se supone  $C(z^{-1}) = 1$  por simplicidad.

Por lo anteriormente mencionado en la sección 2.3.2.1 se tiene que la predicción viene dada por:

$$\hat{y}(t+j|t) = \mathbb{E}[y(t+j)|t] = G_j \Delta u(t-d+j) + F_j y(t) \quad (*)$$

con  $G_j = BE_j$  y  $\frac{1}{A\Delta} = E_j + \frac{z^{-j}F_j}{A\Delta}$  (Ecuación diofántica).

**Problema.** Demostrar que  $G_j$  es igual a los primeros  $j$  términos de la respuesta a escalón unitario de la planta  $Ay(t) = Bu(t-d)$

De este modo, como  $G_j$  representa los  $j$  primeros términos de la respuesta escalón, entonces  $g_{ji} = g_i$  para  $i = 1, 2, 3, \dots < j$ . Es decir,

$$\begin{aligned} g_{j0} &= g_{10} = \dots = g_{k0} = \dots = g_0 \\ &\vdots \\ g_{ji} &= g_{ji} = \dots = g_{ki} = \dots = g_i \end{aligned}$$

Por lo tanto, se tiene

$$\begin{aligned} G_1 &= g_0 + g_1 z^{-1} + \dots + g_{n_b} z^{-n_b} \\ G_2 &= g_0 + g_1 z^{-1} + \dots + g_{n_b+1} z^{-(n_b+1)} \\ G_3 &= g_0 + g_1 z^{-1} + \dots + g_{n_b+2} z^{-(n_b+2)} \\ &\vdots \\ G_j &= g_0 + g_1 z^{-1} + \dots + g_{n_b+j-1} z^{-(n_b+j-1)} \end{aligned}$$

Por otra parte,

$$\hat{y}(t + j | t) = G_j \Delta u(t - d + j) + F_j y(t)$$

y si  $d = 1$ ,  $N_1 = 1$  y  $N_2 = N$ , se tiene

$$\begin{aligned} \hat{y}(t + 1 | t) &= G_1 \Delta u(t) + F_1 y(t) \\ \hat{y}(t + 2 | t) &= G_2 \Delta u(t + 1) + F_2 y(t) \\ &\vdots \\ \hat{y}(t + N | t) &= G_N \Delta u(t - 1 + N) + F_N y(t) \end{aligned}$$

A continuación, se agrupan los términos conocidos hasta  $t$  en el vector  $f$   $\{\Delta u(t-1), \Delta u(t-2), \dots, y(t), y(t-1), \dots\}$ , es decir

$$\begin{aligned} f(t+1) &= [G_1(z^{-1}) - g_0] \Delta u(t) + F_1 y(t) \\ f(t+2) &= [G_2(z^{-1}) - g_1 z^{-1} - g_0] \Delta u(t+1) + F_2 y(t) \quad (*) \\ &\vdots \end{aligned}$$

Entonces, la expresión (\*) se puede expresar de manera matricial como

$$\hat{y} = G\tilde{u} + f$$

Donde  $\hat{y} \equiv [\hat{y}(t+1 | t), \hat{y}(t+2 | t), \dots, \hat{y}(t+N | t)]$  son las predicciones desde el horizonte de predicción  $N_1 = 1$  hasta  $N_2 = N$ .

$$G = \begin{bmatrix} g_0 & 0 & \cdots & \cdots & 0 \\ g_1 & g_0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \ddots & & \\ \vdots & \vdots & & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & \cdots & g_0 \end{bmatrix}_{N \times N}$$

$\tilde{u} \equiv [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+2), \dots, \Delta u(t+N-1)]^T$  son las acciones de control futuras. Por lo tanto,  $G\tilde{u}$  contiene términos desconocidos por determinar  $\{\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)\}$  y  $f$  agrupa los términos conocidos  $\{\Delta u(t-1), \Delta u(t-2), \dots, y(t), y(t-1), \dots\}$

Para el cálculo de las acciones de control futuras, el algoritmo de Control Predictivo Generalizado G.P.C. se minimiza la siguiente función de costos:

$$J = \sum_{j=N_1}^{N_2} [\hat{y}(t+j|t) - r(t+j)]^2 + \sum_{i=1}^{N_u} \lambda_i [\Delta u(t+i-1)]^2 \quad (**)$$

En general,

- $N_1$  horizonte predictivo mínimo. ( $N_1 = d$ , retardo, por defecto)
- $N_2$  es del orden del tiempo de estabilización ( $T_s$ ) del sistema y es el horizonte de predicción máximo.
- $N_u$ : hasta el orden de la planta.
- $\lambda$ : Parámetro de ponderación de control.

Por otra parte, para una planta con  $n$  polos se necesitan  $n$  cambios de  $u(t)$  para llegar a valor de  $y(t) = r(t)$  dado, entonces  $\Delta u(t) = 0$  para  $t > N_u$ . Por lo tanto,  $u(t)$  queda fijo para. Además, la función de costo definida anteriormente (ecuación (\*\*)) se puede expresar en forma más compacta por ( $\tilde{u} = \Delta u$ )

$$J = (\hat{y} - r)^T (\hat{y} - r) + \lambda \tilde{u}^T \tilde{u}$$

donde

$$\hat{y} = G\tilde{u} + f$$

$$r = [r(t+1), r(t+2), \dots, r(t+N)]^T \quad (\text{con } N_1 = 1, N_2 = N)$$

$$G = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_u-1} & g_{N_u-2} & \cdots & g_0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_{N-N_u} \end{bmatrix}_{N \times N_u}$$

pues  $\Delta u(t+j) = 0$  para  $j \geq N_u$ .

Entonces,

$$J = (G\tilde{u} + f - r)^T (G\tilde{u} + f - r) + \lambda \tilde{u}^T \tilde{u}$$

Para la minimización de esta función de costos, se tiene  $\frac{\partial J}{\partial \tilde{u}(t)} = 0$

Entonces,

$$\begin{aligned} (G\tilde{u} + f - r)^T G + \lambda \tilde{u}^T &= 0 \quad / \quad ()^T \\ G^T (G\tilde{u} + f - r) + \lambda \tilde{u} &= 0 \\ G^T G\tilde{u} + \lambda \tilde{u} &= G^T (r - f) \end{aligned}$$

Por lo tanto las acciones de control vienen dadas por:

$$\tilde{u} = (G^T G + \lambda I)^{-1} G^T (r - f)$$

De este modo, se tiene

$$\begin{bmatrix} \Delta u_t \\ \Delta u_{t+1} \\ \vdots \\ \Delta u_{t+N_u-1} \end{bmatrix} = \begin{bmatrix} H_1^T \\ H_2^T \\ \vdots \end{bmatrix} (r - f)$$

Sin embargo, sólo se utiliza  $\Delta u(t)$ , dado que en el siguiente instante se desplazan los horizontes de predicción y control, siendo posible calcular una nueva acción de control para ese instante.

La acción de control en  $t$  está dada por:  $\Delta u(t) = H_1^T (r - f)$ , con  $H_1^T$  la primera fila de  $(G^T G + \lambda I)^{-1} G^T$

Notar que:  $\Delta u(t) = u(t) - u(t-1)$

$$\Rightarrow \boxed{u(t) = u(t-1) + H_1^T (r - f)}$$

• **Caso**  $N_u < N_2$

Se considera el caso en que  $N_u < N_2$ ,  $N_1 = 1$ , para la función

$$J = \sum_{j=N_1}^{N_2} [\hat{y}(t+j|t) - r(t+j)]^2 + \sum_{i=1}^{N_u} \lambda_i [\Delta u(t+i-1)]^2$$

Matricialmente se tiene

$$J = (\hat{y} - r)^T (\hat{y} - r) + \lambda \Delta u^T \Delta u$$

Donde  $\hat{y}(t) = G\Delta u + f$

Además

$$\begin{aligned} \hat{y} &= [\hat{y}(t+1) \quad \hat{y}(t+2) \quad \dots \quad \hat{y}(t+N_2)]^T \\ \Delta u &= [\Delta u(t) \quad \Delta u(t+1) \quad \dots \quad \Delta u(t+N_u-1)] \end{aligned}$$

Entonces,

$$J = (G\Delta u + f - r)^T (G\Delta u + f - r) + \lambda \Delta u^T \Delta u$$

Como la dimensión de  $\hat{y} < \Delta u$ , es decir, la multiplicación de  $G\Delta u$  no calza, por lo cual se crea un vector  $\bar{u}$  “ficticio” tal que cumpla con la relación  $\Delta u = M \cdot \bar{u}$ , con

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{N_u \times N_2}, \quad \bar{u} = \begin{pmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N_u-1) \\ \vdots \\ \Delta u(t+N_2-1) \end{pmatrix}_{N_2 \times 1}$$

Donde  $M$  tiene tantos **unos** como las componentes de  $\Delta u$ .

Reemplazando en la función de costos, dada por:

$$J = (G\bar{u} + f - r)^T (G\bar{u} + f - r) + \lambda (M \cdot \bar{u})^T (M \cdot \bar{u})$$

Para la minimización, se tiene  $\frac{\partial J}{\partial \bar{u}} = 0$

Entonces,

$$\begin{aligned} (G\bar{u} + f - r)^T G + \lambda \bar{u}^T \cdot M &= 0 \quad / \quad ()^T \\ G^T (G\bar{u} + f - r) + \lambda M^T \cdot \bar{u} &= 0 \\ G^T G\bar{u} + \lambda M^T \cdot \bar{u} &= G^T (r - f) \end{aligned}$$



Por lo tanto el controlador  $\Delta u$  viene dado por la primera fila de:

$$\bar{u} = (G^T G + \lambda M^T)^{-1} G^T (r - f)$$

### 2.7.1. Control predictivo en variables de estado

Considere que un proceso con  $n$  salidas y  $m$  estados puede ser descrito por el siguiente modelo en variables de estado:

$$\begin{aligned} x(t+1) &= Ax(t) + B\Delta u(t) + Gw(t) \\ y(t) &= Dx(t) + w(t) \end{aligned}$$

De la sección 2.3.3 se tiene que la predicción viene dada por:

$$\hat{y}(t+j|t) = DA^j \hat{x}(t|t) + \sum_{i=1}^j A^{j-i} BD\Delta u(t+i-1) + DA^{j-1}G(y(t) - D\hat{x}(t|t))$$

donde la repuesta libre es:  $DA^{j-1}(A - GD)\hat{x}(t|t) + DA^{j-1}Gy(t)$  y la repuesta

forzada es:  $\sum_{i=1}^j A^{j-i} BD\Delta u(t+i-1)$

Entonces, las predicciones sobre el horizonte de predicción son:

$$\begin{aligned} \begin{bmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \vdots \\ \hat{y}(t+N|t) \end{bmatrix} &= \begin{bmatrix} D \\ DA \\ \vdots \\ DA^{N-1} \end{bmatrix} (A - GD)\hat{x}(t|t) + \begin{bmatrix} D \\ DA \\ \vdots \\ DA^{N-1} \end{bmatrix} Gy(t) \\ &+ \begin{bmatrix} DB & & & \\ DAB & DB & & \\ \vdots & \vdots & \vdots & \ddots \\ DA^{N-1}B & DA^{N-2}B & \cdots & DB \end{bmatrix} \begin{bmatrix} \Delta u(t) \\ \vdots \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix} \end{aligned}$$

Matricialmente, se tiene:

$$\hat{Y} = S_N \Delta U + F$$

donde

- $\hat{Y} = [\hat{y}(t+1|t) \quad \hat{y}(t+2|t) \quad \cdots \quad \hat{y}(t+N|t)]^T$
- $\Delta U = [\Delta u(t) \quad \Delta u(t+1) \quad \cdots \quad \Delta u(t+N-1)]^T$

$$\begin{aligned}
\bullet \quad S_N &= \begin{bmatrix} DB \\ DAB & DB \\ \vdots & \vdots \\ DA^{N-1}B & DA^{N-2}B & \dots & DB \end{bmatrix} \\
\bullet \quad F &= \begin{bmatrix} D \\ DA \\ \vdots \\ DA^{N-1} \end{bmatrix} \left[ (A - GD)\hat{x}(t|t) + Gy(t) \right]
\end{aligned}$$

Matricialmente, la función objetivo del control predictivo se puede expresar como:

$$J = (\hat{Y} - R)^T \Lambda_e (\hat{Y} - R) + \Delta U \Lambda_u \Delta U$$

Minimizando esta función objetivo se obtiene que el vector de acciones futuras está dado por:

$$\Delta U = (S_N^T \Lambda_e S_N + \Lambda_u)^{-1} S_N^T \Lambda_e (R - F)$$

con  $R$  es la matriz de referencia y  $\Lambda_u, \Lambda_e$  son las matrices de ponderación.

## 2.8. Control Predictivo con Restricciones

En la práctica, se requiere mantener las variables del proceso en rangos que aseguren el buen comportamiento de los equipos y evitar situaciones críticas. Por ejemplo, los actuadores tienen restricciones de límite y velocidad, existen restricciones ambientales en los procesos, válvulas, seguridad, etc. Además, los puntos de operación del proceso están determinados por objetivos económicos, que usualmente llevan al proceso a operar cerca de las restricciones.

En especial, los sistemas de control predictivo se anticipan a estas restricciones y corrigen las acciones de control.

### 2.8.1. Tipos de restricciones

- **Restricciones de rango en las variables manipuladas**

$$u_{\min} \leq u(t+i-1) \leq u_{\max} \quad \forall i = 1, \dots, N_u$$

En general, se pueden saturar las variables manipuladas pero la solución obtenida no es óptima con respecto a la función objetivo predictiva. No considerar restricciones en las variables manipuladas, puede provocar que alcance valores muy elevados en la función objetivo, en el peor de los casos.

Violar los límites de las variables manipuladas puede ser muy costoso y peligroso, ya que puede causar daño a los equipos y pérdidas de producción.

En forma matricial, la restricción de rango en las variables manipuladas está dada por:

$$u_{\min} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \leq \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+N_u-1) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u_{\max}$$

$$\Leftrightarrow u_{\min} \mathbf{1} \leq u \leq \mathbf{1} u_{\max}$$

Esta misma restricción en función de  $\Delta u(t)$ , se puede expresar como:

$$\begin{bmatrix} u_{\min} - u(t-1) \\ 1 \\ \vdots \\ 1 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & \\ \vdots & & \ddots \\ 1 & \dots & 1 & 1 \end{bmatrix} \Delta u \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [u_{\max} - u(t-1)]$$

$$\Leftrightarrow [u_{\min} - u(t-1)] \mathbf{1} \leq T \Delta u \leq \mathbf{1} [u_{\max} - u(t-1)]$$

Con  $\Delta u = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u+1)]^T$

- **Restricciones de variación de las variables manipuladas**

Los actuadores generalmente no responden con una rapidez adecuada, por lo cual es usual que se establezcan restricciones sobre las variaciones de las variables manipuladas.

$$\Delta u_{\min} \leq \Delta u(t+i-1) \leq \Delta u_{\max} \quad \forall i = 1, \dots, N_u$$

Matricialmente, se tiene

$$\Delta u_{\min} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \Delta u \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \Delta u_{\max}$$

$$\Delta u_{\min} \mathbf{1} \leq \mathbf{I} \Delta u \leq \mathbf{1} \Delta u_{\max}$$

- **Restricciones de rango en las variables controladas**

En muchos procesos productivos, por problemas de seguridad o costos, se debe limitar las variables controladas a rangos de operación establecidos.

$$y_{\min} \leq \hat{y}(t + j | t) \leq y_{\max} \quad \forall j = 1, \dots, N_2$$

Utilizando el predictor del GPC, se tiene:

$$\hat{y} = G\Delta u + f$$

Con  $\Delta u = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u+1)]^T$  e  $\hat{y} = [\hat{y}(t+1|t), \dots, \hat{y}(t+N_2|t)]^T$  y  $f$  agrupa los términos conocidos hasta  $t$ .

Matricialmente, las restricciones de rango en las variables controladas están dadas por:

$$y_{\min} \mathbf{1} \leq G\Delta u + f \leq \mathbf{1} y_{\max}$$

$$y_{\min} \mathbf{1} - f \leq G\Delta u \leq \mathbf{1} y_{\max} - f$$

- **Restricciones de variación en las variables controladas**

$$\Delta y_{\min} \leq \Delta \hat{y}(t + j | t) \leq \Delta y_{\max} \quad \forall j = 1, \dots, N_2$$

Matricialmente, expresando estas restricciones en función de  $\Delta u$ , se tiene:

$$\begin{aligned} \Delta \hat{y} &= \begin{bmatrix} \Delta \hat{y}(t + 1 | t) \\ \vdots \\ \Delta \hat{y}(t + N_2 | t) \end{bmatrix} = \begin{bmatrix} \hat{y}(t + 1 | t) - \hat{y}(t | t) \\ \vdots \\ \hat{y}(t + N_2 | t) - \hat{y}(t + N_2 - 1 | t) \end{bmatrix} \\ \Delta \hat{y} &= \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ 0 & 0 & \ddots & \\ \vdots & & \ddots & 1 \end{bmatrix} \begin{bmatrix} \hat{y}(t + 1 | t) \\ \vdots \\ \hat{y}(t + N_2 | t) \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} y(t) \\ \Delta \hat{y} &= \mathbf{D} \begin{bmatrix} \hat{y}(t + 1 | t) \\ \vdots \\ \hat{y}(t + N_2 | t) \end{bmatrix} - \mathbf{1}_0 y(t) \\ \hat{y} &= G\Delta u + f \\ \Delta \hat{y} &= \mathbf{D}G\Delta u + \mathbf{D}f - \mathbf{1}_0 y(t) \end{aligned}$$

Entonces,

$$\Delta y_{\min} - \mathbf{D}f + \mathbf{1}_0 y(t) \leq \mathbf{D}G\Delta u \leq \Delta y_{\max} - \mathbf{D}f + \mathbf{1}_0 y(t)$$

- **Restricciones para asegurar el comportamiento monotónico**

Algunos sistemas de control presentan oscilaciones en las variables controladas antes de alcanzar una condición de régimen. En general, estas oscilaciones no son deseables, pues entre otras razones pueden dar origen a perturbaciones sobre otros procesos.

Para evitar el comportamiento monotónico y en caso que  $y(t) \neq w(t)$ , se tiene:

$$\begin{aligned} \hat{y}(t + j | t) &\leq \hat{y}(t + j - 1 | t), & y(t) < w(t) \\ \hat{y}(t + j | t) &\geq \hat{y}(t + j - 1 | t), & y(t) > w(t) \end{aligned}$$

Matricialmente, se tiene:

$$\begin{aligned} G\Delta u + f &\leq \begin{bmatrix} 0 \\ G' \end{bmatrix} \Delta u + \begin{bmatrix} y(t) \\ f' \end{bmatrix}, & y(t) < w(t) \\ G\Delta u + f &\geq \begin{bmatrix} 0 \\ G' \end{bmatrix} \Delta u + \begin{bmatrix} y(t) \\ f' \end{bmatrix}, & y(t) > w(t) \end{aligned}$$

con  $G'$  se compone de las primeras  $N - 1$  filas de  $G$  y  $f'$  se compone de las primeras  $N - 1$  componentes de  $f$ .

- **Restricciones para evitar comportamiento de fase no mínima**

Existen procesos que presentan un comportamiento de fase no mínima, es decir, cuando son excitados responden inicialmente en sentido inverso a como lo hacen finalmente.

Para evitar este comportamiento, que en ciertos procesos es muy perjudicial, se incorporan las siguientes restricciones cuando  $y(t) \neq w(t)$ .

$$\begin{aligned} \hat{y}(t + j \mid t) &\leq y(t), & y(t) > w(t) \\ \hat{y}(t + j \mid t) &\geq y(t), & y(t) < w(t) \end{aligned}$$

Matricialmente,

$$\begin{aligned} G\Delta u + f &\leq \mathbf{1}y(t), & y(t) > w(t) \\ G\Delta u + f &\geq \mathbf{1}y(t), & y(t) < w(t) \end{aligned}$$

### 2.8.2. Consideraciones: Control predictivo con restricciones

1. Si no existen restricciones o si éstas no se consideran, la minimización de la función objetivo genera una solución analítica.
2. Si se consideran restricciones, la solución, en general, se obtiene utilizando un algoritmo numérico de optimización con restricciones. Para esta clase de problemas, en que la función objetivo es cuadrática, se cumple:

- Para que exista solución al problema sin restricciones, la función objetivo debe ser convexa.
- Para que exista solución al problema de optimización con restricciones, debe haber al menos un valor para el cual las variables de optimización cumplan todas las restricciones impuestas.
- Para asegurar la existencia de una solución única, el espacio de restricciones debe ser convexo.

## 2.9. Algoritmo de Programación Cuadrática

La implementación de GPC con restricciones requiere la solución de un problema de programación cuadrática.

$$\begin{aligned} \min J &= (G\Delta + f - r)^T (G\Delta + f - r) + \lambda \Delta u^T \Delta u \\ \text{s.a.} \quad A\Delta u &\leq b \end{aligned}$$

Donde  $A$  y  $b$  se explicaron anteriormente.

**Programación cuadrática (QP): Función objetivo cuadrática y restricciones lineales.**

- Restricciones de igualdad

$$\begin{aligned} \min J &= \frac{1}{2} \Delta u^T H \Delta u + b^T \Delta u + f_0 \\ \text{s.a.} \quad A\Delta u &= a \end{aligned}$$

con  $A$  matriz de  $m \times n$ . Si  $A\Delta u = a$ , entonces:

$$\Delta u = Ya + Zv$$

donde  $Y, Z$  son matrices de  $n \times m$  y  $n \times (n - m)$  respectivamente tales que

$$AY = I, \quad AZ = 0$$

Entonces, la función objetivo queda expresada como:

$$\begin{aligned} J(v) &= \frac{1}{2} [Ya + Zv]^T H [Ya + Zv] + b^T [Ya + Zv] + f_0 \\ J(v) &= \frac{1}{2} v^T Z^T H \cdot Zv + [b^T + a^T Y^T] Zv + \left[ b^T + \frac{1}{2} a^T Y^T H \right] Ya + f_0 \end{aligned}$$

Entonces, el óptimo global se encuentra resolviendo:

$$Z^T H Z v = -Z^T [b + H Y a]$$

Nótese que  $\Delta u^k$  es un punto que satisface la restricción  $A\Delta u^k = a$ . Cualquier otro punto  $u$  que satisface la restricción se puede expresar como:

$$\begin{aligned} \Delta u &= \Delta u^k + Zv \\ A\Delta u &= A\Delta u^k + AZv = A\Delta u^k \end{aligned}$$

Entonces, el vector  $v$  se puede expresar como la solución de:

$$Z^T H Z v = -Z^T g(\Delta u^k)$$

con  $g(\Delta u^k) = H\Delta u^k + b$ , que es el gradiente de  $J(\Delta u)$  en  $\Delta u^k$ . Obtenido  $v$ , se encuentra

$$\Delta u^* = Ya + Zv$$



## 2.10. Problemas Resueltos

### Problema 2.1

Considerar el siguiente modelo ARIMAX

$$y(t) = 1.7y(t-1) - 0.7y(t-2) + 0.9\Delta u(t-1) - 0.6\Delta u(t-2) + w(t)$$

donde  $w(t)$  es ruido blanco. Calcular las predicciones  $\hat{y}(t+1), \hat{y}(t+2), \hat{y}(t+3)$

### Solución

Se debe notar que para el modelo quede como un ARIMAX, se le debe restar a  $y(t-1)$  toda la ecuación, es decir,

$$y(t) = 1.7y(t-1) - 0.7y(t-2) + 0.9\Delta u(t-1) - 0.6\Delta u(t-2) + w(t) \quad / \quad -y(t-1)$$

$$y(t) - y(t-1) = 1.7y(t-1) - y(t-1) - 0.7y(t-2) + 0.9\Delta u(t-1) - 0.6\Delta u(t-2) + w(t)$$

$$\Delta y(t) = 0.7y(t-1) - 0.7y(t-2) + 0.9\Delta u(t-1) - 0.6\Delta u(t-2) + w(t)$$

En efecto, para formar el modelo ARIMAX, se hace lo siguiente:

$$\Delta y(t) = 0.7\Delta y(t-1) + 0.9\Delta u(t-1) - 0.6\Delta u(t-2) + w(t) \quad / \quad \frac{1}{\Delta}$$

$$y(t) = 0.7y(t-1) + 0.9u(t-1) - 0.6u(t-2) + \frac{w(t)}{\Delta}$$

Donde se concluye que:

$$A(z^{-1}) = 1 - 0.7z^{-1}, \quad B(z^{-1}) = 0.9 - 0.6z^{-1} \quad \text{y} \quad d = 1$$

En primer lugar, se calcularan las salidas predichas por sustitución.

$$\hat{y}(t+1|t) = 1.7y(t) - 0.7y(t-1) + 0.9\Delta u(t) - 0.6\Delta u(t-1)$$

$$\begin{aligned} \hat{y}(t+2|t) &= 1.7\hat{y}(t+1|t) - 0.7y(t) + 0.9\Delta u(t+1) - 0.6\Delta u(t) \\ &= 2.19y(t) - 1.19y(t-1) + 0.9\Delta u(t+1) + 0.93\Delta u(t) - 1.02\Delta u(t-1) \end{aligned}$$

$$\begin{aligned} \hat{y}(t+3|t) &= 1.7\hat{y}(t+2|t) - 0.7\hat{y}(t+1|t) + 0.9\Delta u(t+2) - 0.6\Delta u(t+1) \\ &= 2.533y(t) - 1.533y(t-1) + 0.9\Delta u(t+2) + 0.93\Delta u(t+1) \\ &\quad + 0.951\Delta u(t) - 1.314\Delta u(t-1) \end{aligned}$$

Por otra parte, utilizando la ecuación diofántica también se pueden predecir las salidas. Para  $j = 1$ , se tiene

$$1 = E_1 A \Delta + z^{-1} F_1$$

donde  $E_1 = 1$  y  $F_1 = f_{10} + f_{11}z^{-1}$ . Entonces igualando polinomios

$$1 = 1(1 - 0.7z^{-1})(1 - z^{-1}) + z^{-1}(f_{10} + f_{11}z^{-1})$$

Se obtiene que:  $E_1 = 1$  y  $F_1 = 1.7 - 0.7z^{-1}$

Similarmente, se obtiene

$$\begin{aligned} E_2 &= 1 + 1.7z^{-1} \text{ y } F_2 = 2.19 - 1.19z^{-1} \\ E_3 &= 1 + 1.7z^{-1} + 2.19z^{-2} \text{ y } F_3 = 2.533 - 1.533z^{-1} \end{aligned}$$

Se debe mencionar que el polinomio  $E_j$  repite sus coeficientes, agregando uno nuevo en cada predicción.

Entonces,

$$\begin{aligned} G_1 &= E_1 B = 0.9 - 0.6z^{-1} \\ G_2 &= E_2 B = 0.9 + 0.93z^{-1} - 1.02z^{-2} \\ G_3 &= E_3 B = 0.9 + 0.93z^{-1} + 0.951z^{-2} - 1.314z^{-3} \end{aligned}$$

Así, como la predicción está dada por  $\hat{y}(t + j | t) = G_j \Delta u(t - 1 + j) + F_j y(t)$ , se tiene que:

$$\begin{aligned} \hat{y}(t + 1 | t) &= (0.9 - 0.6z^{-1}) \Delta u(t) + (1.7 - 0.7z^{-1}) y(t) \\ \hat{y}(t + 2 | t) &= (0.9 + 0.93z^{-1} - 1.02z^{-2}) \Delta u(t + 1) + (2.19 - 1.19z^{-1}) y(t) \\ \hat{y}(t + 3 | t) &= (0.9 + 0.93z^{-1} + 0.951z^{-2} - 1.314z^{-3}) \Delta u(t + 2) + (2.533 - 1.533z^{-1}) y(t) \end{aligned}$$

Además, en forma vectorial, la predicción está dada por:

$$\underbrace{\begin{bmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \hat{y}(t+3|t) \end{bmatrix}}_{\hat{\mathbf{y}}} = \underbrace{\begin{bmatrix} 0.9 & 0 & 0 \\ 0.93 & 0.9 & 0 \\ 0.951 & 0.93 & 0.9 \end{bmatrix}}_G \underbrace{\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \end{bmatrix}}_{\tilde{\mathbf{u}}} + \underbrace{\begin{bmatrix} 1.7y(t) - 0.7y(t-1) - 0.6\Delta u(t-1) \\ 2.19y(t) - 1.19y(t-1) - 1.02\Delta u(t-1) \\ 2.533y(t) - 1.533y(t-1) - 1.314\Delta u(t-1) \end{bmatrix}}_f$$

### Problema 2.3

Considera un calentador de agua donde el agua fría es calentada por medio de un quemador de gas (ver figura 2.6).

La temperatura del agua que sale depende de la energía agregada al agua, es decir, cuanto gas se quema. Para controlar la temperatura existe una válvula, la cual manipula el gas que fluye al calentador. Se quiere diseñar un controlador por DMC, para regular la temperatura del agua que sale en el calentador de agua.

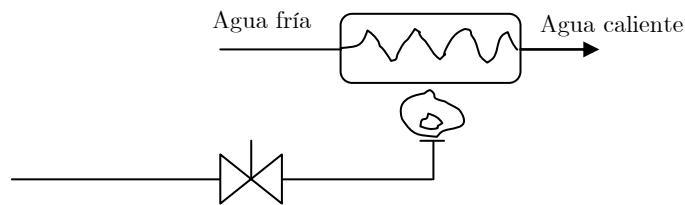


Figura 2.6: Calentador de agua.

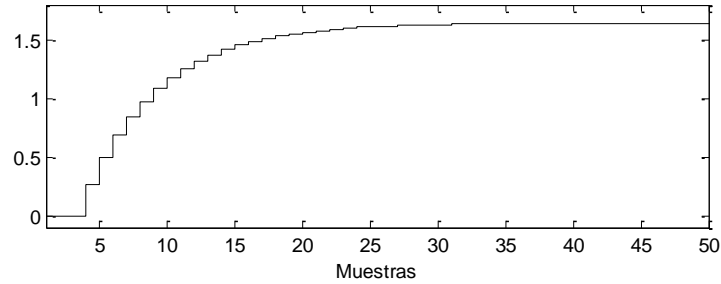
La función de transferencia discreta (con tiempo de muestreo  $T=1[s]$ ) que relaciona la válvula con la salida del sistema está dada por:

$$G(z) = \frac{0.2713z^{-3}}{1 - 0.8351z^{-1}}$$

### Solución

Dado que es un diseño de DMC, se calcula la respuesta al escalón del sistema como se muestra en la figura 2.7, mediante el código en matlab:

```
num=[0 0 0 0.2713];
den=[1 -0.8351 0 0];
sys=tf(num,den, 1, 'Variable','z^-1')
step(sys)
xlabel('muestras')
```



**Figura 2.7: Respuesta al escalón del sistema.**

Es fácil ver en la figura 2.7, que la salida del sistema se estabiliza después de los 30 periodos, así que el modelo de la respuesta al escalón está dado por:

$$y(t) = \sum_{i=1}^{30} g_i \cdot \Delta u(t-i)$$

Los coeficientes  $g_i$  se muestran a continuación:

$g_1 : 0$	$g_2 : 0$	$g_3 : 0.271$	$g_4 : 0.498$	$g_5 : 0.687$	$g_6 : 0.84$
$g_7 : 0.977$	$g_8 : 1.087$	$g_9 : 1.179$	$g_{10} : 1.256$	$g_{11} : 1.32$	$g_{12} : 1.74$
$g_{13} : 1.42$	$g_{14} : 1.456$	$g_{15} : 1.487$	$g_{16} : 1.513$	$g_{17} : 1.535$	$g_{18} : 1.553$
$g_{19} : 1.565$	$g_{20} : 1.581$	$g_{21} : 1.59$	$g_{22} : 1.6$	$g_{23} : 1.608$	$g_{24} : 1.614$
$g_{25} : 1.619$	$g_{26} : 1.623$	$g_{27} : 1.627$	$g_{28} : 1.63$	$g_{29} : 1.633$	$g_{30} : 1.635$

Cabe notar que los coeficientes son obtenidos de la respuesta grafica del sistema. Otra forma de obtener los coeficientes es mediante los coeficientes de la función de transferencia dada por la expresión:

$$g_j = -\sum_{i=1}^j a_i g_{j-i} + \sum_{i=0}^{j-1} b_i, \quad g_k = 0, \quad k \leq 0$$

Donde los coeficientes  $a_i, b_i$  son los coeficientes del denominador y numerador respectivamente (comenzando en  $i = 0$ ).

En este ejemplo los primeros dos coeficientes de la respuesta al escalón son cero, eso se debe a que el sistema tiene dos periodos de muestreo que corresponden al retardo del sistema.

Considerando un horizonte de predicción de 10 y un horizonte de control de 5, la matriz dinámica viene dada por:

$$G = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.271 & 0 & 0 & 0 & 0 \\ 0.498 & 0.271 & 0 & 0 & 0 \\ 0.678 & 0.498 & 0.271 & 0 & 0 \\ 0.84 & 0.678 & 0.498 & 0.271 & 0 \\ 0.977 & 0.84 & 0.678 & 0.498 & 0.271 \\ 1.087 & 0.977 & 0.84 & 0.678 & 0.498 \\ 1.179 & 1.087 & 0.977 & 0.84 & 0.678 \\ 1.256 & 1.179 & 1.087 & 0.977 & 0.84 \end{bmatrix}$$

Considerando  $\lambda = 1$ , la matriz  $[G^T G + \lambda I]^{-1} G^T$  es calculada y la ley de control está dada por la primera fila de esta matriz ( $K$ ) multiplicada por la diferencia entre la trayectoria y la respuesta libre.

$$\Delta u(t) = K(w - f)$$

$$K = [0 \quad 0 \quad 0.1465 \quad 0.1836 \quad 0.164 \quad 0.1224 \quad 0.078 \quad 0.041 \quad 0.01 \quad -0.0157]$$

% Código Utilizado para el cálculo de parámetros

```
a1=den(2); a2=den(3); a3=den(4);
b1=num(2); b2=num(3); b3=num(4);
```

```
% Horizontes de control
N2 =10;
```

```
% Cálculo de g_i
g(1) = b1;
g(2) = - a1*b1 + b1 + b2;
g(3) = - a1*g(2) - a2*g(1) + b1 + b2 + b3;
```

```
for k=4:N2
```

```
    g(k) = - a1*g(k-1) - a2*g(k-2) - a3*g(k-2) + b1 + b2 + b3;
```

```
end
```

```
% Matriz Dinámica
```

```
G=[g(1) 0 0 0 0
    g(2) g(1) 0 0 0
    g(3) g(2) g(1) 0 0
    g(4) g(3) g(2) g(1) 0
    g(5) g(4) g(3) g(2) g(1)
    g(6) g(5) g(4) g(3) g(2)
    g(7) g(6) g(5) g(4) g(3)
    g(8) g(7) g(6) g(5) g(4)
    g(9) g(8) g(7) g(6) g(5)
    g(10) g(9) g(8) g(7) g(6)];
```

### Problema 2.3

Considere un proceso descrito por:

$$y(t) = -ay(t-1) + bu(t-1) + e(t)$$

Con  $e(t)$  ruido blanco de varianza unitaria.

- Encontrar el controlador de varianza mínima con control de energía.
- Resolver el problema de a) considerando que la variable manipulada esta en un rango entre 0 y 1 y el límite máximo de la variable controlada es 2 (suponer  $a, b, \lambda > 0$  y referencia positiva).
- Resolver el problema dado por:

$$\begin{aligned} \min J &= \sum_{j=1}^{10} (\hat{y}(t+j) - r)^2 + \lambda u(t+j-1)^2 \\ \text{s.a.} \quad &0 \leq u(t+i-1) \leq 1, \quad i = 1, \dots, 10 \end{aligned}$$

### Solución

- Propuesto, el controlador viene dado por  $u^*(t) = \frac{b(a \cdot y(t) + r)}{b^2 + \lambda}$
- Se debe resolver el problema dado por:

$$\begin{aligned} \min J &= (\hat{y}(t+1) - r)^2 + \lambda u^2(t) \\ \text{s.a.} \quad &\hat{y}(t+1) = -ay(t) + bu(t) \\ &0 \leq u(t) \leq 1 \\ &y(t) \leq 2 \end{aligned}$$

Aplicando las condiciones de KKT, dado que  $J$  es convexa, se tiene que el problema queda dado por:

$$\begin{aligned} \min J &= (-ay(t) + bu(t) - r)^2 + \lambda u^2(t) \\ \text{s.a.} \quad &-u(t) \leq 0 \\ &u(t) - 1 \leq 0 \\ &y(t) - 2 \leq 0 \end{aligned}$$

Resolviendo para  $u$ , la función a considerar:

$$\mathbb{L} = (-ay(t) + bu(t) - r)^2 + \lambda u^2(t) + \beta_1(-u(t)) + \beta_2(u(t) - 1) + \beta_3(y(t) - 2)$$

Resolviendo el Lagrangeano y aplicando las condiciones de complementariedad y no negatividad:

$$\frac{\partial \mathbb{L}}{\partial u(t)} = 2(-ay(t) + bu(t) - r) \cdot b + 2\lambda u(t) - \beta_1 + \beta_2$$

$$\beta_1(-u(t)) = 0, \quad \beta_1 \geq 0$$

$$\beta_2(u(t) - 1) = 0, \quad \beta_2 \geq 0$$

$$\beta_3(y(t) - 2) = 0, \quad \beta_3 \geq 0$$

- Si  $\beta_1 = \beta_2 = \beta_3 = 0$ , se tiene que el control no está limitado por restricciones, por lo cual esta solución no tiene sentido analizarla.
- Si  $\beta_1 = \beta_2 \neq 0$ ,  $\beta_3 = 0$  se tiene que  $u(t) = 0$  y  $u(t) = 1$ , contradicción; análogo al caso  $\beta_1 = \beta_2 = \beta_3 \neq 0$
- Si  $\beta_1 = 0$ ,  $\beta_2 = \beta_3 \neq 0$  se tiene que el controlador  $u(t) = 1$  e  $y(t) = 2$ , por lo tanto se tiene que:

$$2(-ay(t) + bu(t) - r) \cdot b + 2\lambda u(t) - \beta_1 + \beta_2 = 0$$

$$2(-a \cdot 2 + b - r) \cdot b + 2\lambda + \beta_2 = 0, \quad \beta_2 \geq 0$$

$$2(-2 \cdot a + b - r) \cdot b + 2\lambda \geq 0$$

$$-2ab - br \geq -b^2 - \lambda$$

$$2ab + br \leq b^2 + \lambda$$

Lo anterior es equivalente

$$\left( \frac{b(a \cdot y(t) + r)}{b^2 + \lambda} \leq \right) \underbrace{\frac{b(a \cdot 2 + r)}{b^2 + \lambda}}_{u^*(t)} \leq 1$$

Por lo tanto cumple con la restricción, posible solución del problema y el controlador viene dado por  $u^*(t) = 1$ .

- Si  $\beta_2 = 0$ ,  $\beta_1 = \beta_3 \neq 0$  se tiene que el controlador  $u(t) = 0$  e  $y(t) = 2$ , por lo tanto se tiene que:

$$2(-ay(t) + bu(t) - r) \cdot b + 2\lambda u(t) - \beta_1 + \beta_2 = 0$$

$$2(-a \cdot 2 - r) \cdot b - \beta_1 = 0, \quad \beta_1 \geq 0$$

$$\beta_1 = 2(-a \cdot 2 - r) \geq 0$$

$$2(a \cdot 2 + r) \leq 0$$

Resolviendo la primera ecuación,

$$u^*(t) = \frac{b(a \cdot y(t) + r)}{b^2 + \lambda} = \frac{b(a \cdot 2 + r)}{b^2 + \lambda} \leq 0$$

Por lo tanto el óptimo viene dado por  $u^*(t) = 0$  (ya que  $b, \lambda > 0$ )

- Analizar los demás casos y comprobar que el mínimo de la función objetivo viene dado para  $u^*(t) = 1$  e  $y^*(t) < 2$ .

c) Las variables de optimización vienen dadas por

$$\hat{y}(t+j), u(t+j-1) \quad j = 1, \dots, 10$$

Aplicando el método de programación cuadrática:

$$\begin{aligned} \min J &= x^T H x + p^T x + f^T f \\ \text{s.a.} \quad &Ax = b \end{aligned}$$

Donde

$$x = \begin{bmatrix} \hat{y}(t+1) \\ \vdots \\ \hat{y}(t+10) \\ u(t) \\ \vdots \\ u(t+9) \end{bmatrix}, \quad H = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & \lambda & & \\ & & & & \ddots & \\ & & & & & \lambda \end{bmatrix}, \quad p = \begin{bmatrix} -2r \\ \vdots \\ -2r \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad f = \begin{bmatrix} r \\ \vdots \\ r \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Si suponemos  $Ax = b \Leftrightarrow x_{\inf} \leq x \leq x_{\sup}$

Para calcular las predicciones, se hace mediante recursividad:

$$\begin{aligned} \hat{y}(t+1) &= -ay(t) + bu(t) \\ &\vdots \\ \hat{y}(t+10) &= -a\hat{y}(t+9) + bu(t+9) \end{aligned}$$

$$A = \begin{bmatrix} 1 & & & & & & & & & & \\ & a+1 & & & & & & & & & \\ & & \ddots & & & & & & & & \\ & & & a+1 & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & a+1 & & & & & \\ & & & & & & -b & & & & \\ & & & & & & & \ddots & & & \\ & & & & & & & & \ddots & & \\ & & & & & & & & & -b & \end{bmatrix}, \quad b = \begin{bmatrix} ay(t) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \quad x_{\inf} = \begin{bmatrix} -\infty \\ \vdots \\ -\infty \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad x_{\sup} = \begin{bmatrix} \infty \\ \vdots \\ \infty \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Los valores de  $x$  son infinitos, ya que no hay restricción sobre  $y$ .

Propuesto, resolver el problema en algún programa computacional para algún valor de  $a$  y  $b$ .



## Capítulo 3

### 3. Fundamentos de la Lógica Difusa

La lógica difusa o borrosa (fuzzy logic en inglés), como su nombre indica, es una lógica alternativa a la lógica clásica que pretende introducir un grado de incertidumbre en las cosas que califica. El razonamiento y pensamiento humano frecuentemente conlleva información de este tipo, probablemente originada de la inexactitud inherente de los conceptos humanos y del razonamiento basado en experiencias similares pero no idénticas a experiencias anteriores. En 1964 Zadeh propone por primera vez la noción de conjuntos difusos en un memorando interno de investigación.

Los elementos de un conjunto difuso son pares ordenados que indican el valor del elemento y su grado de pertenencia. Para un conjunto difuso  $A = \{(x, \mu_A(x)) / x \in X\}$ , se tiene que el elemento  $x$  pertenece al conjunto  $A$  con un grado de pertenencia  $\mu_A(x) : X \rightarrow [0,1]$ . Por lo tanto, una variable puede ser caracterizada por diferentes valores lingüísticos, cada uno de los cuales representa un conjunto difuso. Por ejemplo, la velocidad puede ser caracterizada por valores lingüísticos como "Bajo", "Medio" y "Alto", que representan "una velocidad aproximadamente menor que 40 km/h", "una velocidad cercana a 55 km/h" y "una velocidad sobre 70 km/h aprox." respectivamente. Estos términos se asocian a conjuntos difusos con funciones de pertenencia como las mostradas en la figura 3.1.

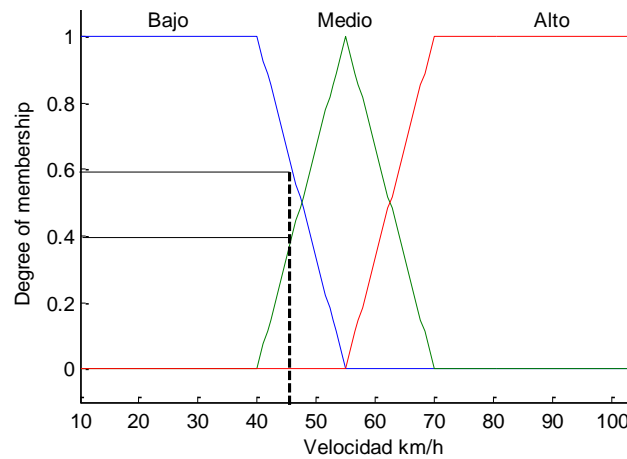


Figura 3.1: Ejemplo de un conjunto difuso.

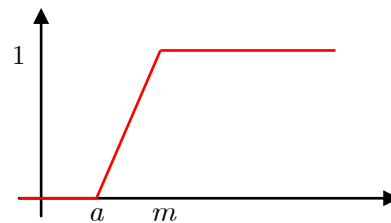
Por lo tanto, si la velocidad es 45 km/h, existen grados de pertenencia 0.6, 0.3 y 0 a los conjuntos difusos "Bajo", "Medio" y "Alto" respectivamente.

### 3.1. Funciones de Pertenencia

Aunque en principio cualquier función sería válida para definir conjuntos difusos, en la práctica hay ciertas funciones típicas que siempre se suelen usar, tanto por la facilidad de computación que su uso conlleva como por su estructura lógica para definir su valor lingüístico asociado. Las funciones más comunes son:

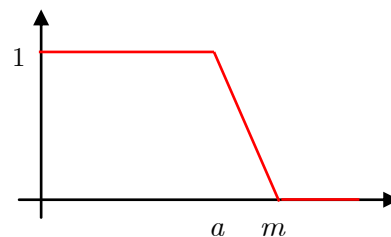
- Función Gamma ( $\Gamma$ )

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a < x < m \\ 1 & x \geq m \end{cases}$$



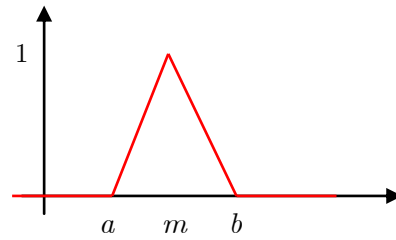
- Función L: se puede definir como

$$\mu(x) = 1 - \Gamma$$



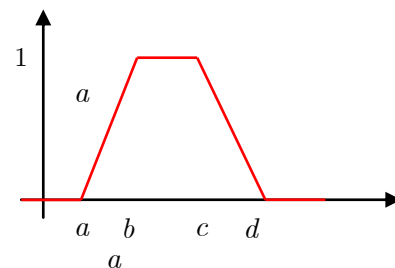
- Función Lambda o Triangular

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a < x \leq m \\ \frac{b-x}{b-m} & m < x \leq b \\ 0 & x > b \end{cases}$$



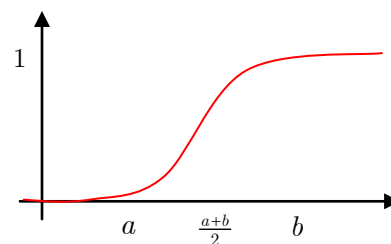
- Función PI o Trapezoidal

$$\mu(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a < x \leq b \\ 1 & b < x \leq c \\ \frac{d-x}{d-c} & c < x \leq d \\ 0 & x > d \end{cases}$$



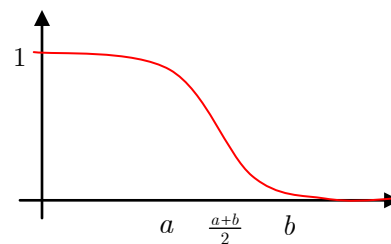
- Función S

$$\mu_s(x) = \begin{cases} 0 & x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2 & a < x \leq \frac{a+b}{2} \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2 & \frac{a+b}{2} < x \leq b \\ 1 & x > b \end{cases}$$



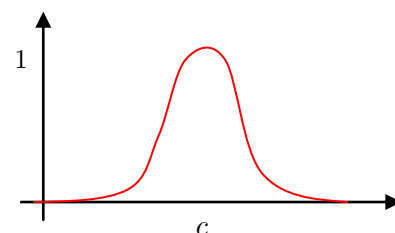
- Función Z

$$\mu_z(x) = 1 - \mu_s(x)$$



- Función Gaussiana

$$\mu(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$



### 3.2. Operaciones básicas de lógica difusa

Dados dos conjuntos difusos  $A$  y  $B$  en el mismo universo  $X$ , con funciones de pertenencia  $\mu_A$  y  $\mu_B$  respectivamente, se pueden definir las siguientes operaciones básicas:

**Unión:** La función de pertenencia de la unión de  $A$  y  $B$  se define como:

$$\mu_{A \cup B} = \max \{(\mu_A(x), \mu_B(x))\}$$

**Intersección:** La función de pertenencia de la intersección de  $A$  y  $B$  es:

$$\mu_{A \cap B} = \min \{(\mu_A(x), \mu_B(x))\}$$

**Complemento:** La función de pertenencia del complemento de  $A$  se define como:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

**Producto cartesiano:** Dados los conjuntos difusos  $A_1, \dots, A_N$  con universos  $X_1, \dots, X_N$  respectivamente, se define el producto cartesiano como un conjunto difuso en  $X_1 \times \dots \times X_N$  con la siguiente función de pertenencia:

$$\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \min \{ \mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n) \}$$

según Mamdani (1974)

$$\mu_{A_1 \times \dots \times A_n}(x_1, \dots, x_n) = \mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \dots \cdot \mu_{A_n}(x_n)$$

según Larsen (1980).

### 3.3. Modelos Difusos

Los modelos difusos se basan en un conjunto de reglas heurísticas donde las variables lingüísticas de las entradas y salidas se representan por conjuntos difusos.

#### 3.3.1. Sistema de Lógica Difusa tipo Mamdani

La figura 3.2 muestra las principales componentes de un modelo difuso lingüístico tipo Mamdani: interfaz de fusificación, base de conocimiento, motor de inferencia e interfaz de defusificación (Lee, 1990).

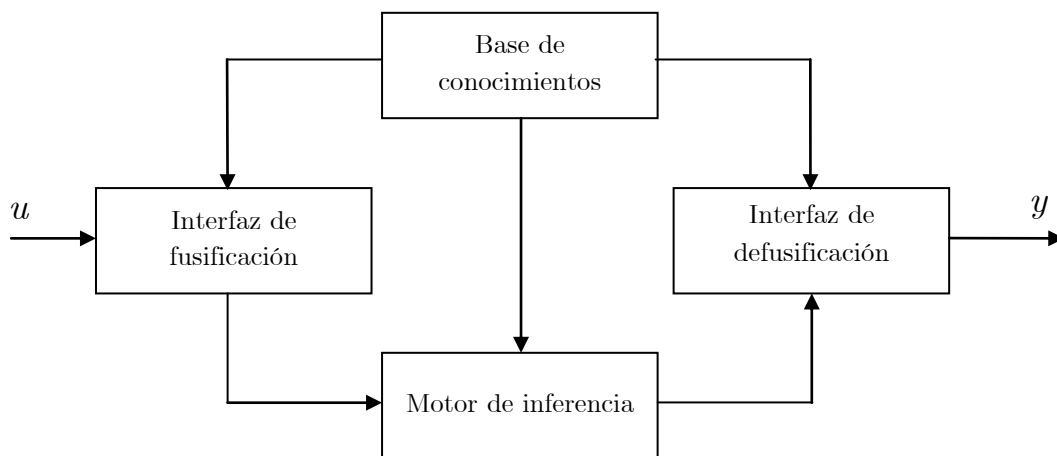


Figura 3.2: Estructura de un Sistema de Lógica Difusa tipo Mamdani.

- **Interfaz de fusificación.** Este elemento transforma las variables de entrada del modelo ( $u$ ) en variables difusas. Para esta interfaz se deben tener definidos los rangos de variación de las variables de entrada y los conjuntos difusos asociados con sus respectivas funciones de pertenencia.

- **Base de conocimientos o reglas.** Contiene las reglas lingüísticas del control y la información referente a las funciones de pertenencia de los conjuntos difusos. Estas reglas lingüísticas, tienen típicamente la siguiente forma: “Si  $u_1$  es  $A$  y  $u_2$  es  $B$ , entonces  $y$  es  $C$ ”, donde  $A$ ,  $B$  y  $C$  son los conjuntos difusos de las variables de entrada  $u_1$  y  $u_2$ , y de la variable de salida y respectivamente.

Existen varias formas de derivar las reglas (Lee, 1990), entre las que destacan las basadas en:

- La experiencia de expertos y el conocimiento de ingeniería de control. La base de reglas se determina a partir de entrevistas con el operador o a través del conocimiento de la dinámica del proceso.

- La modelación del proceso. Los parámetros de la base de conocimiento se obtienen a partir de datos de entrada y salida del proceso.
- **Motor de inferencia.** Realiza la tarea de calcular las variables de salida a partir de las variables de entrada, mediante las reglas del controlador y la inferencia difusa, entregando conjuntos difusos de salida. Por ejemplo, dada una base de conocimiento con  $n$  reglas del tipo: “Si  $u_1$  es  $A_i$  y  $u_2$  es  $B_i$  entonces  $y$  es  $C_i$ ”. La secuencia de cálculos que realiza el motor de inferencia incluye:

- Determinar el grado de cumplimiento  $W_i$  de cada regla a partir de los grados de pertenencia de las variables de entrada obtenidos en la etapa de fusificación, es decir,

$$W_i = \min(\mu_{A_i}, \mu_{B_i})$$

debido a que las premisas de la reglas están unidos por operadores AND, definidos como la intersección de conjuntos difusos.

- Para cada regla se tiene una consecuencia “ $y$  es  $C_i$ ”, que tiene asociado una función de pertenencia  $\mu_{C_i}$ . Por lo tanto, se tiene un conjunto de salida  $C'_i$ , cuya función de pertenencia es:

$$\mu_{C'_i} = \min(W_i, \mu_{C_i})$$

donde  $W_i$  es el grado de cumplimiento para la regla  $i$ .

- Para evaluar el conjunto total de reglas, se unen los conjuntos difusos  $C'_i$  resultantes de cada regla, generándose un conjunto de salida con la siguiente función de pertenencia:

$$\mu_{C'} = \max(\mu_{C'_i}), \quad i = 1, \dots, n$$

De esta forma, se obtiene una salida difusa del controlador, con una función de pertenencia  $\mu_{C'}$ .

- **Interfaz de defusificación.** Este elemento provee salidas discretas y determinísticas a partir de los conjuntos difusos  $C'$  obtenidos como resultado de la inferencia. Existen diferentes métodos de defusificación, algunos de los cuales se describen a continuación:

- **Método del máximo.** La salida corresponde al valor para el cual la función de pertenencia  $\mu_{C'}$  alcanza su máximo. Se ocupa la media del máximo (MOM), mínimo del máximo (LOM) o el mayor del máximo

(SOM), donde la salida es el promedio, el menor y el mayor valor entre los elementos de  $C'$  que tienen un grado de pertenencia máximo.

- **Centro de gravedad (centroide).** Genera como salida el valor correspondiente al centro de gravedad de la función de pertenencia del conjunto de salida  $C'$ .

En forma continua  $x_0 = \frac{\int f(x)xdx}{\int f(x)dx}$

En forma discreto  $x_0 = \frac{\sum_{i=1}^n f(x_i)x_i}{\sum_{i=1}^n f(x_i)}$

- **Bisección.** Genera como salida el valor correspondiente al punto en el cual la función de pertenencia del conjunto de salida  $C'$  se divide en dos áreas iguales.

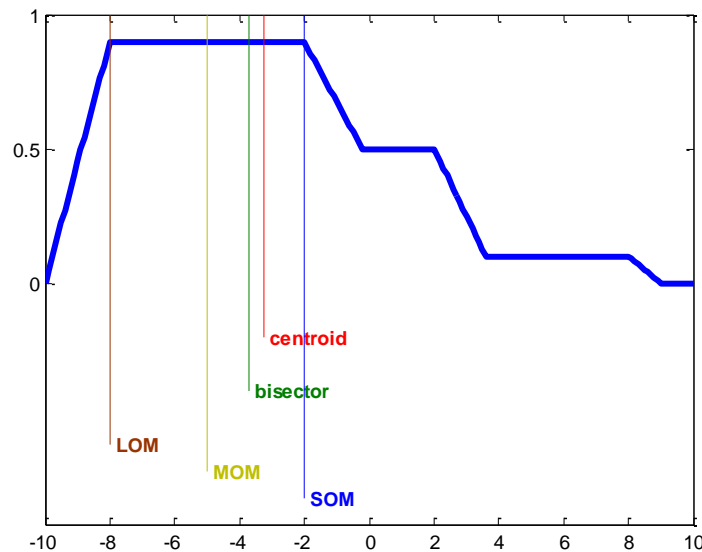


Figura 3.3: Esquema de los métodos de defusificación

En particular, los modelos dinámicos difusos lingüísticos, se pueden expresar como:

$R_i$  : Si  $y(t-1)$  es  $A_{1i}$  & ... &  $y(t-n_y)$  es  $A_{ni}$  &  $u(t-1)$  es  $B_{1i}$  & ... &  $u(t-n_u)$  es  $B_{ni}$   
 Entonces  $y_i(t)$  es  $C_i$

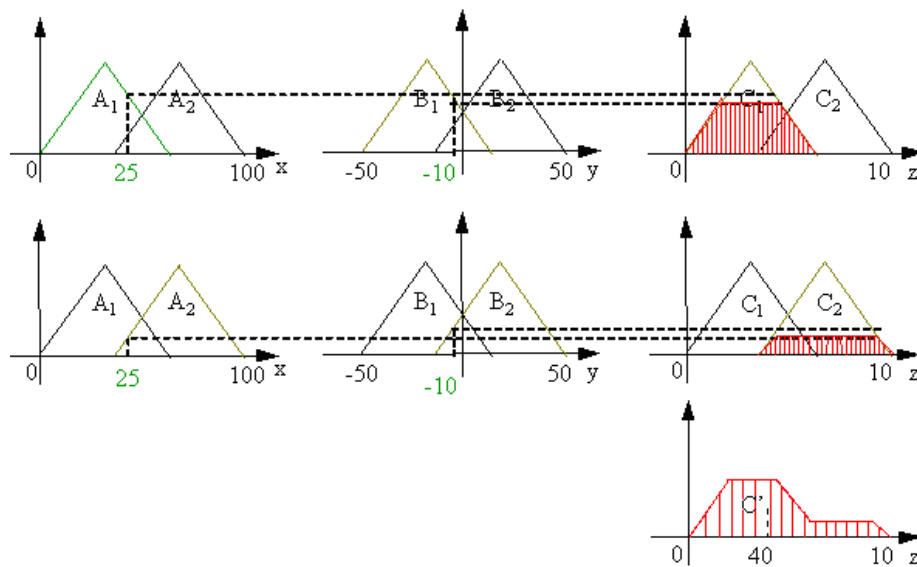
donde  $y_i$  es la salida de la regla  $i$ .

**Ejemplo 1.** Sea el set de reglas dado por

$R_1$ : Si  $x$  es  $A_1$  e  $y$  es  $B_1$  entonces  $z$  es  $C_1$

$R_2$ : Si  $x$  es  $A_2$  e  $y$  es  $B_2$  entonces  $z$  es  $C_2$

Para  $x = 25$  e  $y = -10$ , ¿cuánto vale  $z$ ?



**Ejemplo 2.** Modelación de las acciones de control de un operador de un horno cementero. En la figura 3.4 se presenta un diagrama del proceso, donde el carbón proveniente de un molino, alimenta una tolva y es transportado hacia el horno. El ventilador primario sirve para mantener la llama en la zona de cocción.

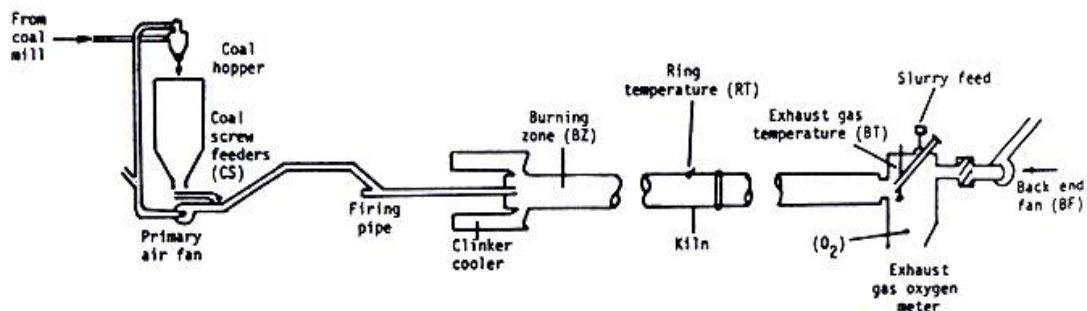


Figura 3.4: Diagrama del proceso de producción de un Horno Cementero

La mezcla, que permite la formación de los compuestos del cemento, se desplaza desde la parte posterior del horno (derecha de la figura 3.5) en contracorriente al flujo de calor, calcinándose y cociéndose, hasta obtenerse el clinker o producto



final del horno que pasa posteriormente al enfriador. El ventilador de inducción sirve para succionar los gases producidos en la combustión.

Es necesario identificar las variables de entrada y salida del proceso, las cuales vienen dadas por:

**Entradas:**

- Flujo de alimentación del carbón (CS),
- Velocidad del ventilador de inducción (BF)
- Velocidad del horno (KS).

**Salidas:**

- Porcentaje de monóxido de carbono en los gases (CO),
- Temperatura de los gases en la zona intermedia (RT),
- Porcentaje de oxígeno en los gases (O<sub>2</sub>),
- Temperatura de los gases en la zona posterior (BT)
- Porcentaje de óxido nitroso en los gases (NO<sub>x</sub>).

Experimentalmente, se ha comprobado que la dinámica de este sistema es no lineal, con retardos, fuertes interacciones y muy dependiente de las condiciones iniciales. A partir de la experiencia de operadores para hornos cementeros, se puede deducir, en términos generales, que el flujo de carbón es la variable manipulada que produce el mayor efecto. Por ejemplo, un aumento de CS genera:

- una disminución de O<sub>2</sub> y CO, y
- un aumento de las temperaturas RT y BT, y del NO<sub>x</sub>.

Por su parte, un aumento en BF

- aumenta el O<sub>2</sub>, CO y BT, y
- disminuye RT y NO<sub>x</sub>.

Al aumentar KS

- aumentan RT y BT.

A partir de estas afirmaciones, una regla que representa las acciones de control del operador puede ser:

- Si CO es alto entonces CS aumenta y BF disminuye

### 3.3.2. Modelos difusos de Takagi y Sugeno

Estos modelos se caracterizan por relaciones basadas en reglas difusas, donde las premisas de cada regla representan subespacios difusos y las consecuencias son una relación lineal de entrada-salida (Takagi y Sugeno, 1995).

Las variables de entrada en las premisas de cada regla son relacionadas por operadores "y, & o AND" y la variable de salida es función de las variables de estado, en general, una función lineal. Por lo tanto, las reglas del modelo tienen la siguiente forma:

$$R_i : \text{Si } X_1 \text{ es } A_{1i} \text{ y } \dots \text{ y } X_k \text{ es } A_{ki} \text{ entonces } Y_i = f_i(X_1, \dots, X_k)$$

donde

- $X_1, \dots, X_k$  son las variables de entrada o premisas de las reglas.
- $A_{1i}, \dots, A_{ki}$  son los conjuntos difusos asociados a las variables de entrada.
- $p_0^i, \dots, p_k^i$  son los parámetros de la regla  $i$ .
- $Y_i$  es la salida de la regla  $i$ .
- $f_i$  es una función que generalmente es lineal, es decir:

$$Y_i = p_0^i + p_1^i X_1 + \dots + p_k^i X_k$$

Por lo tanto, la salida del modelo,  $Y$ , se obtiene ponderando la salida de cada regla por su respectivo grado de cumplimiento  $W_i$ , es decir:

$$Y = \sum_{i=1}^M (W_i Y_i) / \left( \sum_{i=1}^M W_i \right)$$

donde  $M$  es el número de reglas del modelo y  $W_i$  corresponde al grado de activación de la regla  $i$ , que se define como:

$$w_i = \text{oper}(\mu_{A_{1i}}, \dots, \mu_{A_{mi}}, \dots, \mu_{A_{ki}})$$

donde "oper" es el operador mínimo o el producto, y  $\mu_{A_{mi}}$  es el grado de pertenencia de la variable de entrada  $X_m$  al conjunto difuso  $A_{mi}$  para  $m = 1, \dots, k$

En particular, los modelos dinámicos de tiempo discreto de Takagi y Sugeno están dados por:

$$\begin{aligned}
 R_i : & \text{ Si } y(t-1) \text{ es } A_{l_i} \text{ y } \dots \text{ y } y(t-n_y) \text{ es } A_{n_{y_i}} \text{ y} \\
 & u(t-1) \text{ es } B_{l_i} \text{ y } \dots \text{ y } u(t-n_u) \text{ es } B_{n_{u_i}} \\
 & \text{ entonces } y_i(t) = a_1^i y(t-1) + \dots + a_{n_y}^i y(t-n_y) + \\
 & b_1^i u(t-1) + \dots + b_{n_u}^i u(t-n_u) + c^i
 \end{aligned}$$

donde  $a_j^i$ ,  $b_j^i$  y  $c^i$  son parámetros de los modelos de las consecuencias.

También los modelos dinámicos de Takagi y Sugeno se pueden representar en variables de estado, es decir:

$$\begin{aligned}
 R_i : & \text{ Si } x_1(t) \text{ es } A_{l_i} \text{ y } \dots \text{ y } x_n(t) \text{ es } A_{n_i} \\
 & \text{ entonces } x_i(t+1) = A^i x(t) + B^i u(t) + C^i
 \end{aligned}$$

donde  $x = [x_1, \dots, x_n]$  es un vector de variables de estado del proceso, y  $A^i$ ,  $B^i$  y  $C^i$  son las matrices de los modelos lineales en variables de estado de las consecuencias.

**Ejemplo 3:** Modelo difuso de Takagi y Sugeno para un fermentador batch de alimentación. La presión en el estanque de fermentación puede ser controlada a través del cambio de flujo de aire de salida manteniendo constante el flujo de aire de entrada.

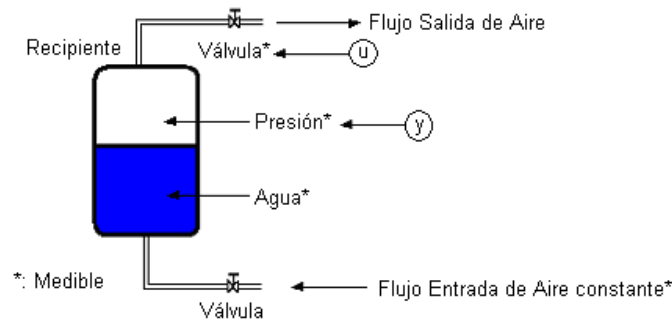


Figura 3.5: Fermentador Batch.

A continuación se presentan la base de reglas del modelo de Takagi y Sugeno que caracterizan al fermentador.

**R1:** Si la presión  $y(k)$  es **BAJA** y la válvula  $u(k)$  está **ABIERTA**

Entonces  $y(k+1) = 0.67y(k) + 0.0007u(k) + 0.35$

**R2:** Si la presión  $y(k)$  es **MEDIA** y la válvula  $u(k)$  está **MEDIA CERRADA**

Entonces  $y(k+1) = 0.80y(k) + 0.0028u(k) + 0.07$

**R3:** Si la presión  $y(k)$  es **ALTA** y la válvula  $u(k)$  está **CERRADA**

Entonces  $y(k+1) = 0.90y(k) + 0.0071u(k) - 0.39$

Funciones de pertenencia del modelo de Takagi y Sugeno que caracterizan al fermentador

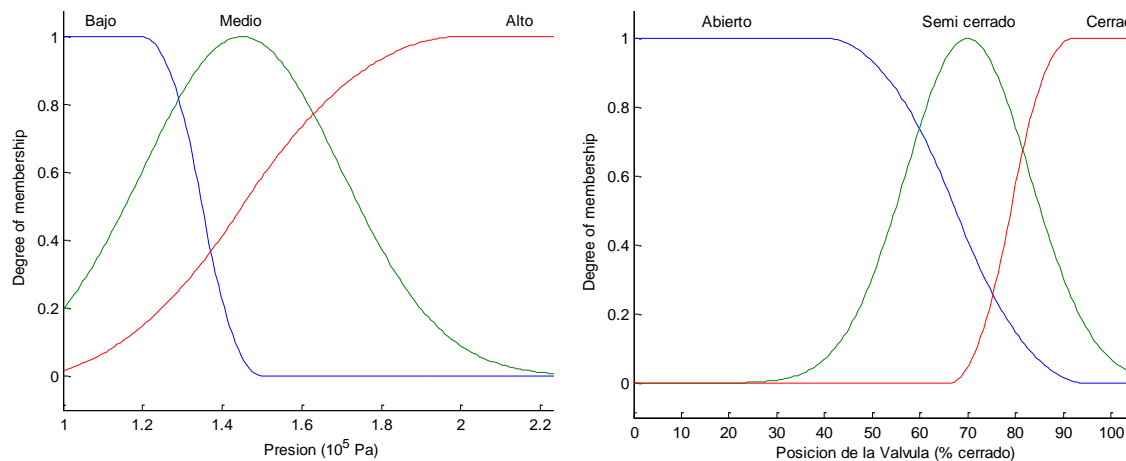


Figura 3.6: Conjuntos difusos para la válvula

### 3.4. Método de Identificación de un Modelo Takagi y Sugeno

El problema de Identificación de Sistemas puede entenderse como el problema de obtener un modelo de un sistema real, a partir de mediciones efectuadas sobre las entradas y salidas de ese sistema. En la siguiente figura se presenta un diagrama del método de identificación:

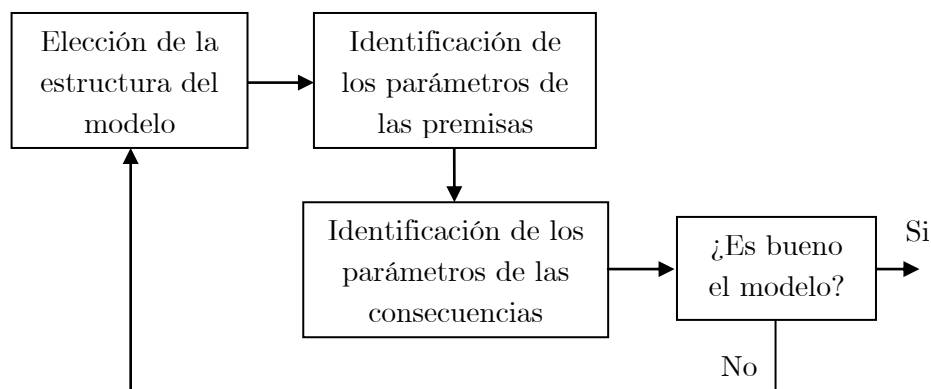


Figura 3.7: Diagrama del método de identificación

## Identificación de parámetros de las premisas

- Conocimiento del proceso
- “Clustering” difuso

## Identificación de parámetros de las consecuencias

- Método de Takagi y Sugeno basado en mínimos cuadrados.

## Elección de la estructura del modelo

- Método heurístico
- Análisis de sensibilidades

### 3.4.1. Identificación de los parámetros de las premisas

En esta estructura los conjuntos difusos  $A_{1i}, \dots, A_{ki}$  y sus respectivas funciones de pertenencia pueden ser determinadas basándose en un conocimiento previo del proceso o por métodos más complejos como “clustering” difuso.

- **“Clustering” difuso:** El número óptimo de reglas y conjuntos difusos del modelo se determina haciendo una partición del universo de la variable de salida y luego proyectándolo al espacio de entrada (Sugeno, 1993). Como se presenta en la figura 3.8, generalmente para la modelación difusa, se pone primero atención en las premisas de las reglas y se encuentra la partición de las variables de entrada. Esta partición se realiza independientemente para cada una de las variables, encontrándose las reglas difusas al realizar las diversas combinaciones posibles entre los conjuntos difusos obtenidos.

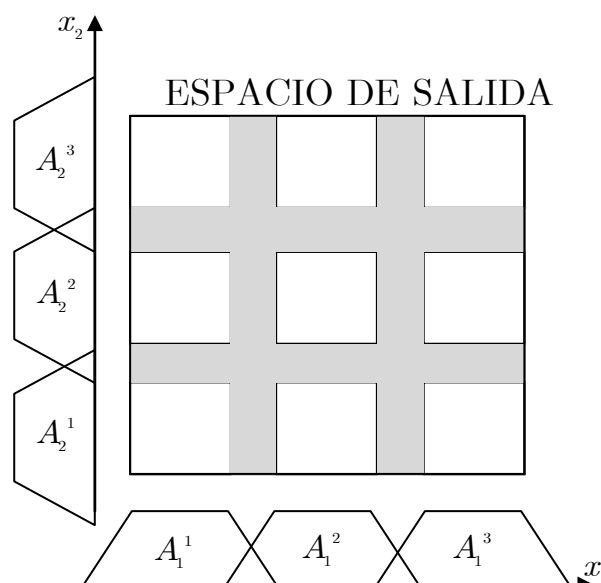


Figura 3.8: Partición de espacios de salida.

En “Clustering” difuso, el número óptimo de reglas y conjuntos difusos del modelo se determina haciendo una partición del universo de la variable de salida y luego proyectándolo al espacio de entrada (Sugeno, 1993).

Para obtener los conjuntos difusos de la salida o los clusters asociados a la salida, el criterio utilizado es considerar una partición del espacio de salida tal que cada conjunto o cluster sea lo más distinto posible a los otros pero que sus elementos sean lo más parecido posible entre sí. Esto es que la varianza entre clusters debe ser máxima pero mínima al interior de éstos:

$$S(c) = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^m \left( \|x_k - v_i\|^2 - \|v_i - \bar{x}\|^2 \right)$$

donde

- $n$ : número de datos para el clustering,
- $c$ : número de clusters,
- $x_k$ :  $k$ -ésimo vector de datos,
- $\bar{x}$ : promedio de los datos,
- $v_i$ : centro del cluster  $i$ -ésimo,
- $\mu_{ik}$ : grado del  $k$ -ésimo dato perteneciente al  $i$ -ésimo cluster,
- $m$ : peso ajustable.

Por lo tanto, se obtiene que todas las salidas tengan asociado un grado de pertenencia al conjunto difuso  $B$ .

A continuación, se consideran los siguientes datos asociados con grado de pertenencia de  $y^i$  en  $B^j$  ( $1 \leq j \leq c$ )

$$(x^i, y^i), B^1(y^i), B^2(y^i), \dots, B^c(y^i)$$

A continuación los conjuntos difusos correspondientes son inducidos al realizar una proyección de los clusters del espacio de salida sobre el espacio de entradas. Esta proyección se realiza de modo tal que se cumpla la siguiente condición:

$$A_1(x_1^i) = A_2(x_2^i) = \dots = A_p(x_p^i) = B(y^i).$$

Como se muestra en la siguiente figura 3.9, el conjunto  $B$  del espacio de salida ha sido proyectado en los conjuntos difusos  $A_j$  del espacio de entrada, posibilitando de esta manera establecer la siguiente regla difusa:

$$R : \text{Si } x_1 \text{ es } A_1 \text{ y } x_2 \text{ es } A_2 \text{ entonces } y \text{ es } B$$

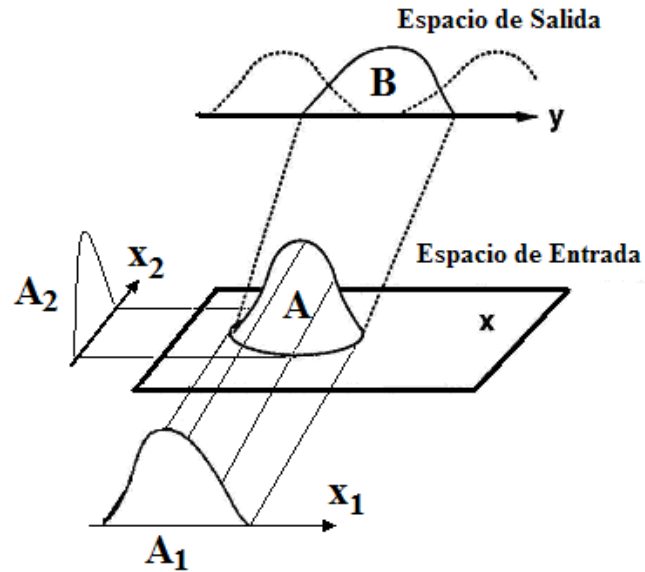


Figura 3.9: Proyección del espacio de salida en el espacio de entrada.

Sin embargo, se debe considerar que el conjunto  $A$  sea no convexo como se muestra en la siguiente figura 3.10. En este caso el conjunto difuso del espacio de salida se proyecta en más de un cluster en el espacio de entradas. Por lo tanto, se obtiene más de una regla para el mismo cluster del espacio de salida, es decir:

$$R^1 : \text{Si } x_1 \text{ es } A_1^1 \text{ y } x_2 \text{ es } A_2^1 \text{ entonces } y \text{ es } B$$

$$R^2 : \text{Si } x_1 \text{ es } A_1^2 \text{ y } x_2 \text{ es } A_2^2 \text{ entonces } y \text{ es } B$$

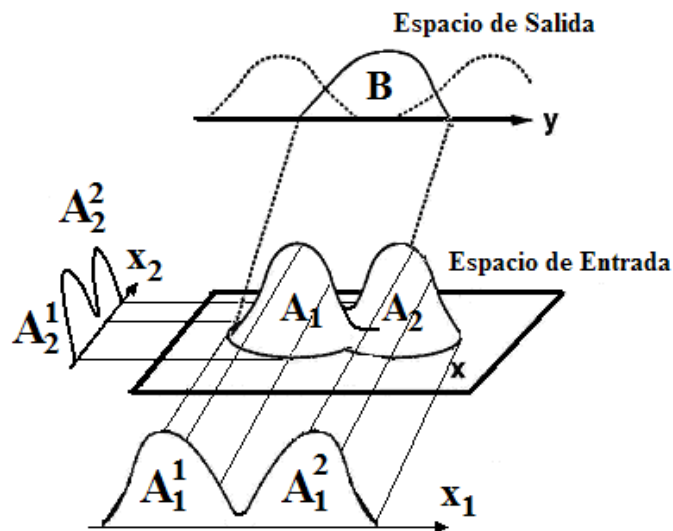


Figura 3.10: Proyección del espacio de salida en más de un cluster.

De esta manera, el clustering difuso entrega el número óptimo de reglas difusas y conjuntos difusos.

### 3.4.2. Identificación de los parámetros de las consecuencias

En general, los parámetros  $p_0^i, \dots, p_k^i$  de las consecuencias se obtienen por el método de mínimos cuadrados, es decir, se minimiza el índice de error dado por:

$$e^2 = \sum_{p=1}^N (y_p - \hat{y}_p)^2$$

donde

- $y_p$  es la salida real del proceso.
- $\hat{y}_p$  es la salida del modelo difuso, considerando las mismas entradas del proceso.
- $N$  es el número de muestras.

Entonces dado un conjunto de  $N$  muestras de entrada/salida, el algoritmo calcula, para la muestra  $j$ , los grados de cumplimiento de cada regla según la definición dada anteriormente, lo que corresponde a:

$$W_{ij} = \min(\mu_{A1i}(X_{1j}); \mu_{A2i}(X_{2j}); \dots; \mu_{Aki}(X_{kj}))$$

con:  $1 \leq i \leq M$ ,  $1 \leq j \leq N$ .  $M$  es el número de reglas,  $N$  es el número de muestras y  $k$  es el número de variables.

La salida del modelo es:

$$y_j = \frac{\sum_{i=1}^M (p_o^i + p_1^i X_{1j} + \dots + p_k^i X_{kj}) W_{ij}}{\sum_{i=1}^M W_{ij}}$$

Se define  $B_{ij} = \frac{W_{ij}}{\sum_{h=1}^M W_{hj}}$  entonces, la predicción de la salida  $j$  está dada por la

siguiente expresión:

$$y_j = \sum_{i=1}^M (p_o^i B_{ij} + p_1^i B_{ij} X_{1j} + \dots + p_k^i B_{ij} X_{kj})$$

Si se construye la matriz  $x$  de dimensiones  $N, M \cdot (k+1)$

$$x = \begin{bmatrix} B_{11} \dots B_{M1} & X_{11} B_{11} \dots X_{11} B_{M1} & \dots & X_{k1} B_{11} \dots X_{k1} B_{M1} \\ \vdots & \vdots & \vdots & \vdots \\ B_{1N} \dots B_{MN} & X_{1N} B_{1N} \dots X_{1N} B_{MN} & \dots & X_{kN} B_{1N} \dots X_{kN} B_{MN} \end{bmatrix}$$



el vector de salida  $y = [y_1, \dots, y_N]^T$  se expresa de la forma:

$$y = xP$$

Donde  $P = [p_0^1, \dots, p_0^M, p_1^1, \dots, p_1^M, \dots, p_k^1, \dots, p_k^M]$  es el vector de parámetros.

La identificación se reduce, entonces, a resolver la ecuación anterior encontrando el vector  $P$  mediante un algoritmo de mínimos cuadrados.

$$\begin{aligned} y &= xP & / & x^T() \\ x^T y &= x^T x P \\ P &= (x^T x)^{-1} x^T y \end{aligned}$$

Finalmente, con los parámetros óptimos de las consecuencias ya determinados se puede alterar la estructura del modelo o las funciones de pertenencia obtenidas, de manera de reducir el índice de error.

### 3.4.3. Elección de la estructura del modelo

En este caso, la estructura del modelo difuso se define como la selección de las variables de entrada significativas.

#### 3.4.3.1. Método heurístico (Sugeno, 1993)

En general, para el modelo difuso definido por:

$$\begin{aligned} \text{Si } y(k-1) \text{ es } A_1^r \ \& \ \dots \ \& \ y(k-n_a) \text{ es } A_{na}^r \ \& \\ u(k-n_k-1) \text{ es } A_{na+1}^r \ \& \ \dots \ \& \ u(k-n_b-n_k) \text{ es } A_{na+nb}^r \\ \text{entonces } y_r(k) &= g_0^r + g_1^r y(k-1) + \dots + g_{na}^r y(k-na) \\ &+ g_{na+1}^r u(k-nk-1) + \dots + g_{na+nb}^r u(k-nb-nk) \end{aligned}$$

Se tienen  $(na + nb)$  variables candidatas de entrada, de este modo el total de posibles modelos considerando todas las entradas es:  $2^{na+nb} - 1$ .

Similarmente al método de regresión por pasos, el método heurístico consiste en seleccionar algunas variables de entrada dentro de todas las variables de entrada candidatas, incrementando el número de entradas de una en una, de acuerdo a un cierto criterio. Por ejemplo, el siguiente criterio puede ser usado:

$$RC = \frac{\left[ \sum_{i=1}^{k_A} (y^A(i) - y^{AB}(i))^2 + \sum_{i=1}^{k_B} (y^B(i) - y^{BA}(i))^2 \right]}{2}$$

donde  $k_A$  y  $k_B$  son el número de datos de dos grupos del conjunto de ajuste,  $y^A(i)$  y  $y^B(i)$  son las salidas de los grupos  $A$  y  $B$ ,  $y^{AB}(i)$  es la salida estimada para el grupo  $A$  con el modelo identificado usando el conjunto de datos  $B$ , e  $y^{BA}(i)$  es la salida estimada para el grupo  $B$  con el modelo identificado usando el conjunto de datos  $A$ .

Pasos del algoritmo:

- Identificar  $(na + nb)$  modelos difusos con una variable de entrada cada uno.
- Calcular el criterio RC para cada modelo y seleccionar el modelo con menor RC.
- Fijar la variable seleccionada en el paso anterior y agregar otra variable de entrada de las restantes candidatas.
- Continuar hasta que RC aumente.

### 3.4.3.2. Análisis de Sensibilidades

Se considera el siguiente modelo no lineal:

$$y(k) = f(x(k))$$

con

$$x(k) = \begin{bmatrix} y(k-1) \\ \vdots \\ y(k-n_a) \\ u(k-n_k-1) \\ \vdots \\ u(k-n_k-n_b) \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{na} \\ x_{na+1} \\ \vdots \\ x_{na+nb} \end{bmatrix}$$

La función  $f$  no lineal con modelos difusos de Takagi-Sugeno está dada por:

$$\begin{aligned} &\text{if } y(k-1) \text{ is } A_1^r \text{ and } \dots \text{ and } y(k-n_a) \text{ is } A_{na}^r \text{ and} \\ &u(k-n_k-1) \text{ is } A_{na+1}^r \text{ and } \dots \text{ and } u(k-n_k-n_b) \text{ is } A_{na+nb}^r \\ &\text{then } y_r(k) = g_0^r + g_1^r y(k-1) + \dots + g_{na}^r y(k-n_a) \\ &\quad + g_{na+1}^r u(k-n_k-1) + \dots + g_{na+nb}^r u(k-n_k-n_b) \end{aligned}$$

En general, la sensibilidad de las variables de entrada de un modelo no lineal está dada por:

$$\xi_i(x) = \frac{\partial f(x)}{\partial x_i}$$

La sensibilidad de cada variable de entrada representa la relevancia de esta entrada en la salida del modelo.

Por ejemplo, en la figura se muestra que la variable de entrada  $x_2$  es irrelevante para la salida del modelo, es decir,  $\frac{\partial f(x_1, x_2)}{\partial x_2} = 0$

Función no lineal  $y = f(x_1, x_2)$

En este caso, la función no lineal  $f$  para los modelos difusos de Takagi (1995):

$$\begin{aligned} \text{Si } y(k-1) \text{ es } A_1^r \text{ y } \dots \text{ y } y(k-n_a) \text{ es } A_{na}^r \text{ y} \\ u(k-n_k-1) \text{ es } A_{na+1}^r \text{ y } \dots \text{ y } u(k-n_b-n_k) \text{ es } A_{na+nb}^r \text{ entonces} \\ y_r(k) = g_0^r + g_1^r y(k-1) + \dots + g_{na}^r y(k-n_a) \\ + g_{na+1}^r u(k-n_k-1) + \dots + g_{na+nb}^r u(k-n_b-n_k) \end{aligned}$$

La salida del modelo difuso es:

$$y(k) = \frac{\sum_{r=1}^{N_r} w_r y_r(k)}{\sum_{r=1}^{N_r} w_r}$$

con el grado de activación de las reglas dado por:

$$w_r = \mu_1^r \cdots \mu_i^r \cdots \mu_{na+nb}^r$$

Se considera la siguiente función de pertenencia para el conjunto difuso  $A_i^r$ , pues es diferenciable:

$$\mu_i^r = e^{\frac{-(a_i^r(x_i-b_i^r))^2}{2}}$$

Para estos modelos difusos definidos, la sensibilidad de una variable de entrada está dada por:

$$\xi_i(x) = \frac{\sum_{r=1}^{N_r} \left( \frac{\partial w_r}{\partial x_i} y_r + \frac{\partial y_r}{\partial x_i} w_r \right) \sum_{r=1}^{N_r} w_r - \sum_{r=1}^{N_r} \left( \frac{\partial w_r}{\partial x_i} \right) \sum_{r=1}^{N_r} (w_r y_r)}{\left( \sum_{r=1}^{N_r} w_r \right)^2}$$

donde

$$\begin{aligned}
\frac{\partial w_r}{\partial x_i} &= \frac{\partial \mu_i^r}{\partial x_i} \times \mu_1^r \times \cdots \times \mu_{i-1}^r \times \mu_{i+1}^r \times \cdots \times \mu_{na+nb}^r \\
\frac{\partial \mu_i^r}{\partial x_i} &= \mu_i^r \times c_i^r \\
c_i^r &= -\left(a_i^r \times (x_i - b_i^r)\right) \times a_i^r \\
\frac{\partial y_r}{\partial x_i} &= g_i^r
\end{aligned}$$

Entonces, la sensibilidad  $\xi_i$  con respecto a la variable de entrada  $x_i$  de un modelo difuso es:

$$\xi_i(x) = \frac{\sum_{r=1}^{Nr} (w_r c_i^r y_r + g_i^r w_r) \sum_{r=1}^{Nr} w_r - \sum_{r=1}^{Nr} (w_r c_i^r) \sum_{r=1}^{Nr} (w_r y_r)}{\left(\sum_{r=1}^{Nr} w_r\right)^2}$$

Las sensibilidades  $\xi_i$  dependen de las variables de entrada  $x$ , siendo evaluadas usando el conjunto de entrenamiento. Para comparar las sensibilidades de las variables de entrada, el siguiente índice de sensibilidades es propuesto  $I_i = \mu_i^2 + \sigma_i^2$ , donde  $\mu_i$  y  $\sigma_i$  es la media y la desviación estándar de las sensibilidades, respectivamente. Entonces, las variables de entrada con menores índices  $I_i$  serán eliminadas. Por último, el modelo difuso es obtenido usando sólo las variables de entrada con mayores sensibilidades.

### **Ejemplo:** Sistema dinámico no lineal Chen (1989)

$$\begin{aligned}
y(k) &= \left(0.8 - 0.5 \exp(-y^2(k-1))\right) y(k-1) - \left(0.3 + 0.9 \exp(-y^2(k-1))\right) y(k-2) \\
&\quad + u(k-1) + 0.2u(k-2) + 0.1u(k-1)u(k-2) + \varepsilon(k)
\end{aligned}$$

con  $u(k)$  una serie independiente e idénticamente distribuida ( $\mu = 0$ ,  $\sigma=1$ ), y  $\varepsilon(k)$  es un ruido blanco ( $\mu = 0$ ,  $\sigma=0.2$ ). Además se tienen 250 datos para el conjunto de entrenamiento 250 datos para el conjunto de validación.

Realizando un análisis de sensibilidades para el sistema, se considera la estructura del modelo inicial difuso considera las 8 variables de entrada  $y(k-1), \dots, y(k-4), u(k-1), \dots, u(k-4)$ .

En el gráfico 3.1, se observa que las variables  $y(k-3), y(k-4), u(k-3), u(k-4)$  presentan menores índices de las sensibilidades por lo cual no se serán incluidos en el modelo difuso. De esta manera, la estructura obtenida considera las variables de entrada  $y(k-1), y(k-2), u(k-1), u(k-2)$ .

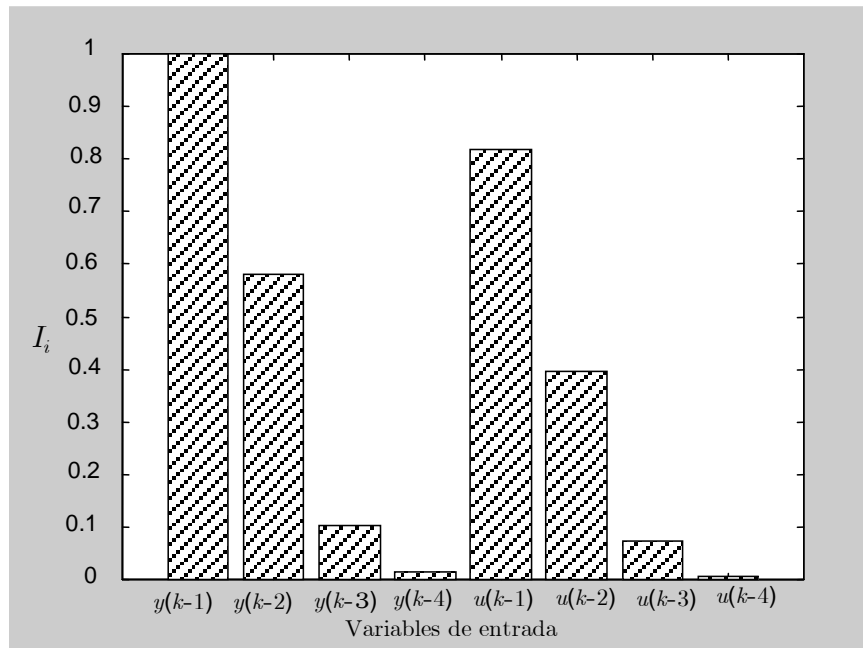


Gráfico 3.1: Índices de sensibilidades para el modelo inicial y las 8 variables de entrada

Modelo	Variables de entrada	$e^2$
1	$y(k-1) - y(k-2) - y(k-3) - y(k-4)$	0.5802
	$u(k-1) - u(k-2) - u(k-3) - u(k-4)$	
2	$y(k-1) - y(k-2) - u(k-1) - u(k-2)$	0.4696

Tabla 3.1: Cálculo de índices de sensibilidad

Realizando un análisis comparativo, el método de sensibilidades para identificación de estructura de modelos difusos permite estudiar el universo completo de modelos posibles, dentro de una complejidad de estructura definida por el orden del modelo inicial.

Además, el método propuesto permite obtener la mejor estructura ajustando menos modelos (dos) que con el método heurístico, en el cual es necesario construir varios modelos.

### 3.5. Control basado en modelos de Takagi y Sugeno

Los modelos difusos propuestos por Takagi y Sugeno (1985), tienen la particularidad de que las consecuencias de cada regla son ecuaciones lineales. Esta característica permite que para cada subespacio lineal o regla del modelo, se pueda derivar un controlador basado en la teoría de control lineal. Por lo tanto, se puede obtener un controlador difuso, que combine las acciones de los controladores lineales de cada regla.

La secuencia de pasos, propuesta por Sugeno y Kang (1986), para diseñar un controlador basado en modelo difuso consiste en:

- a) Identificar un modelo basado en reglas como el siguiente:

$$R_i : \text{Si } Z_1 \text{ es } F_{1i} \text{ y } \dots \text{ y } Z_m \text{ es } F_{mi} \text{ entonces } Y_i = p_0^i + p_1^i X_1 + \dots + p_k^i X_k \quad (*)$$

donde  $Z_1, \dots, Z_m$  son las variables de entrada del modelo y  $X_1, \dots, X_k$  son las variables de estado.

Los parámetros de las premisas y de las consecuencias se obtienen utilizando los métodos descritos anteriormente.

- b) Para cada regla de modelo, se tiene una consecuencia que representa un submodelo lineal del proceso. Esta relación está dada por:

$$Y_i = p_0^i + p_1^i X_1 + \dots + p_k^i X_k$$

- c) A partir de este submodelo, se puede derivar un controlador lineal convencional, como por ejemplo: un controlador de varianza mínima, un controlador lineal cuadrático, etc. De esta manera, se obtiene una acción de control para la regla  $i$  dada por:

$$u_i = f_i(Y_i, X_1, \dots, X_k) \quad (**)$$

donde  $f_i$  es una función lineal descrita por el controlador lineal empleado.

- d) Para construir la base de reglas del controlador difuso, se utilizan las mismas premisas del modelo difuso descritas por la ecuación (\*), mientras las consecuencias se cambian por las acciones de control dadas por la ecuación (\*\*). Por lo tanto, las reglas del controlador son:

$$R_i : \text{Si } Z_1 \text{ es } F_{1i} \text{ y } \dots \text{ y } Z_m \text{ es } F_{mi} \text{ entonces } u_i = f_i(Y_i, X_1, \dots, X_k)$$

- e) Por último, cabe mencionar que para calcular la acción de control final se utiliza la ponderación de la salida de cada regla por su respectivo grado de cumplimiento  $W_i$ . De esta manera, la acción de control final  $u$  es:

$$u = \frac{\sum_{i=1}^M W_i u_i}{\sum_{i=1}^M W_i}$$

donde  $W_i$  es el grado de activación de la regla  $i$  y  $M$  es el número de reglas del modelo difuso.

### 3.5.1. Controladores PD difusos

La Figura 3.9 presenta un diagrama de un controlador PD difuso incremental, donde las entradas son el error  $e(k) = ref - y(k)$  y su tasa de cambio  $de(k) = e(k) - e(k-1)$ , y la salida es el cambio incremental en la variable manipulada  $du(k)$ .

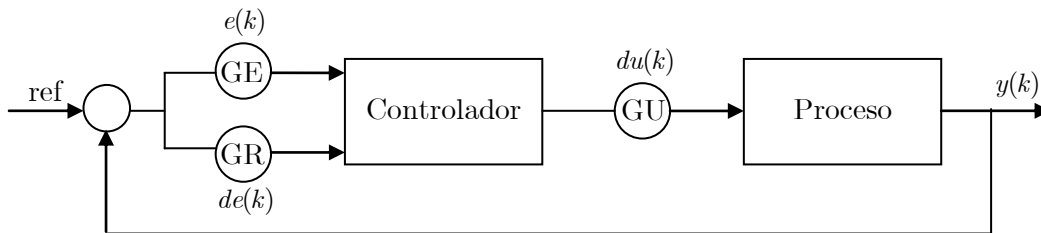


Figura 3.11: Diagrama de un controlador PI difuso.

Los parámetros del controlador son las ganancias GE, GR y GU, que multiplican a  $e(k)$ ,  $de(k)$  y  $du(k)$  respectivamente.

En general, estos controladores presentan las siguientes características: dos o siete conjuntos difusos para las variables de entrada, tres o siete conjuntos difusos para la variable de salida, funciones de pertenencia triangulares, fusificación con universos continuos, implicación utilizando operador min, inferencia basada en implicancia difusa y defusificación por el método de la media de los máximos modificada.

Este tipo de controlador difuso se deriva a partir del comportamiento deseado del sistema en lazo cerrado. En la figura 3.10 se aprecia la respuesta típica de un sistema controlado, donde las entradas al controlador son  $e(k)$  y  $de(k)$ , y la salida es  $du(k)$ .

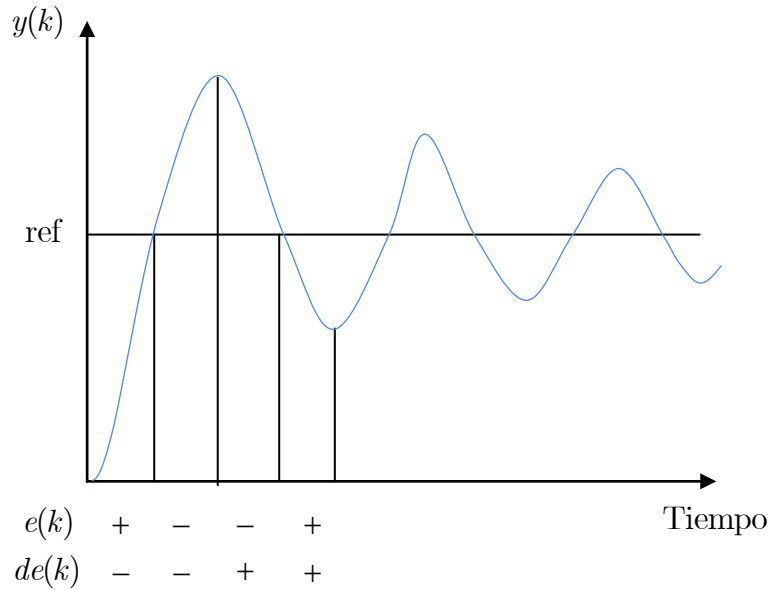


Figura 3.12: Respuesta típica de un sistema controlado con un PI difuso

Analizando en detalle esta respuesta se pueden observar diferentes situaciones. Considerando el valor de  $e(k)$  y el signo de  $de(k)$ , se tienen los casos presentados en la tabla 3.2(a). Además, existen dos tipos de situaciones especiales, cuando el error  $e(k)$  es cero y cuando su tasa de cambio  $de(k)$  es cero. Estas situaciones se muestran en las siguientes tablas 3.2(b) y 3.2(c).

	$e(k)$	$de(k)$
a1	$>0$	$<0$
a2	$<0$	$<0$
a3	$<0$	$>0$
a4	$>0$	$>0$

(a)

	$e(k)$	$de(k)$
b1	$=0$	$<<<0$
b2	$=0$	$<<0$
b3	$=0$	$<0$
b4	$=0$	$>0$
b5	$=0$	$>>0$
b6	$=0$	$>>>0$

(b)

	$de(k)$	$e(k)$
c1	$=0$	$<<<0$
c2	$=0$	$<<0$
c3	$=0$	$<0$
c4	$=0$	$>0$
c5	$=0$	$>>0$
c6	$=0$	$>>>0$

(c)

Tabla 3.2: Situaciones de las variables  $e(k)$  y  $de(k)$

A partir de estas condiciones, se puede configurar una tabla en función de las variables de entrada del controlador (ver Tabla 3.3). En ella se consideran siete conjuntos difusos NB ("Negative Big"), NM ("Negative Medium"), NS ("Negative Small"), ZE ("Zero"), PS ("Positive Small"), PM ("Positive Medium") y PB ("Positive Big") para las variables de entrada que describen los estados  $<<<$ ,  $<<$ ,  $<$ ,  $=$ ,  $>$ ,  $>>$  y  $>>>$  0, respectivamente.



		$de(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	$a2$	$a2$	$a2$	$c1$	$a3$	$a3$	$a3$
	NM	$a2$	$a2$	$a2$	$c2$	$a3$	$a3$	$a3$
	NS	$a2$	$a2$	$a2$	$c3$	$a3$	$a3$	$a3$
	ZE	$b1$	$b2$	$b3$	ZE	$b4$	$b5$	$b6$
	PS	$a1$	$a1$	$a1$	$c4$	$a4$	$a4$	$a4$
	PM	$a1$	$a1$	$a1$	$c5$	$a4$	$a4$	$a4$
	PB	$a1$	$a1$	$a1$	$c6$	$a4$	$a4$	$a4$

Tabla 3.3: Diagrama de estados  $e(k)$  y  $de(k)$ .

Las acciones de control, es decir, los incrementos en la variable manipulada, se definen a partir de la proposición de MacVicar-Whelan (1976), como lo muestra la Tabla 3.4. Por ejemplo, para el elemento de la tercera fila y sexta columna, la regla de control se interpreta como: "Si el error es negativo pequeño y la variación incremental del error es positiva mediana, entonces hacer positiva pequeña la variación incremental en el control".

		$de(k)$						
		NB	NM	NS	ZE	PS	PM	PB
$e(k)$	NB	NB	NB	NB	NB	NM	NS	ZE
	NM	NB	NB	NM	NM	NS	ZE	PS
	NS	NB	NM	NS	NS	ZE	PS	PM
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NM	NS	ZE	PS	PS	PM	PB
	PM	NS	ZE	PS	PM	PM	PB	PB
	PB	ZE	PS	PM	PB	PB	PB	PB

Tabla 3.4: Reglas de control PI difuso.

El diseño de un controlador difuso PI incluye, además de definir las reglas de control, determinar las funciones de pertenencia de cada conjunto difuso. En general, se utilizan funciones triangulares como se muestran en la figura 3.13, donde el universo varía entre  $-L$  y  $L$ , siendo  $L$  un factor de escalamiento de las variables.

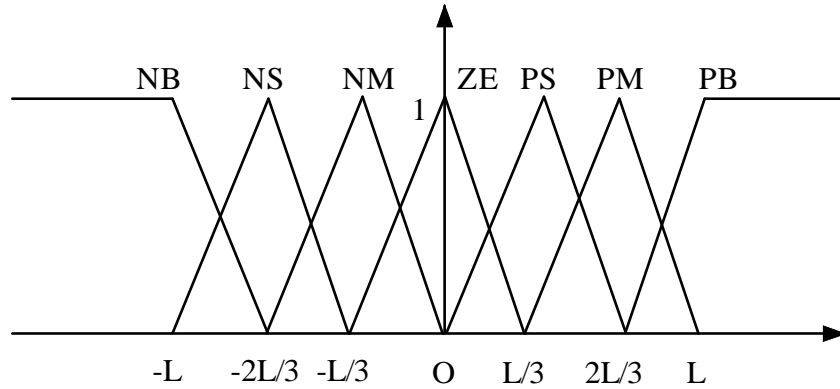


Figura 3.13: Funciones de pertenencia triangulares.

Los principales parámetros de sintonía de estos controladores son las ganancias GE, GR y GU. Existen diversos trabajos que describen la equivalencia entre un controlador PI convencional y un controlador PI difuso (Tang, 1987).

### 3.5.2. Controlador LQR Difuso

El diseño del controlador lineal cuadrático (LQR: "Linear Quadratic Regulator") se realiza considerando un modelo lineal en variables de estado del sistema y minimizando la siguiente función de costos (Astrom, 1984):

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt$$

donde  $x$  es el vector de variables de estado,  $u$  es la variable manipulada.

La solución de este problema se obtiene resolviendo la siguiente ecuación de Ricatti para la matriz  $P$ :

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

donde  $A$  y  $B$  son las matrices del modelo en variables de estado linealizado.

De esta manera, el control obtenido está dado por  $u = -Kx$  con  $K = R^{-1}B^T P$

Para el diseño del controlador LQR difuso, se considera el siguiente modelo difuso en variables de estado:

$$R_i : \text{Si } z_1 \text{ es } F_{l_i} \text{ y } \dots \text{ y } z_m \text{ es } F_{m_i} \text{ entonces } \dot{x} = A^i x + B^i u$$

donde  $z_j$  son las variables de entrada del modelo difuso y  $x$  es el vector de variables de estado

Por lo tanto, el controlador basado en un modelo difuso consiste en una base de reglas con la siguiente estructura:

$$R_i : \text{ Si } z_1 \text{ es } F_{l_i} \text{ y } \dots \text{ y } z_m \text{ es } F_{m_i} \text{ entonces } u_i = f_i(x_1, \dots, x_n)$$

donde la ley de control  $f_i$  corresponde a un controlador lineal cuadrático obtenido a partir del modelo lineal en variables de estado para cada regla  $i$  del modelo difuso

Entonces,

$$R_i : \text{ Si } z_1 \text{ es } F_{l_i} \text{ y } \dots \text{ y } z_m \text{ es } F_{m_i} \text{ entonces } u_i = -k_1^i x_1 - \dots - k_n^i x_n$$

### 3.5.2.1. Controlador LQR Difuso para un péndulo invertido

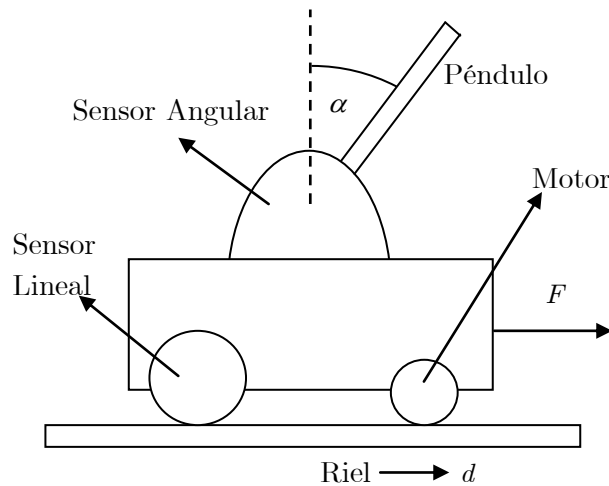


Figura 3.14: Péndulo invertido

El péndulo invertido está compuesto por una viga montada sobre un carro que desliza sobre un riel. El motor de corriente continua permite ejercer la fuerza sobre el sistema y un potenciómetro acoplado a él por un mecanismo de engranaje mide la posición del carro. Además, un potenciómetro montado en el eje de rotación permite medir el ángulo del péndulo con respecto a la vertical.

Se considera como variables de entrada al controlador o variables controladas el ángulo y la velocidad angular del péndulo, y la variable de salida o manipulada es la fuerza aplicada al motor. No se considera el control de la posición del carro.

- **Diseño del controlador LQR difuso**

En este caso, para diseñar el controlador difuso para la estabilización angular de un péndulo invertido se deriva el siguiente modelo difuso del proceso:

$$R_i : \text{Si } x_1 \text{ es } A_i \text{ y } x_2 \text{ es } B_i \text{ entonces } \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

donde  $x_1$  y  $x_2$  son las variables de estado del proceso  $\alpha$  y  $\dot{\alpha}$ ,  $u$  es la variable manipulada  $F$ . Además,  $A_i$  y  $B_i$  son los conjuntos difusos para  $x_1$  y  $x_2$  respectivamente.

En este caso, debido a que el proceso es inestable y no es posible la obtención de datos en lazo abierto, en vez de realizar una identificación del modelo por alguno de los métodos descritos anteriormente, los coeficientes  $a_{ij}$ ,  $b_j$  y  $c_j$  son determinados a partir de la linealización de la siguiente ecuación fenomenológica del proceso en torno a diferentes puntos de operación.

$$\ddot{\alpha} = \frac{(M+m)g \cdot \sin \alpha - F \cdot \cos \alpha - \frac{m\ell}{2} \dot{\alpha}^2 \cdot \sin \alpha \cdot \cos \alpha}{\frac{2(M+m)\ell}{3} - \frac{m\ell \cdot \cos^2 \alpha}{2}}$$

Por lo tanto, el controlador basado en un modelo difuso consiste en una base de reglas con la siguiente estructura:

$$R_i : \text{Si } \alpha \text{ es } A_i \text{ y } \dot{\alpha} \text{ es } B_i \text{ entonces } F_i = -k_1^i \alpha - k_2^i \dot{\alpha} + k_0^i$$

donde  $k_j^i$  son los coeficientes de la regla  $i$  para la variable de estado  $j$ , ( $j=1,2$ ) y  $k_0^i$  es un sesgo dado por la diferencia entre el punto de operación utilizado para la linealización correspondiente a la regla  $i$  y el estado de equilibrio ( $\alpha = 0$  y  $\dot{\alpha} = 0$ )

Los puntos de operación para la linealización se presentan en la tabla 3.5.

Regla	$\alpha$ (rad)	$\dot{\alpha}$ (rad/seg)
1	-0.6981 (-40°)	-1.7453 (-100°/s)
2	-0.6981 (-40°)	1.7453 (100°/s)
3	0.6981 (40°)	-1.7453 (-100°/s)
4	0.6981 (40°)	1.7453 (100°/s)

**Tabla 3.5: Puntos de operación para el problema.**

Los parámetros de las funciones de pertenencia han sido determinados mediante un procedimiento de prueba y error, y sus valores se presentan en la tabla 3.6.

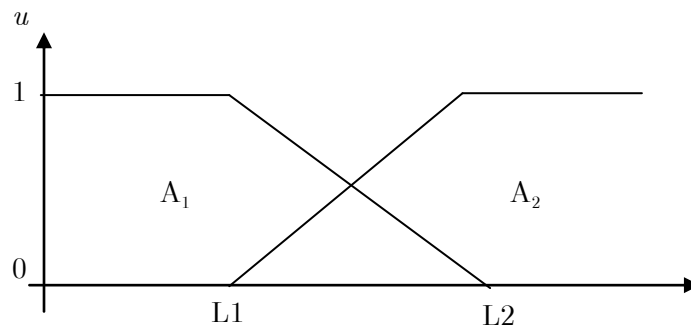


Figura 3.15: Conjuntos difusos para el problema.

	L1	L2
$\alpha$ (rad)	-0.3491 ( $-20^\circ$ )	0.3491 ( $20^\circ$ )
$\dot{\alpha}$ (rad/s.)	-0.8727 ( $-50^\circ/\text{s.}$ )	0.8727 ( $50^\circ/\text{s.}$ )

Tabla 3.6: Parámetros de las premisas.

Los coeficientes de las consecuencias, correspondientes a los de un controlador LQR para cada regla, se presentan a continuación.

Regla	$k_1$	$k_2$	$k_0$
1	-20.2443	-5.9888	-2.0787
2	-20.2443	-6.2131	-2.0787
3	-20.2441	-6.2131	2.0787
4	-20.2441	-5.9887	2.0787

Tabla 3.7: Coeficientes de las consecuencias.

En los siguientes gráficos 3.2 y 3.3, se presentan los resultados del LQR difuso para el péndulo invertido.

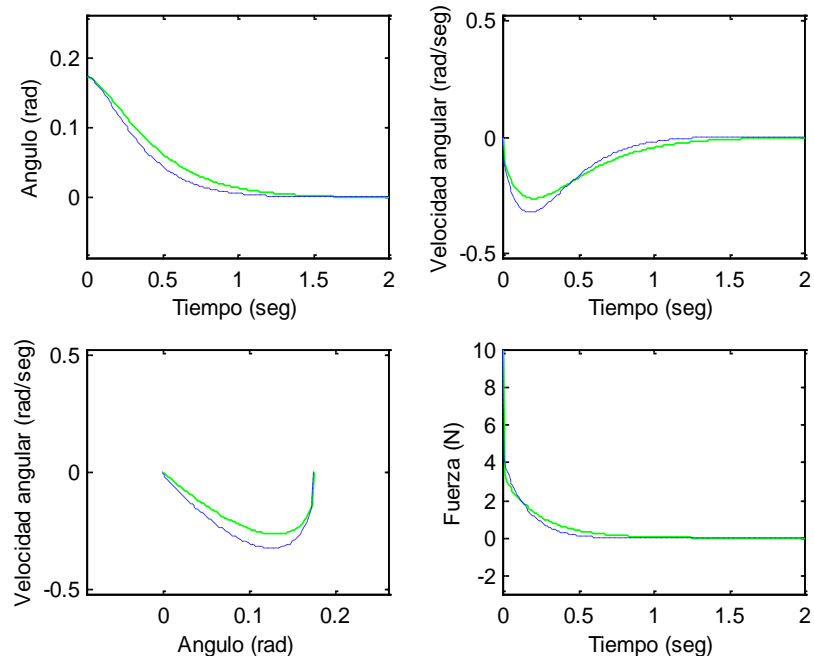


Gráfico 3.2: Condiciones iniciales:  $\alpha_0 = (10^\circ) 0.1745 \text{ rad}$ ,  $\dot{\alpha}_0 = 0$ . Verde: LQR, Azul: LQR difuso.

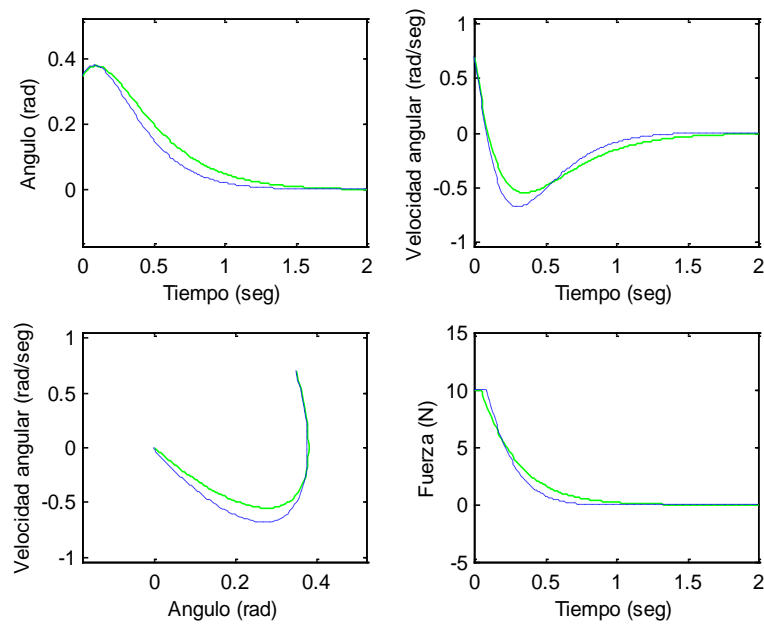


Gráfico 3.3: Condiciones iniciales:  $\alpha_0 = (20^\circ) 0.349 \text{ rad}$ ,  $\dot{\alpha}_0 = (40^\circ/\text{seg}) 0.6981$ .

Verde: LQR, Azul: LQR difuso.

En la tabla 3.8 se presentan los valores de la función de costos del regulador LQR, para todas las pruebas realizadas con diferentes condiciones iniciales.

$\alpha_0$ (rad)	0.1745 ( $10^\circ$ )	0.1745 ( $10^\circ$ )	0.2618 ( $15^\circ$ )	0.3490 ( $20^\circ$ )
$\dot{\alpha}_0$ (rad/s)	0	0.3490 ( $20^\circ/\text{s}$ )	0.5216 ( $30^\circ/\text{s}$ )	0.6981 ( $40^\circ/\text{s}$ )
LQR	4956	8437	19574	37240
LQR difuso	4578	7747	17805	34635

**Tabla 3.8: Valores de la función de costos.**

Finalmente, tomando en cuenta las figuras y los valores de la función de costos podemos decir que el controlador lineal cuadrático difuso (LQR difuso) presenta el mejor desempeño en las pruebas por simulación.

### 3.5.3. Controlador Difuso por ubicación de Polos

Debido a que las consecuencias de los modelos difusos de Takagi y Sugeno son sistemas lineales, se puede diseñar controladores en variables de estado difusos por asignación de polos (Kang y Lee, 1995).

En este caso, se considera el siguiente modelo difuso de Takagi y Sugeno en variables de estado:

$$R_i : \text{Si } z_1 \text{ es } F_{l_i} \text{ y } \dots \text{ y } z_m \text{ es } F_{m_i} \\ \text{entonces } x^i(k+1) = A^i x(k) + B^i u(k) \text{ e } y(k) = Cx(k)$$

Entonces, el vector en variables de estado a partir del modelo difuso es:

$$x(k+1) = \frac{\sum_{i=1}^M w^i(z) x^i(k+1)}{\sum_{i=1}^M w^i(z)}$$

El controlador difuso está compuesto por las premisas del modelo difuso. Entonces, la base de reglas del controlador está dada por:

$$R_i : \text{Si } z_1 \text{ es } F_{l_i} \text{ y } \dots \text{ y } z_m \text{ es } F_{m_i} \text{ entonces } u^i(k) = -G^i x(k)$$

donde  $G^i$  es una matriz de ganancia realimentada en el estado.

En este método, la ganancia  $G^i$  debe ser obtenida tal que el sistema difuso con el controlador difuso tenga los polos deseados. Para esto, se formula el siguiente teorema.

### Teorema 3.1

Se considera que el vector de ganancia  $G^i$  satisface  $\phi = A_i - B_i G^i$ , donde  $\phi$  es la matriz de transición de estados que tiene los polos deseados. Además, se considera que la acción de control debe ser obtenida a partir de la relación:

$$\sum_{i=1}^M w^i(z) B^i u(k) = \sum_{i=1}^M w^i(z) B^i u^i(k) \quad (*)$$

Con el valor de  $u(k) = \frac{\sum_{i=1}^M w^i(z) B^i u^i(k)}{\sum_{i=1}^M w^i(z) B^i}$

Entonces, la matriz de transición del sistema difuso con controlador difuso es  $\phi$  y tiene los polos deseados.

### Demostración

Se considera que  $u(k)$  obtenido de la ecuación (\*) es usado como entrada al sistema, entonces las variables de estado del modelo difuso son:

$$\begin{aligned} x(k+1) &= \frac{\sum_{i=1}^M w^i(z) x^i(k+1)}{\sum_{i=1}^M w^i(z)} \\ &= \frac{\sum_{i=1}^M w^i(z) (A^i x(k) + B^i u(k))}{\sum_{i=1}^M w^i(z)} \\ &= \frac{\sum_{i=1}^M w^i(z) A^i x(k) + \sum_{i=1}^M w^i(z) B^i u(k)}{\sum_{i=1}^M w^i(z)} \end{aligned}$$

Se reemplaza por el controlador difuso  $u^i(k) = -G^i(k)$



$$\begin{aligned}
x(k+1) &= \frac{\sum_{i=1}^M w^i(z) A^i x(k) - w^i(z) B^i G^i x(k)}{\sum_{i=1}^M w^i(z)} \\
&= \frac{\sum_{i=1}^M w^i(z) (A^i - B^i G^i) \cdot x(k)}{\sum_{i=1}^M w^i(z)}
\end{aligned}$$

Reemplazando por la matriz de transición de estados  $\phi = A_i - B_i G^i$

$$x(k+1) = \frac{\sum_{i=1}^M w^i(z) \cdot \phi \cdot x(k)}{\sum_{i=1}^M w^i(z)} = \phi \cdot x(k)$$

Por lo tanto, el sistema en lazo cerrado tiene los polos deseados.

### 3.5.4. Control Predictivo basado en modelos difusos

Los modelos difusos sirven para representar las no linealidades del proceso. A continuación, se presentan las diversas estrategias de control predictivo difuso.

- **Control predictivo difuso, según Cipriano & Ramos (1995)**

GPC basado en modelos difusos de Takagi-Sugeno para un modelo lineal de entrada y salida.

$$\begin{aligned}
R_i : \text{ Si } y(t-1) \text{ es } A_{1i} \text{ y } \dots \text{ y } y(t-n_y) \text{ es } A_{n_{yi}} \text{ y} \\
u(t-1) \text{ es } B_{1i} \text{ y } \dots \text{ y } u(t-n_u) \text{ es } B_{n_{ui}} \\
\text{ entonces } y_i(t) = a_1^i y(t-1) + \dots + a_{n_y}^i y(t-n_y) \\
+ b_1^i u(t-1) + \dots + b_{n_u}^i u(t-n_u) + c^i
\end{aligned}$$

Donde se tiene que la salida viene dada por  $y(t) = \frac{\sum_{i=1}^M w_i y_i}{\sum_{i=1}^M w_i}$

Para cada regla se deriva un controlador GPC lineal. De esta manera, el controlador difuso incluye las mismas premisas que el modelo difuso y las consecuencias están dadas por las acciones de control resultantes (GPC lineales)

$$\begin{aligned}
R_i : & \text{ Si } y(t-1) \text{ es } A_{1i} \text{ y } \dots \text{ y } y(t-n_y) \text{ es } A_{n_{yi}} \text{ y} \\
& u(t-1) \text{ es } B_{1i} \text{ y } \dots \text{ y } u(t-n_u) \text{ es } B_{n_{ui}} \\
& \text{ entonces } \Delta u_i(t) = f_i(\Delta u(t-1), \dots, y(t), y(t-1), \dots)
\end{aligned}$$

con  $f_i$  es un controlador GPC lineal para la regla  $i$ .

Entonces, la acción de control GPC difusa está dada por:

$$\Delta u(t) = \frac{\sum_{i=1}^M w_i \Delta u_i(t)}{\sum_{i=1}^M w_i}$$

- **Desventajas del controlador**

Consecuencia del modelo difuso regla  $R_i$ :

$$\begin{aligned}
y_i(t) = & a_1^i y(t-1) + \dots + a_{n_y}^i y(t-n_y) \\
& + b_1^i u(t-1) + \dots + b_{n_u}^i u(t-n_u) + c^i
\end{aligned}$$

Para cada regla se minimiza la siguiente función objetivo:

$$J_i = \sum_{j=N_1}^{N_2} [w(t+j) - \hat{y}_i(t+j|t)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2$$

donde  $\hat{y}_i(t+j|t)$  es la predicción a  $j$  pasos con el modelo lineal de la regla  $i$ .

Se debiese minimizar la siguiente función objetivo:

$$J_i = \sum_{j=N_1}^{N_2} [w(t+j) - \hat{y}_i(t+j|t)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2$$

donde  $\hat{y}_i(t+j|t)$  es la predicción a  $j$  pasos con el modelo difuso completo.

Sin embargo, la solución del GPC difuso (Cipriano & Ramos, 1995) es una buena aproximación y es de fácil y rápida implementación.

- **Control predictivo difuso, según Roubos (1998)**

Controlador predictivo basado en la linealización del modelo difuso de Takagi-Sugeno

$$R_i : \text{Si } x_1(t) \text{ es } A_{1i} \text{ y } \dots \text{ y } x_n(t) \text{ es } A_{ni} \\ \text{entonces } x^i(t+1) = A^i x(t) + B^i u(t) + C^i$$

$$x(t+1) = \sum_{i=1}^M w_i x^i(t+1)$$

con  $w_i$  es el grado de activación normalizado.

a) Modelo lineal variante en el tiempo equivalente

$$x(t+1) = A(t)x(t) + B(t)u(t) + C(t)$$

$$\text{con } A(t) = \sum_{i=1}^n w_i(t)A^i, \quad B(t) = \sum_{i=1}^n w_i(t)B^i, \quad C(t) = \sum_{i=1}^n w_i(t)C^i.$$

- b) En cada instante o período de muestreo, se deriva un modelo lineal, evaluando las premisas del modelo difuso o grados de activación.
- c) Para cada modelo lineal resultante se diseña un controlador predictivo lineal.
- d) En el siguiente instante, se actualiza el modelo lineal

Como ventajas se pueden mencionar que es un controlador fácil y de rápida implementación y como desventaja, que entrega una solución sub-óptima.

- **Control predictivo difuso, según Cipriano & Sáez (1996)**

Modelo difuso de Takagi-Sugeno con un modelo ARIX

$$R_i : \text{Si } y(t-1) \text{ es } A_{1i} \text{ y } \dots \text{ y } y(t-n_y) \text{ es } A_{n_{yi}} \text{ y} \\ u(t-1) \text{ es } B_{1i} \text{ y } \dots \text{ y } u(t-n_u) \text{ es } B_{n_{ui}} \\ \text{entonces } A_i(z^{-1})y_i(t) = B_i(z^{-1})u(t-1) + \frac{e_i(t)}{\Delta}$$

Se deriva la predicción lineal para cada modelo lineal de cada regla:

$$R_i : \text{Si } y(t-1) \text{ es } A_{1i} \text{ y } \dots \text{ y } y(t-n_y) \text{ es } A_{n_{yi}} \text{ y} \\ u(t-1) \text{ es } B_{1i} \text{ y } \dots \text{ y } u(t-n_u) \text{ es } B_{n_{ui}} \\ \text{entonces } \hat{y}_i = G_i \Delta u + f_i$$

donde  $\hat{y}_i = [\hat{y}_i(t + N_1 | t), \dots, \hat{y}_i(t + N_2 | t)]$

Entonces, la predicción global está dada por:

$$\hat{y} = \frac{\sum_{i=1}^M w_i(t) \hat{y}_i}{\sum_{i=1}^M w_i(t)} = \frac{\sum_{i=1}^M w_i(t) [G_i \Delta u + f_i]}{\sum_{i=1}^M w_i(t)}$$

con  $w_i(t)$  es el grado de activación y  $M$  es el número de reglas.

Se considera la siguiente función de costos:

$$J = (w - \hat{y})^T (w - \hat{y}) + \lambda \Delta u^T \Delta u$$

con  $w$  es el vector de referencias futuras,  $\hat{y}$  es el vector de predicciones y  $\Delta u$  es el vector de acciones de control futuras.

Sustituyendo el predictor difuso en la función objetivo se tiene:

$$J = \left( w - \frac{\sum_{i=1}^M w_i(t) [G_i \Delta u + f_i]}{\sum_{i=1}^M w_i(t)} \right)^T \left( w - \frac{\sum_{i=1}^M w_i(t) [G_i \Delta u + f_i]}{\sum_{i=1}^M w_i(t)} \right) + \lambda \Delta u^T \Delta u$$

Resolviendo el problema de minimización  $\min_{\Delta u} J$ , se obtiene el controlador

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (w - f)$$

$$\text{con } G = \frac{\sum_{i=1}^M w_i(t) G_i}{\sum_{i=1}^M w_i(t)}, \quad f = \frac{\sum_{i=1}^M w_i(t) f_i}{\sum_{i=1}^M w_i(t)}$$

Como ventajas se pueden mencionar que es un controlador fácil y de rápida implementación y además es solo una mejor aproximación del óptimo global.

### 3.6. Problemas Resueltos

#### Problema 3.1 (C2 – Otoño, 2013)

Un generador eólico funciona mediante el movimiento del rotor, el cual se logra cuando el viento mueve las aspas; sin embargo, es necesario controlar la velocidad de giro a una frecuencia constante. Para esta planta no lineal se desea diseñar una estrategia de control predictivo difuso. Se asume el siguiente modelo del proceso:

$$y(k) = \frac{y(k-1)y(k-2)}{1+u(k-1)^2} + w(k)$$

con  $u(k)$  la variable manipulada, y  $w(k)$  ruido blanco discreto. Para esto, se pide:

- Explicar cómo implementaría el método de fuzzy clustering para este problema. ¿Cuál es su finalidad? (0.5 pts.)
- Construir un modelo dinámico difuso de T&S (Entradas y salidas del modelo, clusters, reglas e identificación de parámetros), explicar detalladamente. (2.0 pto)
- ¿De qué manera se puede obtener el número óptimo de clusters y conjuntos difusos para un modelo determinado? (0.5 pto.)
- Describir un método para determinar las variables de entrada relevantes al modelo difuso de T&S considerado en b) (0.5 pto.)
- Explicar cómo implementar un controlador lineal basado en el modelo T&S. (1 pto.)
- Diseñar un controlador predictivo difuso con solución global que minimice la siguiente función objetivo:

$$J = \sum_{j=1}^N \left[ (\hat{y}(k+j) - r)^2 + \lambda u^2(k+j-1) \right]$$

Plantear ecuaciones y definir variables de optimización. (1.0 pts)

- Desarrollar un diagrama de bloques para implementar la solución f). (0.5 pto)

## Solución

a) Para implementar fuzzy clustering se tomarían datos de entrada y salida del sistema, teniendo en consideración que la entrada logre abarcar la mayor cantidad de puntos de operación posible. La finalidad es lograr encontrar una representación del sistema mediante clusters, para luego realizar algún método de identificación y/o control.

b) Sean las entradas del modelo  $y(k-1)$ ,  $y(k-2)$  y  $u(k-1)$ , y la salida  $y(k)$ . (Se podrían haber puesto  $n$  entradas, pero solo si se explicaba el método de análisis de sensibilidades para obtener las entradas “correctas” al modelo).

Se suponen  $C_y$  clusters para la variable  $y(k)$  y  $C_u$  clusters para  $u(k)$ , con funciones de pertenencia  $\mu_y = f(y(k))$  y  $\mu_u = f(u(k))$ , respectivamente.

Por lo tanto si se tienen reglas de la forma:

$$R_i : \text{si } y(k-1) \text{ es } A_{1i} \text{ \& si } y(k-2) \text{ es } A_{2i} \text{ \& si } u(k-1) \text{ es } B_{1i}, \\ \Rightarrow y_i(k) = \alpha_i y(k-1) + \beta_i y(k-2) + \gamma_i u(k-1)$$

Por lo tanto se tendrán  $C_y^2 C_u$  reglas. Además el parámetro constante se omite.

Para identificar los parámetros, se define  $B_{ij} = \frac{W_{ij}}{\sum_{k=1}^M W_{kj}}$ , donde

$$W_{ij} = \min(\mu_{A_{1i}}(y(j-1)), \mu_{A_{2i}}(y(j-2)), \mu_{B_{1i}}(u(j-1)))$$

Se tienen  $M$  reglas y  $N$  muestras, donde  $1 \leq i \leq M$  y  $2 \leq j \leq N$

Se define:

$$x = \begin{bmatrix} y(1)B_{11} \dots y(1)B_{M1} & y(0)B_{11} \dots y(0)B_{M1} & u(1)B_{11} \dots u(1)B_{M1} \\ \vdots & \vdots & \vdots \\ y(N-1)B_{1N} \dots y(N-1)B_{MN} & y(N-2)B_{1N} \dots y(N-2)B_{MN} & u(N-1)B_{1N} \dots u(N-1)B_{MN} \end{bmatrix}$$

Se calculan los parámetros como  $P = (x^T x)^{-1} x^T y$ .

c) El número óptimo de reglas y conjuntos difusos del modelo se determina haciendo una partición del universo de la variable de salida y luego proyectándolo al espacio de entrada.

d) Mediante análisis de sensibilidad, para cada variable de entrada al modelo. Esto se realiza calculando la derivada parcial de la salida con respecto a cada variable, se fija un umbral y si la derivada es menor a ese umbral se elimina del modelo la entrada relacionada con esa derivada.

e) A cada regla se le asocia un controlador y la salida del controlador difuso viene dada por:

$$u = \sum W_i u_i / \sum W_i$$

f) Para minimizar la función globalmente, se debe notar que la salida viene dada por:

$$y(k) = \frac{\sum W_i y_i}{\sum W_i} = \frac{\sum W_i (\alpha_i y_i(k-1) + \beta_i y_i(k-2) + \gamma_i u_i(k-1))}{\sum W_i}$$

Entonces la predicción es:

$$\hat{y}(k+1) = \frac{\sum W_i \hat{y}_i}{\sum W_i} = \frac{\sum W_i (\alpha_i \hat{y}_i(k) + \beta_i \hat{y}_i(k-1) + \gamma_i u_i(k))}{\sum W_i} = \frac{\sum W_i [G_i u + f_i]}{\sum W_i}.$$

El control para cada regla es:

$$u_i = [G_i^T G_i + \lambda I]^{-1} G_i^T (r - f_i)$$

g) El diagrama de bloques debe contener la fusificación del problema (no es necesario retro alimentarlo con los retardos de la salida), un optimizador que resuelva el problema de encontrar el mínimo de J.

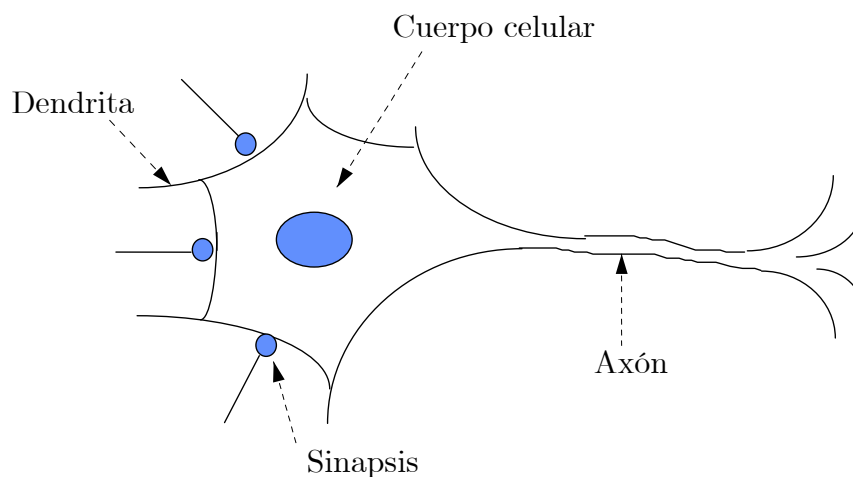
## Capítulo 4

### 4. Introducción a Redes Neuronales

Las redes neuronales constituyen una poderosa herramienta para modelar sistemas, especialmente no lineales, sean dinámicos o estáticos.

El cerebro humano es un sistema muy complejo formado por muchas células llamadas neuronas; se estima que existen entre  $10^{10}$  y  $10^{11}$  de células en el cerebro. Las redes neuronales artificiales emulan la arquitectura y capacidades de sistemas neuronales biológicos.

Un esquema simplificado de una neurona se muestra en la siguiente figura



**Figura 4.1: Esquema biológico de una neurona**

En el cuerpo celular se realizan la mayoría de las funciones lógicas de la neurona. El axón es el canal de salida final de la neurona. Las dendritas reciben las señales de entrada de los axones de otras neuronas y se conectan al cuerpo celular por medio de las sinapsis.



#### 4.1. Representación Matemática de una Neurona

Una neurona o nodo o unidad, es un elemento de procesamiento que toma un número de entradas, los pondera (pesos), los suma y utiliza el resultado como el argumento de una función, llamada función de activación. En la figura 4.2 se observa la estructura de una neurona artificial con múltiples entradas.

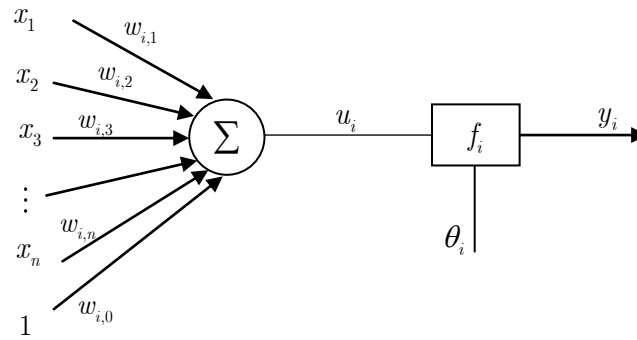


Figura 4.2: Esquema de una neurona artificial

En esta estructura, se tiene  $u_i = \sum_{j=1}^n w_{i,j} x_j + w_{i,0}$

donde

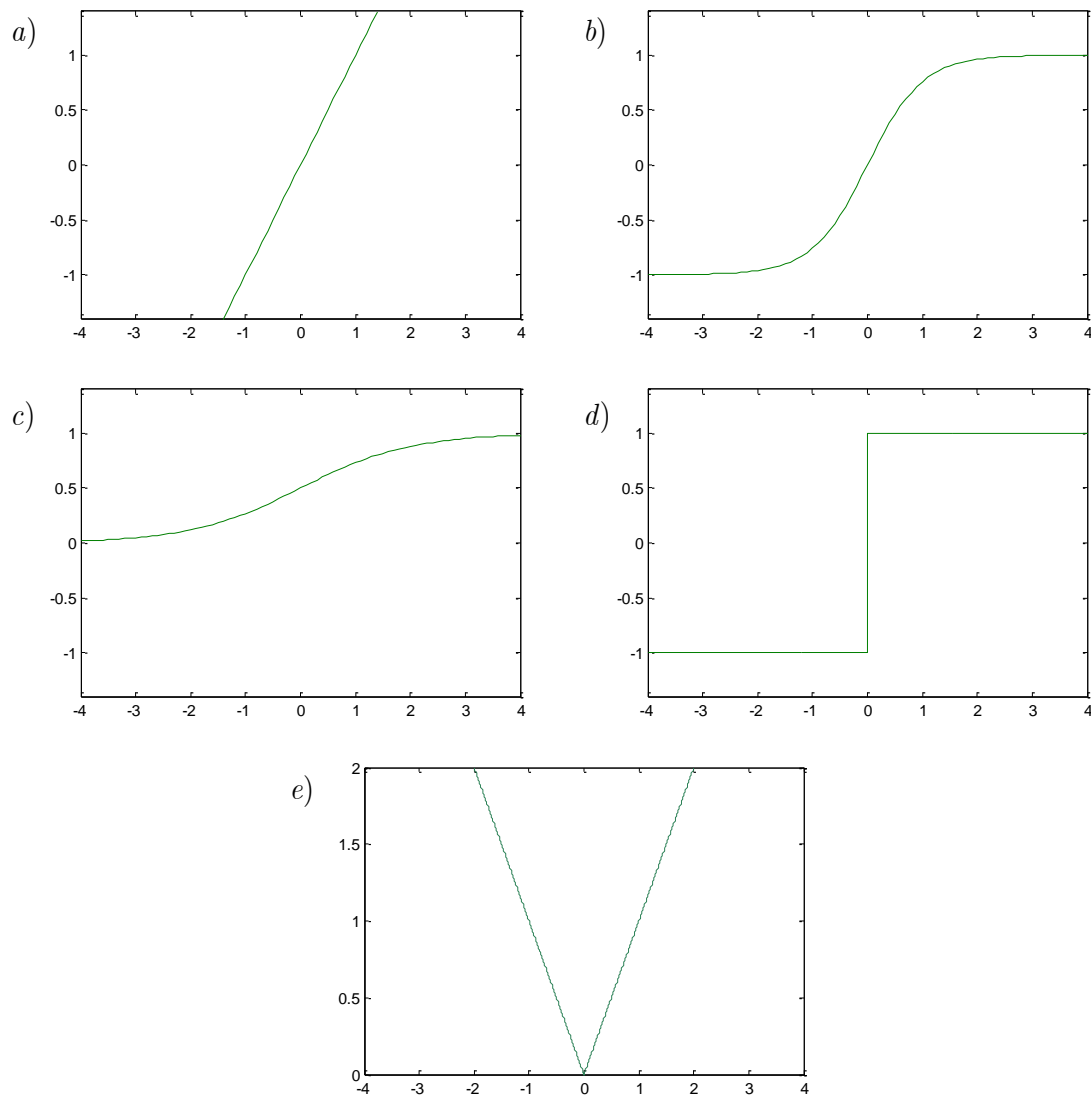
- $w_{i,j}$  son los pesos de la neurona  $i$  (sinápsis)
- $x_i$  son las entradas a la neurona
- $n$  es el número de entradas a la neurona

$$y_i = f_i(u_i) = f_i \left( \sum_{j=1}^n w_{i,j} x_j + w_{i,0} - \theta \right)$$

donde

- $y_i$  es la salida de la neurona (axón)
- $f_i$  es la función de activación, la cual puede tomar cualquier forma (cuerpo celular)
- $\theta_i$  es el sesgo

En general, se utilizan las siguientes funciones de activación en la figura 4.3:



**Figura 4.3:** Cinco diferentes funciones de activación; *a)* lineal,  $f(x) = x$ ; *b)* tangente hiperbólica,  $f(x) = \tanh(x)$ ; *c)* sigmoidea,  $f(x) = 1 / (1 + \exp(-x))$ ; *d)* función signo,  $f(x) = \text{sgn}(x)$ ; *e)* valor absoluto  $f(x) = |x|$

Las redes neuronales son estructuras de procesamiento formadas por una gran cantidad de neuronas, que operan en paralelo. Además, los distintos tipos de redes neuronales se generan a partir de la interconexión de neuronas.

Las principales redes neuronales que se utilizan para modelación no lineal son:

- Redes perceptrón multicapa
- Redes recurrentes
- Redes de funciones de base radiales (RBFN)

## 4.2. Ventajas de las Redes Neuronales

Las redes neuronales deben su capacidad de procesamiento de información a su estructura distribuida y paralela, a su capacidad de aprendizaje y por tanto de generalización.

### Tareas

- Reconocimiento de patrones
- Memorias asociativas
- Aproximación funcional
- Entre otras.

### Propiedades

- **No linealidad.** Las neuronas son elementos de proceso generalmente no lineales. La interconexión de estos elementos genera estructuras de transformación de datos donde este carácter no lineal queda distribuido a lo largo y ancho de la red.
- Modelado de relaciones de entrada/salida.
- **Adaptabilidad.** Las redes neuronales son por definición estructuras adaptativas capaces de ajustar sus pesos, y por tanto su función de transferencia, a cambios en su entorno.
- **Tolerancia ante fallos.** Una red neuronal tiene la capacidad de seguir respondiendo de forma no catastrófica cuando parte de su estructura no está dañada. Esto es debido al tratamiento distribuido de la información y a la redundancia implícita en su estructura.

### 4.3. Perceptrón Multicapa

El perceptrón multicapa es una estructura jerárquica que consiste en varias capas de neuronas totalmente interconectadas, que admiten como entradas las salidas de los elementos de proceso (neuronas) de la capa anterior.

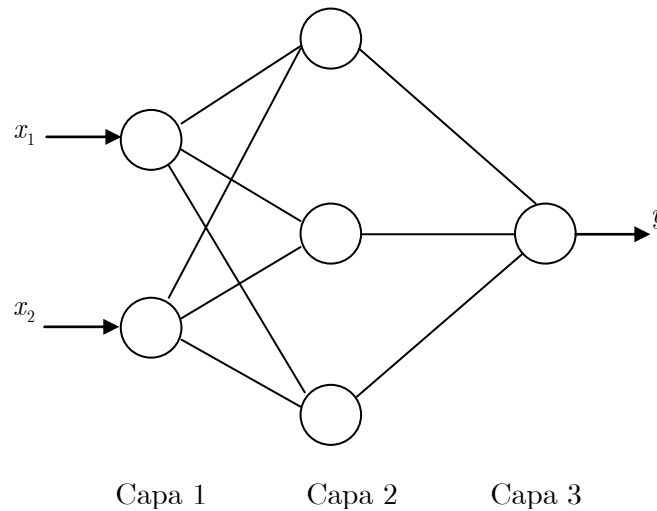


Figura 4.4: Perceptrón multicapa

En las redes perceptrón multicapa se distinguen tres tipos de capas:

- Capa de entrada (1): Está formada por  $n$  unidades (siendo  $n$  el número de entradas externas) que se limitan a distribuir las señales de entrada a la capa siguiente.
- Capa oculta (2): Están formadas por neuronas que no tienen contacto físico con el exterior. El número de capas ocultas es variable, pudiendo incluso ser nulo.
- Capa de salida (3): Está formado por  $m$  neuronas (siendo  $m$  el número de salidas externas) cuyas salidas constituyen el vector de salidas externas del perceptrón multicapa.

Los *modelos dinámicos neuronales* están dados por:

$$y(t) = N(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))$$

donde  $N$  es la red neuronal que puede ser un perceptrón multicapa, como se muestra en la siguiente figura 4.5.

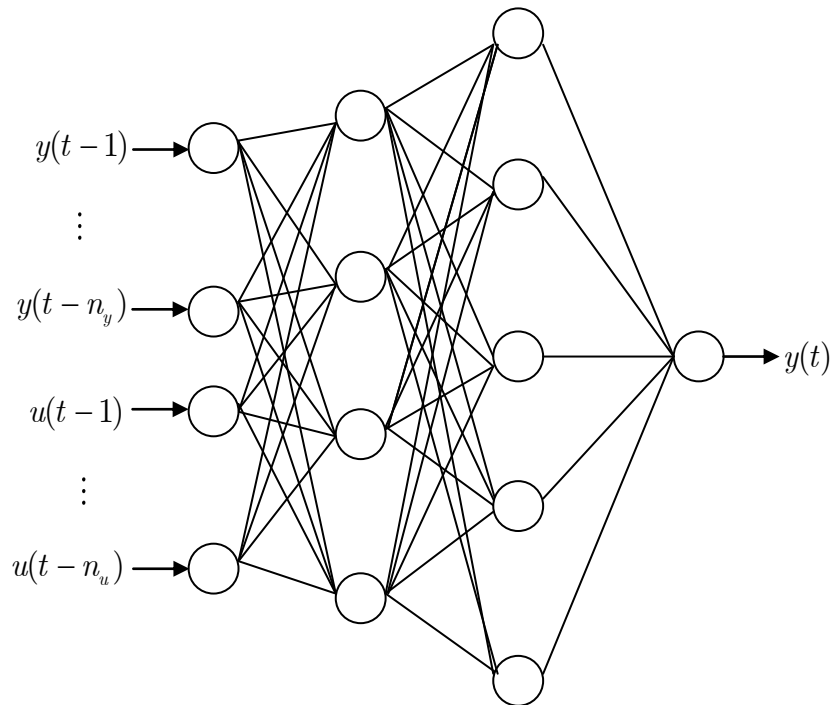


Figura 4.5: Perceptrón multicapa generalizado.

### Aplicaciones

- Aproximación funcional
- Reconocimiento de patrones
- Filtrado de señales
- Eliminación de ruido
- Segmentación de imágenes y señales
- Control adaptivo
- Compresión de datos
- Etc.

### Ventajas

- Capacidad de representación funcional universal. Gran rapidez de procesamiento. Genera buenas representaciones internas de las características de los datos de entrada. Ampliamente estudiada. Es la red neuronal más aplicada en la práctica

### Desventajas

- Tiempo de aprendizaje elevado para estructuras complejas

**Ejemplo.** Modelación de la química del agua de una central térmica utilizando redes neuronales.

Se considera la central térmica a carbón Anllares (350 MW), propiedad de la empresa Unión Eléctrica Fenosa (UEFSA), España. Esta central tiene en operación un sistema experto denominado SEQA que permite adquirir variables relacionadas con las propiedades químicas de los siguientes flujos del ciclo agua-vapor: vapor condensado, agua de alimentación, vapor saturado, vapor sobrecalentado y vapor recalentado.

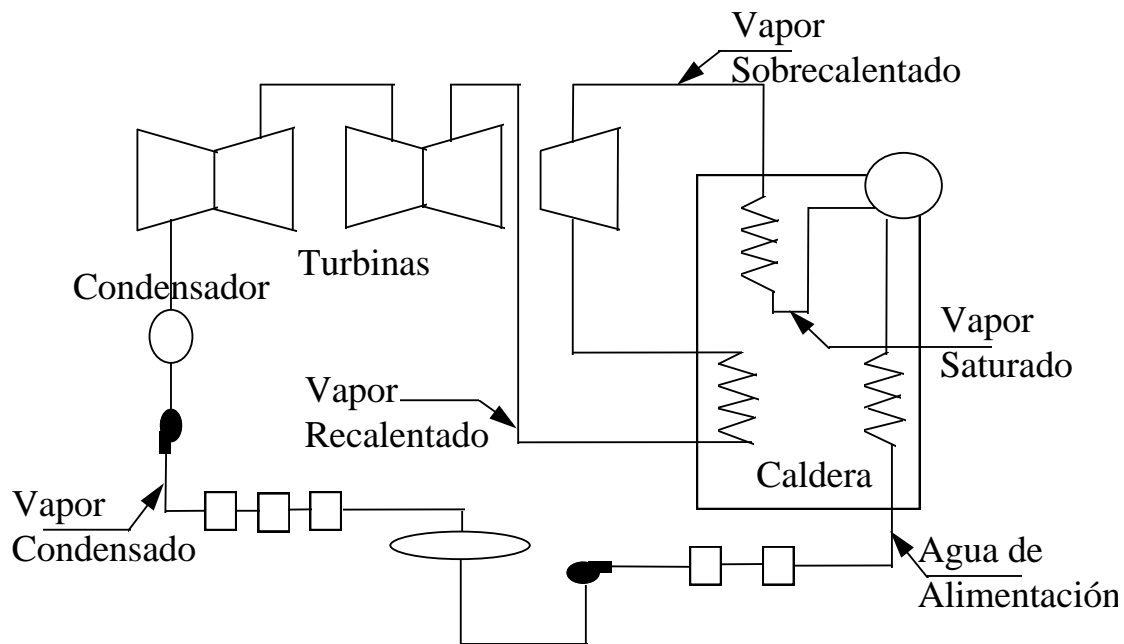


Figura 4.6: Diagrama de la central térmica Anllares.

Las propiedades químicas analizadas para los flujos considerados son: la conductividad catiónica, la conductividad específica, el pH y el porcentaje de  $O_2$ . La utilización de modelos predictivos para estas propiedades químicas, en el sistema experto SEQA, permite controlar los problemas de corrosión de componentes presentes en la producción de energía eléctrica. Especialmente, es importante la modelación de la conductividad catiónica del ciclo agua-vapor, debido a que esta propiedad es muy representativa de las impurezas del agua.

Como ejemplo de la modelación neuronal de las propiedades químicas del agua, se presentan los resultados obtenidos para la modelación de la conductividad catiónica del agua de alimentación ( $CC_{aa}$ ). Las variables de entrada al modelo son: la potencia generada de la central ( $P$ ) y la conductividad catiónica del condensado ( $CC_{cond}$ , flujo precedente). Los datos son adquiridos con un período de muestreo de 15 minutos.

El modelo neuronal para la conductividad catiónica del agua de alimentación está dada por:

$$CC_{aa}(k) = N(CC_{aa}(k-1), P(k-1), P(k-2), CC_{cond}(k), CC_{cond}(k-1))$$

donde  $N$  es un perceptrón multicapa con una capa oculta de neuronas de funciones de activación tangente hiperbólica y una capa de salida lineal.

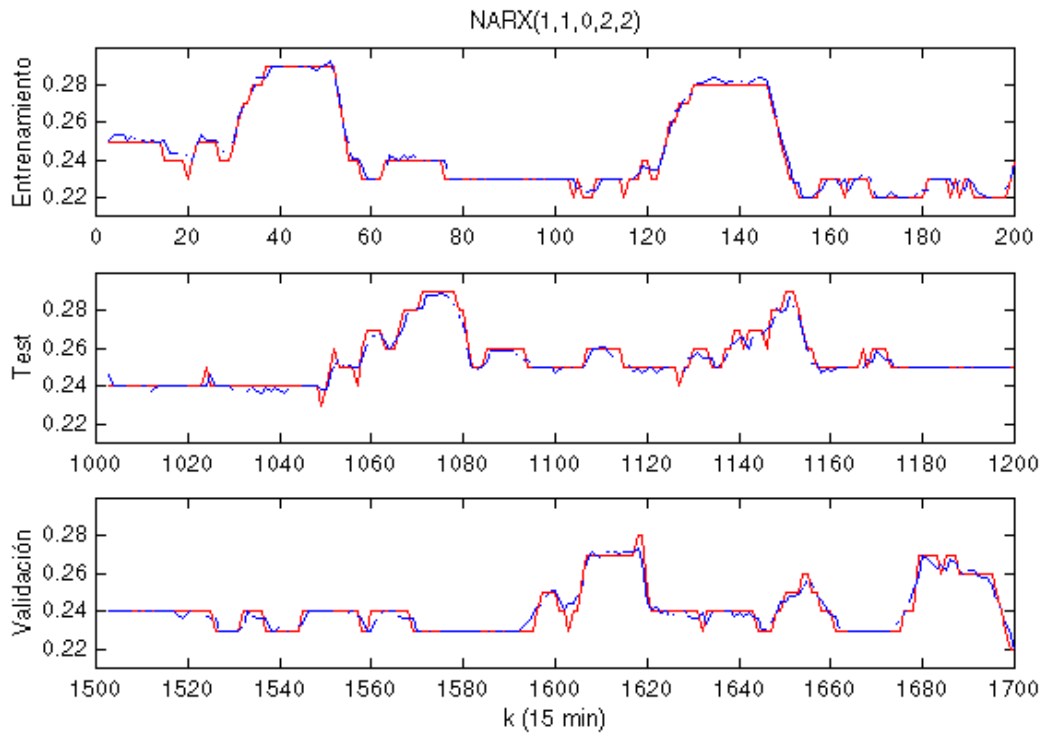


Figura 4.7: Modelo NARX para el proceso.

#### 4.4. Redes Recurrentes

Estos modelos son capaces de representar sistemas realimentados dinámicos no lineales. (Narendra, 1990).

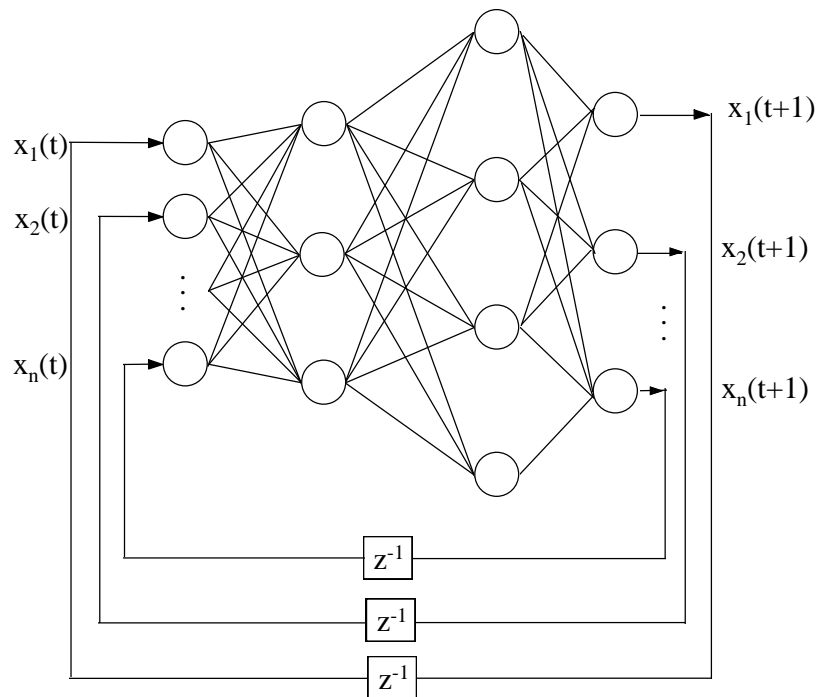


Figura 4.8: Esquema de redes recurrentes

Además, se debe mencionar que existen diversos modelos neuronales que son combinaciones de las redes perceptrón multicapa y redes recurrentes.



#### 4.5. Redes de Funciones de Base Radiales

Las redes de funciones de base radiales (RBFN “Radial Basis Function Networks”) consisten en dos capas (Jang, 1993). Los modelos dinámicos basados en las redes RBFN están dados por:

$$y(t) = N(y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u))$$

donde  $N$  es una red neuronal como se muestra en la siguiente figura con  $n = n_y + n_u$ .

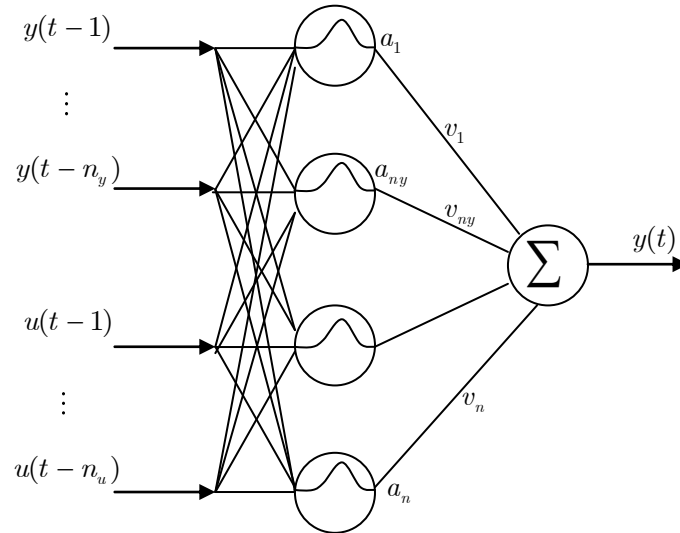


Figura 4.9: Esquema de una RBFN

La capa oculta está compuesta por  $n$  unidades radiales totalmente conectadas al vector de entrada. Las funciones de transferencia de la capa oculta son similares a una función de densidad gaussiana, es decir:

$$a_i = \exp\left(-\frac{\|x - r_i\|^2}{\sigma_i^2}\right)$$

donde  $x = [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]$  es el vector de entradas de la red,  $r_i$  son los centros de las unidades radiales,  $\sigma_i$  representan los anchos.

La salida de la red está dada por:

$$y(t) = \sum_{i=1}^n v_i a_i$$

donde  $v_i$  son los pesos de las unidades radiales.

**Ejemplo** Modelación neuronal basada en RBFN para un fermentador batch de alimentación.

La presión en el estanque de fermentación puede ser controlada a través del cambio de flujo de aire de salida manteniendo constante el flujo de aire de entrada.

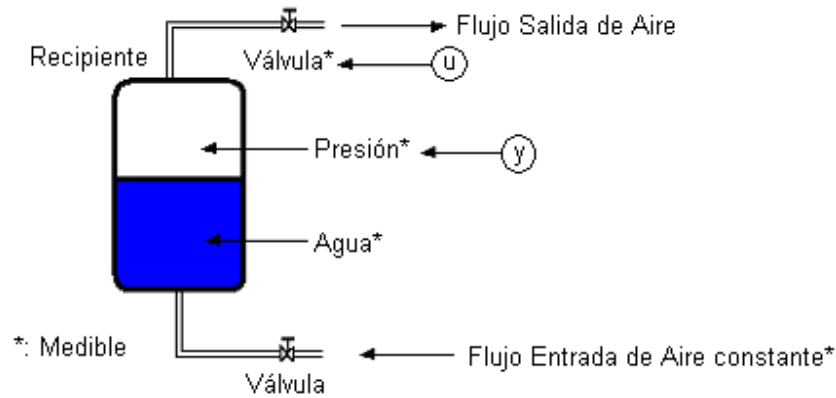


Figura 4.10: Diagrama de un fermentador Batch

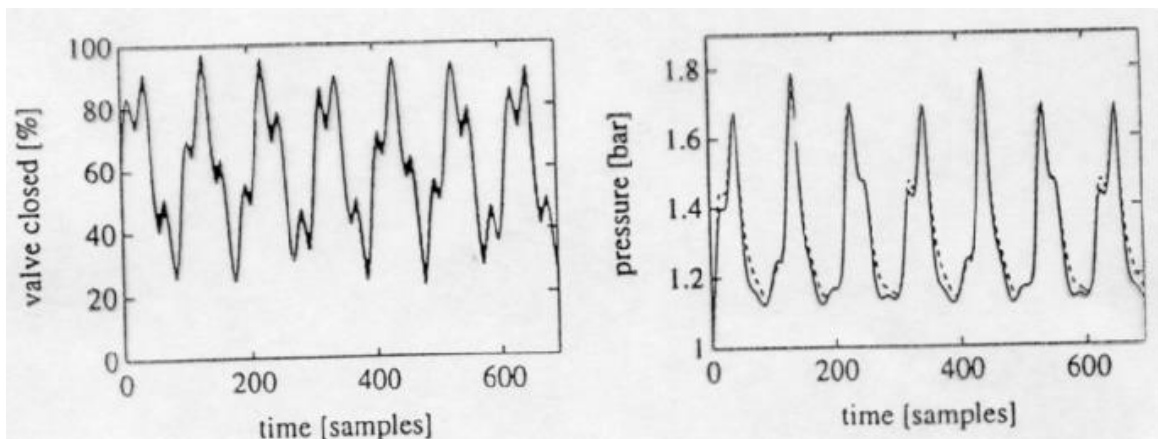
El modelo de la red está dado por:

$$\hat{y}(k+1) = N(y(k), u(k))$$

donde  $y(k)$  es la presión en el estanque y  $u(k)$  es el flujo de salida. Además,  $N$  es una red neuronal lineal/RBF dada por las siguientes ecuaciones:

$$y(k+1) = w_0 + \sum_{i=1}^n w_{1i} \phi_i r_i(k) + w_2^T x(k)$$

Con  $r_i(k) = \|x(k) - c_i\|$  y  $x(k) = [y(k), \dots, y(k-n_y), u(k), \dots, u(k-n_u)]^T$



## Aplicaciones

- Aproximación funcional
- Reconocimiento de patrones

## Ventajas

Capacidad de representación funcional universal. La estructura de esta red tiene interpretación directa, lo que permite realizar una buena inicialización de los pesos de la red, y extraer conocimiento de las estructuras ajustadas. La buena inicialización de los pesos acelera el proceso de aprendizaje.

## Desventajas

El procesamiento realizado es algo más complejo que en el caso del perceptrón multicapa.

### 4.6. Otros tipos de Redes

**Las redes Adaline.** Estas neuronas tienen capacidad de aprendizaje debido a que sus pesos son cambiados adaptivamente de acuerdo a un algoritmo adaptivo. Sus aplicaciones principales son: filtrado adaptivo de señales, reconocimiento de patrones. Son fácilmente implementables en hardware debido a su sencillez y homogeneidad, sin embargo sólo son capaces de resolver problemas de clasificación linealmente separables y llevar a cabo transformaciones lineales.

**Mapas auto organizativos de Kohonen.** En este caso, las neuronas están ordenadas topológicamente. Frente a la presentación de un patrón  $n$ -dimensional de entrada, compiten lateralmente hasta que sólo una de ellas queda activa. El objetivo es que patrones de entrada con características parecidas queden asociados a neuronas topológicamente cercanas. Sus principales aplicaciones son: agrupación y representación de datos, compresión de datos y optimización.

### 4.7. Entrenamiento de Redes Neuronales

Se entiende por entrenamiento el cálculo de pesos y sesgos de manera que la red se comporte de una manera deseada. De acuerdo al tipo de entrenamiento, las redes se pueden subdividir en dos grandes grupos:

- **Redes con entrenamiento supervisado.** Estas redes se entrenan presentando, para cada combinación de entradas, las salidas que se espera ellas produzcan. Los algoritmos de entrenamiento calculan pesos y sesgos

nuevos de manera de minimizar el error entre la salida deseada y la obtenida realmente.

- **Redes sin supervisión.** Los algoritmos de entrenamiento calculan nuevos pesos libremente. Estas redes se utilizan como clasificadores, pues se caracterizan por asociar una combinación de entradas específica con una sola salida.

#### 4.7.1. La regla delta

La regla delta es una regla de aprendizaje para una red de solo una capa. Es un precursor para el algoritmo multicapa backpropagation. La idea es definir una medida de todas las funciones de una red encontrando un camino para optimizar el rendimiento.

#### 4.7.2. Algoritmo de entrenamiento backpropagation

El algoritmo de entrenamiento backpropagation se utiliza para ajustar los pesos y sesgos de una red, con el fin de minimizar la suma del cuadrado de los errores de la red. Es un método iterativo de optimización de descenso según el gradiente, cuyos detalles se presentan a continuación.

Es lógico que el algoritmo de aprendizaje debe cambiar los pesos para que la salida  $o^q$  sea lo más parecido a la salida objetivo  $y^q$ , para los patrones de entrenamiento  $q = 1, 2, \dots, p$ .

La medida adecuada al rendimiento total es:

$$E = \sum_{q=1}^N E^q$$

Donde

$$E^q = \frac{1}{2} \sum_{i=1}^p (y_i^q - o_i^q)^2$$

El objetivo es minimizar  $E$  con respecto a los pesos  $w_{ij}$  de la red.

Sea  $w_{ij}$  el cual denota el peso desde el nodo  $j$  a la salida de la neurona  $i$ . Para utilizar el método del gradiente descendente para la optimización, se necesita que los pesos  $w_{ij}$  sean diferenciables. Esto requiere que lo siguiente sea diferenciable:

$$o_i^q = f_i \left( \sum_{j=0}^n w_{ij} x_j^q \right)$$

Donde  $f_i$  es la función de activación de la  $i^{th}$  neurona.

El error  $E$  es una función de variables  $w_{ij}$ . Recordemos que el gradiente  $\Delta E$  de es un vector de derivadas parciales  $\partial E / \partial w_{ij}$ . Así para minimizar el error, se debe mover proporcionalmente  $\Delta E$  a lo negativo, lo que lleva a la actualización de los pesos como:

$$w_{jk}^q \rightarrow w_{jk}^{q*} + \Delta^q w_{jk}$$

Con  $w_{jk}^{q*}$  los pesos anteriores.

Donde

$$\Delta^q w_{jk} = -\eta \frac{\partial E^q}{\partial w_{jk}}$$

con  $\eta > 0$ , el cual es llamado la tasa de aprendizaje.

Se tiene que:

$$\frac{\partial E^q}{\partial w_{ik}} = \frac{\partial E^q}{\partial o_i^q} \frac{\partial o_i^q}{\partial w_{ik}}$$

con,

$$o_j^q = f_j \left( \sum_{i=0}^n w_{ji} x_i^q \right)$$

y además,

$$\frac{\partial o_i^q}{\partial w_{ik}} = f_i' \left( \sum_{l=0}^n w_{il} x_l^q \right) x_k^q$$

$$\text{Sea } \delta_j^q = \frac{\partial E^q}{\partial o_j^q}$$

$$\delta_j^q = \frac{\partial E^q}{\partial o_j^q} = \sum_{j=1}^p \frac{\partial E^q}{\partial o_j^q} \frac{\partial o_j^q}{\partial o_i^q}$$

Para la  $j^{th}$  salida de la neurona.

Es decir, los valores de  $\delta$  para la capa de salida:

$$\delta_i^q = f_i' \left( \sum_{k=1}^n v_{ik} z_k \right) (o_i^q - y_i^q)$$

Donde  $z_k$  es la salida de la capa oculta y  $v_{ik}$  son los pesos que conectan a la neurona oculta  $k$  y la salida de la neurona  $i$ .

Y los valores de  $\delta$  para las capas ocultas por propagación:

$$\delta_i^q = f'_i \left( \sum_{k=0}^n w_{ik} x_k^q \right) \sum_{j=1}^p v_{ij} \delta_j^q$$

#### 4.8. Identificación Basada en Redes Neuronales

Cuando se trata de identificar un modelo de un sistema dinámico es una práctica común de seguir el procedimiento representado en la figura 4.10.

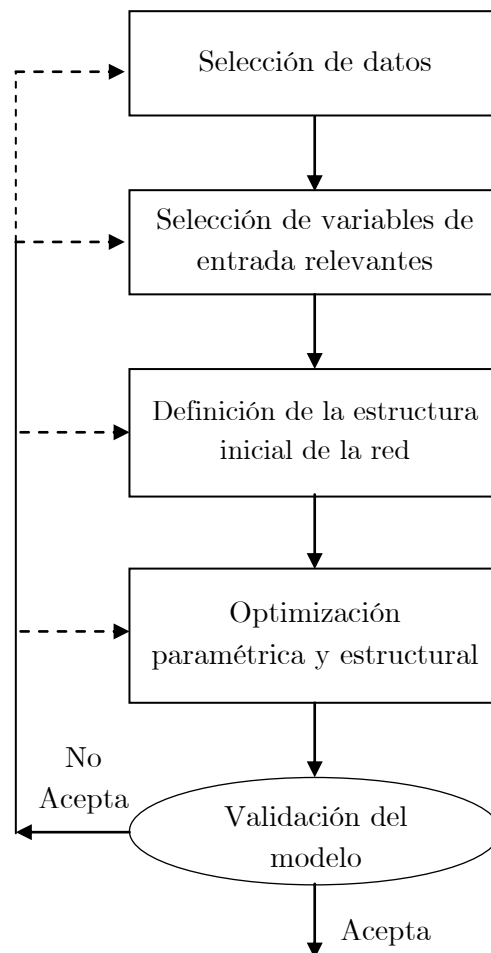


Figura 4.11: Sistema básico del procedimiento de identificación

##### i. Selección de datos

Para la identificación del modelo, es necesario seleccionar datos reales procedentes del proceso. Esos datos deben incluir información suficiente para representar las diferentes condiciones de operación del proceso. Los datos necesarios para el proyecto de modelado no lineal basado en redes neuronales son:

- **Conjunto de entrenamiento:** De estos datos, los parámetros del modelo se obtienen minimizando el error cuadrático medio.
- **Conjunto de test:** Este conjunto adicional se utiliza también en la formación algoritmo. El conjunto de datos no participa directamente en el modelo de ajuste, sin embargo, permite evaluar el modelo generalizado bajo un nuevo conjunto de datos.
- **Conjunto de validación:** Necesarios nuevos datos para evaluar la conducta apropiada del modelo ajustado.

## ii. Selección de variables de entrada relevantes

Para cualquier modelado de procesos, uno de los puntos más importantes es la selección adecuada de las variables de entrada relevantes ( $X_i : y(k-1), y(k-2), \dots, y(k-n_a), u_1(k-n_k), \dots, u_1(k-n_{b1}-n_{k1}+1), \dots, u_{nu}(k-n_{knu}), \dots, u_{nu}(k-n_{bnu}-n_{knu}+1)$ ) que se debe incluir en el modelo. En este trabajo, dos métodos de selección de variables se consideran.

- **Análisis convencional:** En este caso, varios modelos se ajustan, cada uno con diferentes variables de entrada. La elección de las variables de entrada final se basa en la evaluación y comparación de los diferentes modelos.
- **Análisis de sensibilidades:** Esta metodología consiste en el ajuste de un modelo inicial con las máximas posibles variables de entrada, buscando limitar la complejidad del problema. Entonces, las influencias o sensibilidades para cada variable de entrada se determinan. Por último, el modelo óptimo que utiliza sólo las variables de entrada con mayores sensibilidades asociadas se obtiene. Para un análisis más detallado dirigirse a la sección 3.4.3.2. de este apunte.

## iii. Definición de la estructura inicial de la red

Requiere la especificación del número de capas y su número de neuronas. La capa de entrada está dada por las variables de entrada y la capa de salida por las variables de salida. La estructura de capas ocultas se obtiene utilizando el algoritmo de optimización estructural.

Además, la estructura de la red neuronal depende de las peticiones. De esta manera, es posible proponer el uso de un perceptrón multicapa. Esta red es un

muy buen aproximador universal no lineal, y en general tiene un algoritmo de procesamiento rápido y fácil de implementar. Sin embargo, también es posible proponer una RBFN ("Red Radial Basis Function") de la red, debido a que es un aproximador universal muy bueno, y su principal ventaja es una interpretación matemática directa, pero su aplicación es compleja.

#### **iv. Optimización paramétrica y estructural**

La optimización estructural de una red neural es un procedimiento de búsqueda consiste en proponer arquitecturas diferentes, incrementando la complejidad (por ejemplo, aumentando el número de neuronas de las capas ocultas y sus capas de número). Entonces, para cada estructura propuesta, la optimización paramétrica se realiza minimizando el conjunto de entrenamiento del error cuadrático medio y evaluando el conjunto de prueba del error cuadrático medio. Finalmente, ambas optimizaciones concluyen cuando el error de la prueba disminuye.

#### **v. Validación del modelo**

El modelo ajustado se evalúa utilizando un conjunto de validación. Entonces, si la evaluación del modelo ajustado es apropiada, el proceso de identificación finaliza, en caso contrario es conveniente revisar la selección de variables relevantes para encontrar si alguna variable importante no está incluida.



## 4.9. Estructuras en identificación de redes neuronales

Para el proceso de identificación en redes neuronales, es necesario tener en claro que estructura se debe utilizar, para ello se muestran las siguientes:

### 4.9.1. Modelación directa

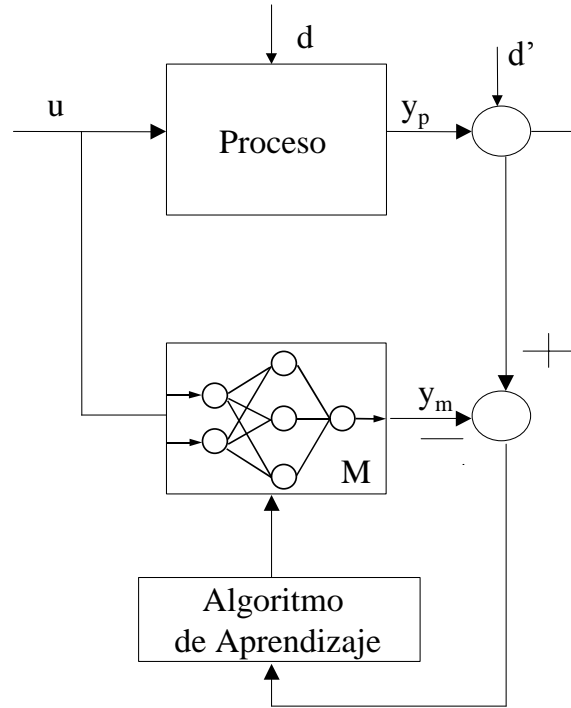


Figura 4.12: Esquema directo de aprendizaje

En este caso, se entrena una red neuronal de manera de obtener la dinámica directa de la planta. La red es colocada en paralela a la planta y el error entre el sistema y las salidas de la red son usados como entrada al entrenamiento (“Backpropagation”).

### 4.9.2. Ecuación del sistema no lineal (Proceso)

$$y_p(t+1) = f(y_p(t), \dots, y_p(t-n+1), u(t), \dots, u(t-n+1))$$

### 4.9.3. Red neuronal (Modelo)

$$y_m(t+1) = \hat{f}(y_p(t), \dots, y_p(t-n+1), u(t), \dots, u(t-n+1))$$

donde  $\hat{f}$  es la relación de entrada – salida dada por la red neuronal. Luego de un tiempo adecuado de entrenamiento, se tiene  $y_m \approx y_p$

De esta manera, la red se independiza de la planta, es decir:

$$y_m(t+1) = \hat{f}(y_m(t), \dots, y_m(t-n+1), u(t), \dots, u(t-n+1))$$

#### 4.9.4. Modelación inversa

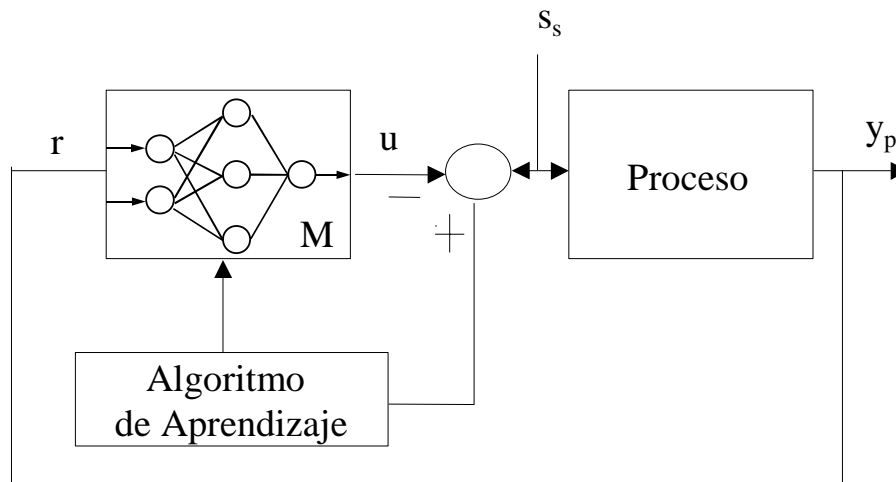


Figura 4.13: Estructura de modelación inversa.

$s_s$  es la señal de entrada para el entrenamiento.

Los modelos inversos de la dinámica de la planta juegan un rol importante en el diseño de control.

La salida  $y_p$  es usada como entrada a la red neuronal. La salida de la red  $u$  es comparada con la entrada del sistema  $s_s$  (señal de entrenamiento) y este error es usado para entrenar la red.

Esta estructura claramente tiende a forzar a la red neuronal a representar la dinámica inversa de la planta.

#### 4.9.5. Modelación inversa especializada

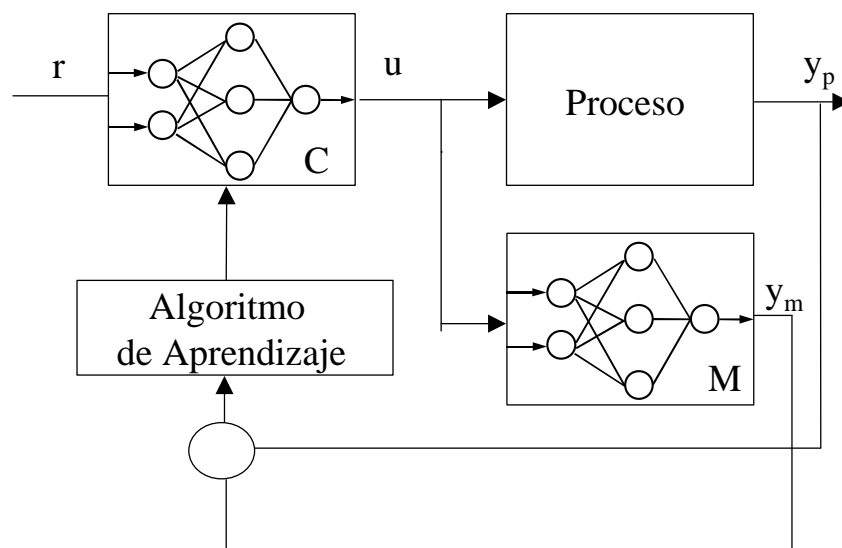


Figura 4.14: Estructura de modelación inversa especializada.

En este caso, el modelo red neuronal inverso precede al sistema y recibe como entrada la referencia deseada de la salida.

Esta estructura de aprendizaje contiene además un modelo red neuronal directo ( $M$ ).

La señal de error para el algoritmo de entrenamiento, en este caso, es la diferencia entre la señal entrenada  $y_m$  y la señal entrenada  $y_p$ .

Alternativamente, la señal de error puede ser la diferencia entre  $r$  y  $y_p$ .

La estructura entrada salida de la modelación del sistema inverso está dada por:

$$u(t) = f^{-1} \left( y_p(t), \dots, y_p(t+n-1), r(t+1), u(t-1), \dots, u(t-n+1) \right)$$

Si no se dispone de  $y_p$ ,

$$u(t) = f^{-1} \left( y_m(t), \dots, y_m(t+n-1), r(t+1), u(t-1), \dots, u(t-n+1) \right)$$

#### 4.10. Estructuras de control con redes neuronales

Existen diversas estructuras de control bien establecidas para sistemas no lineales (Hunt, 1992).

##### 4.10.1. Control por linealización instantánea

El enfoque más común de control para sistemas no lineales es aproximar el sistema por un sistema lineal en una región y luego diseñar una estrategia de control lineal.

Los modelos neuronales son estructuras no lineales que se pueden linealizar. Esta linealización puede ser en cada instante de muestreo y por eso se denomina linealización instantánea.

Consideremos el siguiente modelo neuronal  $y(t) = g(x(t))$ , donde  $x(t) = [y(t-1), \dots, y(t-n), u(t-1), \dots, u(t-d-m)]^T$ .

En el instante  $t = \tau$ , el sistema es linealizado alrededor del punto de operación  $x(\tau)$ , obteniéndose el siguiente modelo aproximado:

$$\tilde{y}(t) = -a_1 \tilde{y}(t-1) - \dots - a_n \tilde{y}(t-n) + b_0 \tilde{u}(t-1) + \dots + b_m \tilde{u}(t-d-m)$$

donde

$$\begin{aligned} \tilde{y}(t-i) &= y(t-i) - y(\tau-i) \\ \tilde{u}(t-i) &= u(t-i) - u(\tau-i) \end{aligned}$$

$$a_i = - \left. \frac{\partial g(x(t))}{\partial x_i} \right|_{t=\tau} \quad \text{para } 1 \leq i \leq n$$

$$b_i = - \left. \frac{\partial g(x(t))}{\partial x_{n+i+1}} \right|_{t=\tau} \quad \text{para } 1 \leq i \leq m$$

Para un modelo neuronal perceptrón multicapa con  $n_x$  entradas, una capa oculta con  $n_h$  neuronas con función de activación tangente hiperbólica y una capa lineal de salida, se tiene:

$$y(t) = \sum_{j=1}^{n_h} W_j \tanh \left( \sum_{k=1}^{n_x} w_{kj} x_k(t) + w_{0j} \right) + W_0$$

Entonces, la derivada de la salida con respecto a la entrada  $x_i(t)$  es:

$$\frac{\partial g(x(t))}{\partial x_i(t)} = \sum_{j=1}^{n_h} W_j w_{ji} \left( 1 - \tanh^2 \left( \sum_{k=1}^{n_x} w_{kj} x_k(t) + w_{0j} \right) \right)$$

Por lo tanto, el modelo aproximado está dado por:

$$y(t) = - \sum_{i=1}^n a_i y(t-i) + \sum_{i=0}^m b_i u(t-i) + \left( y(\tau) + \sum_{i=1}^n a_i y(\tau-i) - \sum_{i=0}^m b_i u(\tau-i) \right)$$

donde

$$\xi(\tau) = \left( y(\tau) + \sum_{i=1}^n a_i y(\tau-i) - \sum_{i=0}^m b_i u(\tau-i) \right)$$

es un término constante (bias).

A partir, de este modelo linealizado se deriva un controlador lineal, que es actualizado en el siguiente instante.

#### 4.10.2. Control por modelo de referencia

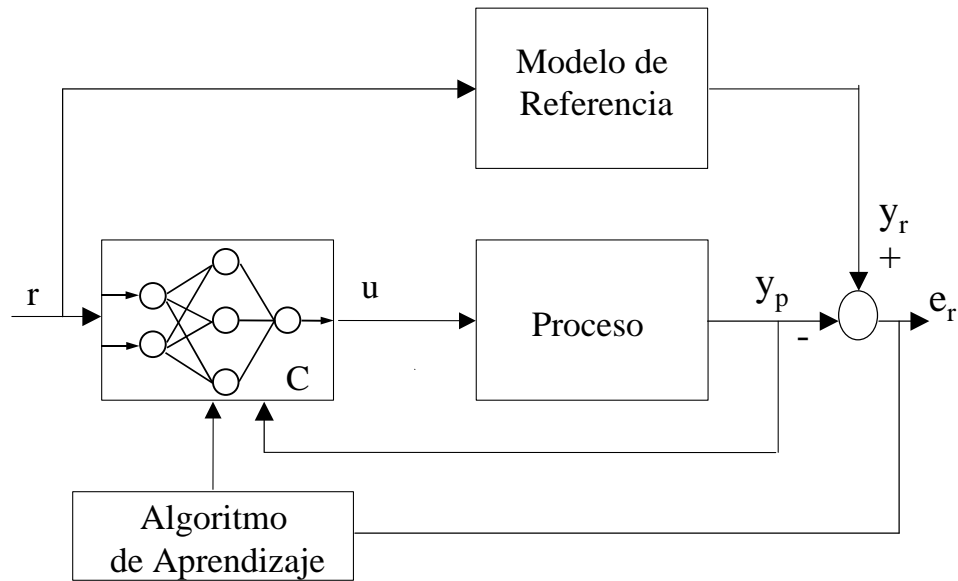


Figura 4.15: Esquema de control neuronal por modelo de referencia

El funcionamiento deseado del sistema en lazo cerrado es especificado a través de un modelo de referencia estable, que se define por el par entrada-salida  $\{r(t), y_r(t)\}$ .

El sistema de control pretende llevar a la salida de la planta  $y_p(t)$  a la salida del modelo de referencia  $y_r(t)$  asintóticamente, es decir:

$$\lim_{t \rightarrow \infty} \|y_r(t) - y_p(t)\| \leq \varepsilon \quad \varepsilon \geq 0$$

En esta estructura, el error entre  $y_r$  e  $y_p$  es usado para entrenar al controlador neuronal.

#### 4.11. Control Predictivo basado en redes neuronales

En general, los controladores predictivos basado en redes neuronales presentan el siguiente esquema básico:

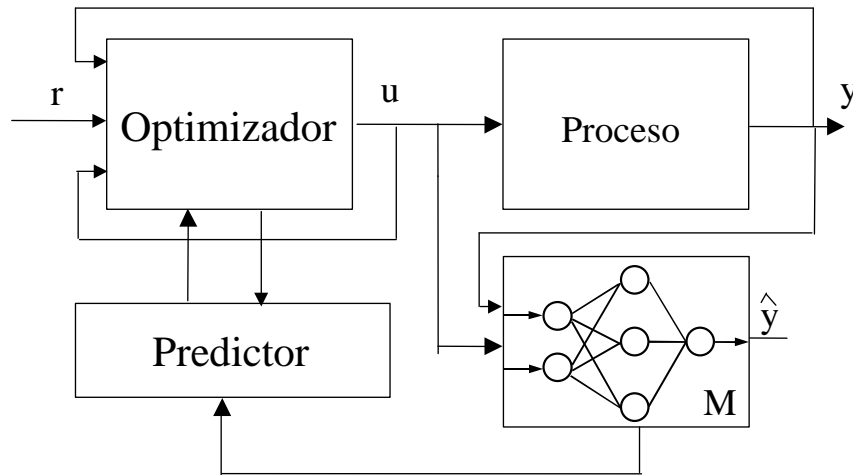
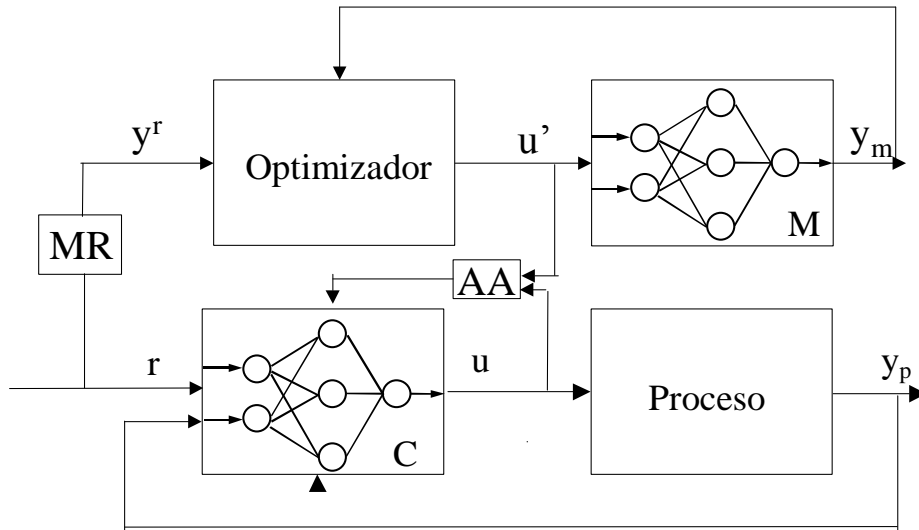


Figura 4.16: Esquema general del control predictivo según redes Neuronales.

En este esquema, la obtención de la ley de control con redes neuronales puede considerar alguna de las siguientes alternativas:

- 1.- Determinar el modelo del sistema con una red neuronal. El modelo es usado para predecir las salidas futuras de la planta. Esto permite tratar con procesos no lineales.
- 2.- Entrenar una red neuronal para que realice la misma tarea que un controlador predictivo. El entrenamiento se realiza por simulación fuera de línea. De este modo, se obtiene un controlador más rápido.
- 3.- Entrenar una red neuronal de forma que optimice un criterio. El modelo es evaluado tan solo en la fase de entrenamiento del controlador.

A continuación, se presentan algunas estrategias de control predictivo basado en redes neuronales:

**Hunt (1992)****Figura 4.17: DMC según Hunt.**

En este caso, una red neuronal ( $M$ ) predice las respuestas futuras de la planta ( $y_m$ ) sobre un horizonte de tiempo.

Las predicciones alimentan al optimizador de manera de optimizar el siguiente criterio:

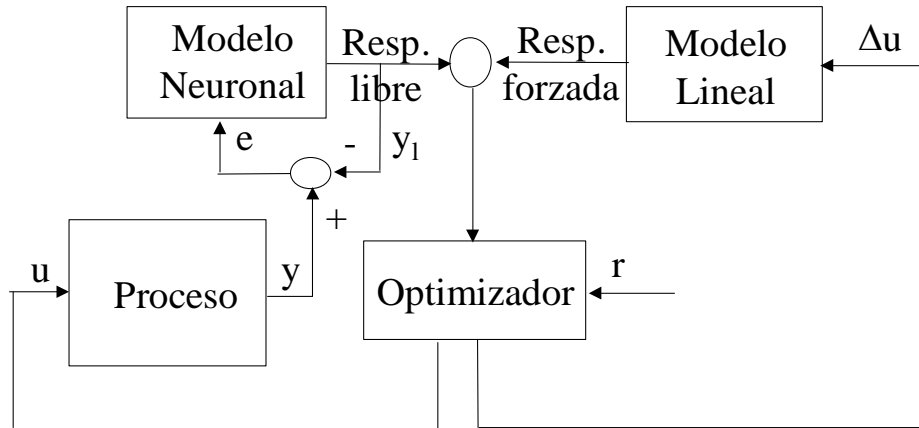
$$J = \sum_{j=N_1}^{N_2} \delta(j) \left[ y^r(t+j) - y_m(t+j/t) \right]^2 + \sum_{i=1}^{N_u} \lambda(i) \left[ \Delta u'(t+i-1) \right]^2$$

donde  $y^r$  puede ser la respuesta de un modelo de referencia (trayectoria de referencia).

Una alternativa es entrenar una red neuronal  $C$  de manera de imitar la acción de control  $u'$ .

**Arahal (1997)**

GPC basado en respuesta libre (red neuronal) y respuesta forzada.



**Figura 4.18: DMC según Arahal.**

Se propone dividir la respuesta del sistema en libre y forzada. La respuesta forzada es debido a la señal de control, el resto se considera respuesta libre.

$$\hat{y}(t + j | t) = \hat{y}_{libre}(t + j | t) + \hat{y}_{forzado}(t + j | t)$$

Para predecir la respuesta forzada se usará un modelo lineal.

La predicción de la respuesta libre se hará en base a un modelo neuronal, válido para todo rango de operación.

El proceso de optimización puede resolverse analíticamente puesto que el modelo dependiente de la señal de mando es lineal.

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (w - \hat{y}_{libre})$$

A fin de obtener mejores resultados, el modelo usado para calcular la respuesta forzada se cambia con el punto de operación.



**Draeger (1995)**

DMC basado en redes neuronales

Se considera el siguiente modelo lineal respuesta al escalón:

$$\hat{y} = G\Delta u + f + d$$

con  $G$  es la matriz con coeficientes de la respuesta al escalón,  $f$  es agrupa los términos conocidos hasta  $t$  (respuesta libre) y  $d$  es un vector de perturbaciones.

El vector de perturbaciones será tal que una parte representará la no linealidad del proceso y la otra parte considera las influencias desconocidas.

$$d = d_{nl} + d^*$$

con  $d^* = y - y_m$

Entonces, la acción de control está dada por:

$$\Delta u = (G^T G + \lambda I)^{-1} G^T (r - f - d)$$

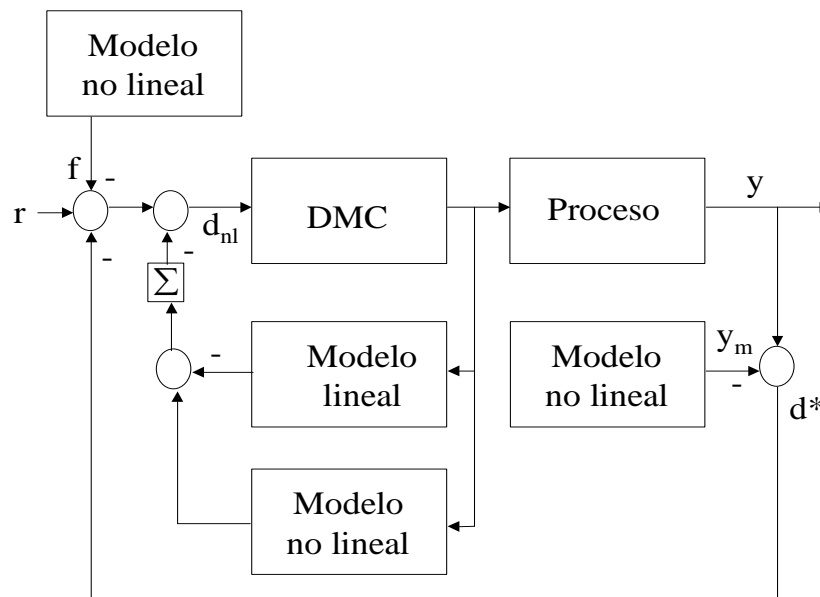


Figura 4.19: DMC según Draemer.

#### 4.12. Principios básicos de sistemas neuro-difusos

La principal razón de representar un sistema difuso en términos de una red neuronal es utilizar la capacidad de aprendizaje de las redes para mejorar su funcionamiento.

##### 4.12.1. Sistema de inferencia difusa neuronal adaptativa (ANFIS)

Estas estructuras son muy utilizadas en control y muchas otras aplicaciones. Para entender el concepto, se considera el problema grafico representado por un control difuso tipo Takagi-Sugeno basado en dos reglas.

$$R_1 : \text{Si } x_1 \text{ es } A_1 \text{ y } x_2 \text{ es } B_1 \text{ entonces } y = f_1(x)$$

$$R_2 : \text{Si } x_1 \text{ es } A_2 \text{ y } x_2 \text{ es } B_2 \text{ entonces } y = f_2(x)$$

donde  $A_i$  y  $B_i$  son conjuntos difusos y

$$f_1(x) = z_{11}x_1 + z_{12}x_2 + z_{13}$$

$$f_2(x) = z_{21}x_1 + z_{22}x_2 + z_{23}$$

Dado  $x = (x_1, x_2)$  un valor numérico de la entrada, el mecanismo de inferencia difusa produce:

$$y^* = \frac{A_1(x_1)B_1(x_2)f_1(x) + A_2(x_1)B_2(x_2)f_2(x)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)}$$

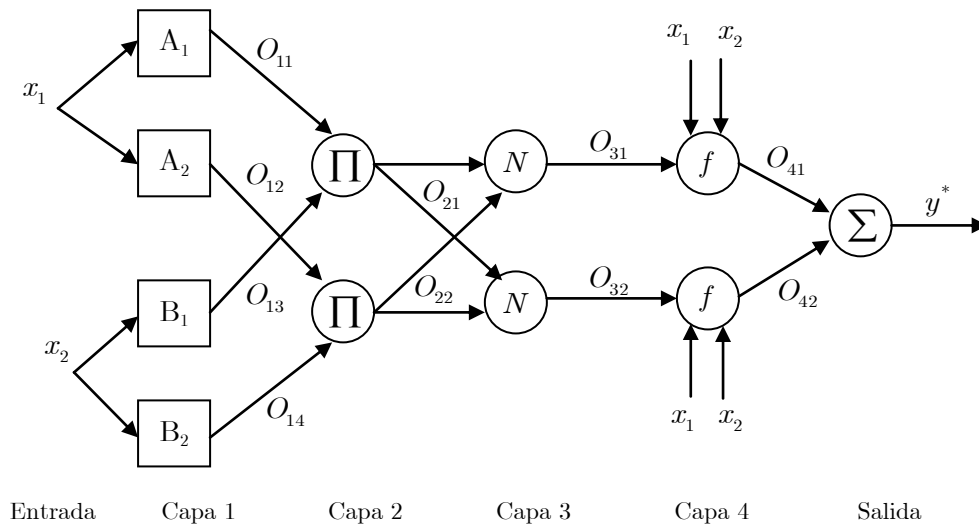


Figura 4.20: Modelo difuso Takagi Sugeno de primer orden con dos reglas.

La salida de la capa 1 es:

$$(O_{11}, O_{12}, O_{13}, O_{14}) = (A_1(x_1), A_2(x_1), B_1(x_2), B_2(x_2))$$

La capa 2 consiste en neuronas difusa, entonces sus salidas son:

$$(O_{21}, O_{22}) = (A_1(x_1)B_1(x_2), A_2(x_1)B_2(x_2))$$

La capa 3 es un normalizador. Su salida es:

$$\begin{aligned} (O_{31}, O_{32}) &= \left( \frac{O_{21}}{O_{21} + O_{22}}, \frac{O_{22}}{O_{21} + O_{22}} \right) \\ &= \left( \frac{A_1(x_1)B_1(x_2)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)}, \frac{A_2(x_1)B_2(x_2)}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \right) \end{aligned}$$

La salida de la capa 4 es:

$$\begin{aligned} (O_{41}, O_{42}) &= (O_{31}f_1, O_{32}f_2) \\ &= \left( \frac{A_1(x_1)B_1(x_2)(z_{11}x_1 + z_{12}x_2 + z_{13})}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)}, \frac{A_2(x_1)B_2(x_2)(z_{21}x_1 + z_{22}x_2 + z_{23})}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \right) \end{aligned}$$

La salida resultante es:

$$\begin{aligned} y^* &= O_{41} + O_{42} \\ &= \left( \frac{A_1(x_1)B_1(x_2)(z_{11}x_1 + z_{12}x_2 + z_{13}) + A_2(x_1)B_2(x_1)(z_{21}x_1 + z_{22}x_2 + z_{23})}{A_1(x_1)B_1(x_2) + A_2(x_1)B_2(x_2)} \right) \end{aligned}$$

#### 4.12.2. Algoritmo de Aprendizaje ANFIS

En la estructura ANFIS, los parámetros de las premisas y consecuencias son los pesos. Específicamente, las funciones de pertenencia  $A_i^j$  son parámetros determinados por el número finito de parámetros  $a_i, b_i, c_i, i = 1, 2$ , son los parámetros de las consecuencias.

El algoritmo de aprendizaje ANFIS consiste en ajustar el conjunto de parámetros a partir de los datos muestreados  $((x_1^k, x_2^k), y^k), k = 1, \dots, N$ .

Se describe un método de aprendizaje de las funciones de pertenencia para las premisas y parámetros de las consecuencias.

El conjunto de entrenamiento está dado por:

$$((x^1, y^1), \dots, (x^k, y^k))$$

Donde  $x^k = (x_1^k, \dots, x_n^k) \in \mathbb{R}^n$  e  $y^k \in \mathbb{R}$

Las reglas difusas definidas son:

$$R_i : \text{Si } x_1^k \text{ es } A_i^1 \text{ y } \dots \text{ y } x_n^k \text{ es } A_i^n \text{ entonces } y = \sum_{j=1}^n z_i^j x_j^k + z_i$$

Donde  $A_i^j$  son funciones de pertenencia difusas y  $z_i^j$  son números reales.

Sea  $O^k$  la salida desde el sistema difuso correspondiente a la entrada  $x^k$ .

Se considera que el operador AND está dado por el producto, entonces el antecedente de la regla  $i$  es:

$$\alpha_i^k = \prod_{j=1}^n A_i^j(x_j^k)$$

Entonces, la salida del sistema es:

$$O^k = \frac{\sum_{i=1}^m \alpha_i^k (\sum_{j=1}^n z_i^j x_j^k + z_i^0)}{\sum_{i=1}^m \alpha_i^k} = \frac{\sum_{i=1}^m \left( \prod_{j=1}^n A_i^j(x_j^k) \right) (\sum_{j=1}^n z_i^j x_j^k + z_i^0)}{\sum_{i=1}^m \prod_{j=1}^n A_i^j(x_j^k)}$$

Se define la medida del error para el patrón de entrenamiento  $k^{th}$  como:

$$E^k = \frac{1}{2} (O^k - y^k)^2$$

donde  $O^k$  es la salida del sistema difuso correspondiente al patrón de entrada  $x^k$  e  $y^k$  es la salida deseada,  $k = 1, \dots, K$ .

### 4.13. Problemas Resueltos

#### Problema 4.1

Para la siguiente red neuronal multicapa de la figura 4.20, encontrar las salidas  $y_1$  e  $y_2$ . Las entradas externas están dadas por  $\varphi_1, \varphi_2, \varphi_3$

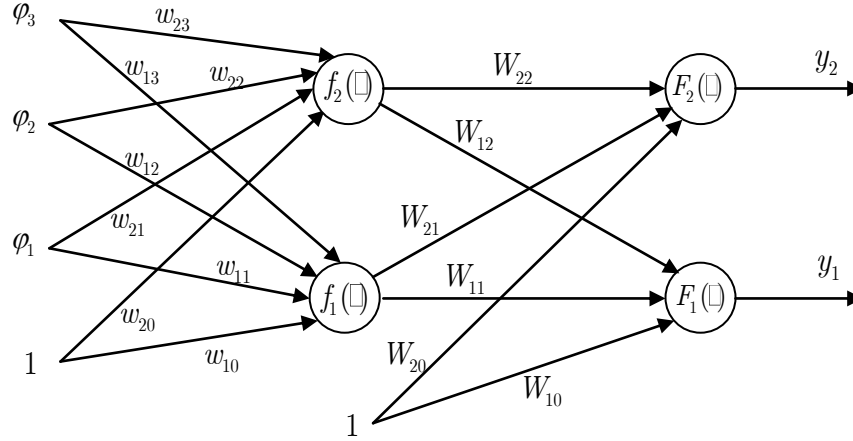


Figura 4.21: Esquema del problema 4.1

#### Solución

Para encontrar la solución a este tipo de problemas, se debe utilizar la metodología de partir de adelante hacia atrás, es decir, para  $y_1$  se tendrá que:

$$y_1 = F_1(\square) = F_1(W_{11}f_1(\square) + W_{12}f_2(\square) + W_{10}) = F_1\left(\sum_{j=1}^2 W_{1,j}f_j(\square) + W_{10}\right)$$

Para calcular  $f_1(\square)$  y  $f_2(\square)$ , se utiliza el mismo procedimiento:

$$f_1(\square) = f_1(w_{11}\varphi_1 + w_{12}\varphi_2 + w_{13}\varphi_3 + w_{10}) = f_1\left(\sum_{m=1}^3 w_{1,m}\varphi_m + w_{10}\right)$$

Análogo para  $f_2(\square)$

$$f_2(\square) = f_2\left(\sum_{m=1}^3 w_{2,m}\varphi_m + w_{20}\right)$$

Por lo tanto la salida  $y_1$  está dada por:

$$y_1 = F_1\left(\sum_{j=1}^2 W_{1,j}f_j\left(\sum_{m=1}^3 w_{j,m}\varphi_m + w_{j,0}\right) + W_{10}\right)$$

Para encontrar la salida  $y_2$ , se realiza el mismo procedimiento.

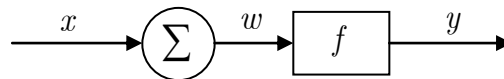
### Problema 4.2

Se tiene una red neuronal de una capa, una entrada - una salida, cuya función de transferencia no lineal es una sigmoide logarítmica.

- Dibuje un esquema que represente la red antes mencionada, indicando claramente: entradas, pesos, expresiones para la salida y para la función NET. Considere que no hay sesgos asociados a la red.
- Dados los siguientes valores para la entrada  $X = \{-3; 2\}$  y para la salida  $Y = \{0.3; 0.6\}$ , determine mediante el algoritmo de entrenamiento, backpropagation, el peso asociado a dicha entrada. Considere la tasa de aprendizaje es 0.9 y que inicialmente  $w$  es 0.1.
- Compruebe la efectividad de su entrenamiento evaluando la salida de la red, para las entradas del punto b), utilizando el peso encontrado. Comente.

### Solución

a)



donde

- $x$  es la entrada
- $y$  es la salida
- $w$  es el peso
- $f$  es una función sigmoide

$$y = f(wx) = \frac{1}{1 + e^{-wx}}$$

b) Datos

$p$	$x$	$y$
1	-3.0	0.3
2	2.0	0.6

$$\eta = 0.9$$

$$w(0) = 0.1$$

1º iteración (con primer conjunto de patrones)

$$o = f(w(0) \cdot x) = f(0.1 \cdot -3.0) = \frac{1}{1 + e^3} = 0.4255$$

$$\begin{aligned}\delta &= (1 - o)o(y - o) \\ &= (1 - 0.4255) \cdot 0.4255 \cdot (0.3 - 0.4255) \\ &= -3.069 \cdot 10^{-2}\end{aligned}$$

$$\Delta w = \eta \delta x = 0.9 \cdot -3.069 \cdot 10^{-2} \cdot (-3.0) = 8.2873 \cdot 10^{-2}$$

$$\begin{aligned}w(1) &= w(0) + \Delta w = 0.1 + 8.2873 \times 10^{-2} \\ &= 0.18287\end{aligned}$$

2º iteración (con segundo conjunto de patrones)

$$o = f(w(1) \cdot x) = f(0.18287 \cdot 2.0) = \frac{1}{1 + e^{-0.36574}} = 0.59043$$

$$\begin{aligned}\delta &= (1 - o)o(y - o) \\ &= (1 - 0.59043) \cdot 0.59043 \cdot (0.6 - 0.59043) \\ &= 2.314 \cdot 10^{-2}\end{aligned}$$

$$\Delta w = \eta \delta x = 0.9 \cdot 2.3144 \cdot 10^{-2} \cdot 2.0 = 4.166 \cdot 10^{-3}$$

$$\begin{aligned}w(2) &= w(1) + \Delta w = 0.18287 + 4.166 \times 10^{-3} = 0.187036 \\ &= 0.187036\end{aligned}$$

c)

$p$	$x$	$y$
1	-3.0	0.3
2	2.0	0.6

$$o^p = f(wx^p)$$

$$o^1 = f(wx^1) = f(0.187036 \times -3.0) = 0.3633$$

$$o^2 = f(wx^2) = f(0.187036 \times 2.0) = 0.5924$$

Entonces, los errores de estimación son:

$$e^1 = y^1 - o^1 = 0.3 - 0.3633 = -0.0633$$

$$e^2 = y^2 - o^2 = 0.6 - 0.5924 = 0.0076$$

Entonces, el error de ajuste es:

$$e = \frac{1}{2} \sum_{i=1}^2 (y^i - o^i)^2 = 0.002$$

### Comentario

Se podría mejorar este peso resultante ( $w^*$ ), si se realizan más iteraciones con los mismos set de datos. También se podría realizar de nuevo las iteraciones empezando con el  $x = 2$  e  $y = 0.6$ , lo cual podría mejora el resultado.



## Capítulo 5

### 5. Control Adaptativo de sistemas

En el contexto de automática el término adaptativo se refiere a la facultad de cambiar el comportamiento o parámetros del control en respuesta a cambios en las circunstancias del sistema controlado.

La configuración típica de un controlador adaptativo es la que se muestra en la figura 5.1. Como se puede observar hay un lazo principal de realimentación negativa en el que aparece un regulador ajustable y otro lazo que se utiliza para ajustar los parámetros de dicho regulador. Para ello, se obtiene un cierto índice de actuación en el cual se expresa el comportamiento del controlador. Dicho índice de actuación se compara con un cierto comportamiento deseado y según el resultado de la comparación se ajustan los parámetros del regulador. Para ello se utiliza un mecanismo de adaptación que en algunos casos también puede actuar directamente sobre la señal de control.

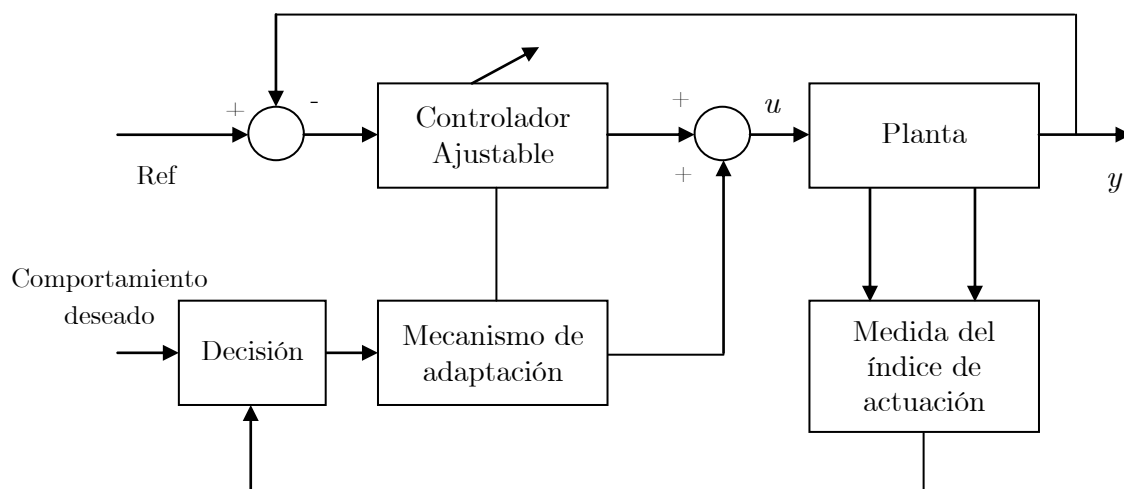


Figura 5.1: Configuración general de un controlador adaptativo.

### 5.1. Control Adaptativo por sistema de referencia (MRAC)

Es una de las técnicas más antiguas de control adaptativo y se basa, como su nombre indica, en disponer de un modelo de bucle cerrado que es el que se desea que describa al conjunto controlador-planta. Es decir, se debe partir de un conjunto de especificaciones deseadas de bucle cerrado que se expresan mediante el modelo de referencia (ver figura 5.2).

Para diseñar un MRAC se ha de definir el modelo de referencia, el controlador y la ley de adaptación. En cuanto al modelo de referencia, sabemos que este especifica el comportamiento en bucle cerrado deseado. Por tanto, el modelo ha de ser tal que el conjunto controlador ajustable-planta pueda reproducir dicho modelo.

Finalmente, para la ley de adaptación existen diferentes estrategias en la literatura tales como el método de hiperestabilidad y la estrategia basada en la teoría de estabilidad de Lyapunov y la más popular, la regla MIT.

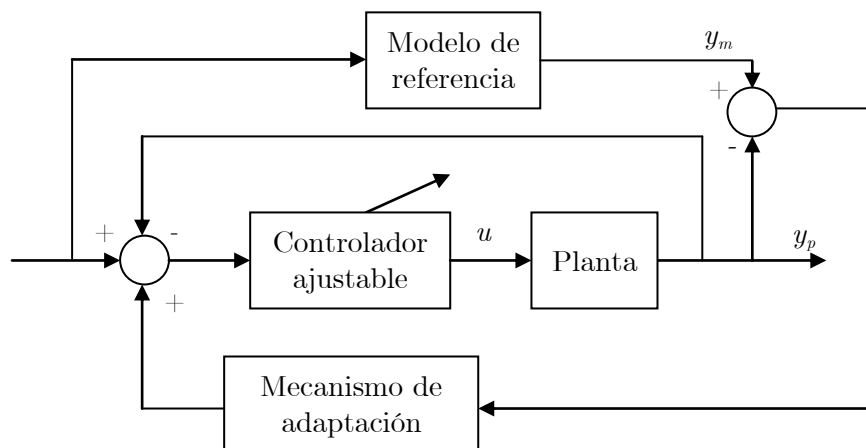


Figura 5.2: Configuración general de un MRAC

### 5.2. Regla MIT

La regla MIT es el enfoque original para el control adaptable basado en el modelo de referencia. El nombre se deriva del hecho de que fue desarrollado en el Laboratorio de instrumentación en el Massachusetts Institute of Technology.

Considérese un sistema de lazo cerrado en el cual el controlador tiene un parámetro ajustable  $\theta$ . La respuesta deseada en lazo cerrado se especifica mediante un modelo que tiene la salida  $y_m$ . Sea  $e = y(t) - y_m(t)$ , el error entre la salida y del sistema de lazo cerrado y la salida  $y_m$  del modelo. Una posibilidad es ajustar los parámetros de una manera tal que se minimice una función de costo:

$$J(\theta) = \frac{1}{2} e(t)^2$$

Para minimizar  $J$  es razonable cambiar los parámetros en la dirección del gradiente negativo de  $J$ , como se muestra en la ecuación, la cual es conocida como la regla MIT.

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}$$

La ecuación anterior aplica también cuando hay varios parámetros para ajustar, de aquí que el símbolo  $\theta$  debe ser interpretado como un vector y  $\frac{\partial e}{\partial \theta}$  como el gradiente del error con respecto a los parámetros.

Alternativamente se puede considerar la regla

$$J(t) = |e(t)|, \text{ que resulta en } \frac{\partial \theta}{\partial t} = -\gamma \frac{\partial e(t)}{\partial \theta} \text{signo}(e(t))$$

Otra alternativa, es el algoritmo sign-sign usado en telecomunicaciones, ya que es fácil de implementar y computacionalmente rápido:

$$\frac{\partial \theta}{\partial t} = -\gamma \cdot \text{signo}\left(\frac{\partial e(t)}{\partial \theta}\right) \cdot \text{signo}(e(t))$$

## Capítulo 6

### 6. Referencias

EN CONSTRUCCION!!!!

# Capítulo 7

## 7. Anexos

En esta sección se agregaran contenidos ya visto en otros ramos de control.

### 7.1. Linealización

Considerar el sistema no lineal representado por la ecuación:

$$\dot{x}(t) = f(x(t), u(t)), \quad x \in R^n, u \in R^m$$

La linealización del sistema puede realizarse mediante una expansión en serie de Taylor en torno a un punto de operación. Sean los puntos de operación  $(x_0, u_0)$ , realizando la expansión:

$$\dot{x}(t) = f(x_0(t), u_0(t)) + \left. \frac{\partial f}{\partial x} \right|_{x_0, u_0} (x - x_0) + \left. \frac{\partial f}{\partial u} \right|_{x_0, u_0} (u - u_0)$$

Para  $i = 1, 2, \dots, n$

Donde

$$\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}, \quad \frac{\partial f}{\partial u} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_n} \\ \frac{\partial f_2}{\partial u_1} & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \dots & \frac{\partial f_n}{\partial u_n} \end{pmatrix}$$

## 7.2. Controlabilidad y observabilidad

Una planta es **controlable** si en cada instante dado, es posible controlar cada estado de modo que un resultado deseado se puede alcanzar.

Sea un sistema lineal, con parámetros invariantes en el tiempo:

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

Es completamente controlable ssi la matriz de controlabilidad  $n \times nm$ :

$$C = [B \quad AB \quad \dots \quad A^{n-1}B]$$

Tiene rango completo ( $n$ ), donde  $A$  es  $n \times n$  de y  $B$  es de  $n \times m$ .

Una planta es **observable** si los estados pueden ser determinados de las mediciones de la salida.

Sea un sistema lineal, con parámetros invariantes en el tiempo:

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

Es completamente observable ssi la matriz de observabilidad  $nr \times n$ :

$$\mathbb{O} = [C \quad CA \quad \dots \quad CA^{n-1}]^T$$

Tiene rango completo ( $n$ ), donde  $A$  es  $n \times n$  de y  $C$  es de  $n \times r$ .