

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



CONTROL DIFUSO DEL SISTEMA DEL PÉNDULO INVERTIDO

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

CARLOS ANTONIO KOBASHIKAWA CARRASCO

PROMOCIÓN

1996-2

LIMA - PERÚ

2003

A mi familia.

**CONTROL DIFUSO DEL SISTEMA DEL PÉNDULO
INVERTIDO.**

SUMARIO

En el presente trabajo se da una base teórica concisa sobre lógica difusa ("fuzzy logic") y posteriormente se hace el análisis y diseño de un controlador fuzzy para el sistema del péndulo invertido. Este trabajo muestra los pasos en el diseño del controlador así como los aspectos que hay que considerar dadas las características de la planta. También se incluyen simulaciones numéricas del controlador diseñado y de otro basado en la teoría clásica del espacio de estados, para luego hacer una comparación entre ellos y mostrar las ventajas en cuanto a flexibilidad en el diseño que tiene el primero.

ÍNDICE

| | |
|--------------------------------------------------------------------|----|
| PRÓLOGO | 01 |
| CAPÍTULO I | |
| BASE TEÓRICA | 03 |
| 1.1 Conceptos sobre lógica difusa (“fuzzy logic”) | 03 |
| 1.2 Operaciones fuzzy | 09 |
| 1.3 Teoría fuzzy versus teoría probabilística | 13 |
| 1.4 Control fuzzy | 15 |
| CAPÍTULO II | |
| MODELO MATEMÁTICO DEL SISTEMA DEL PÉNDULO INVERTIDO | 22 |
| CAPÍTULO III | |
| DISEÑO DEL CONTROLADOR | 26 |
| 3.1 Diseño del controlador usando la teoría del espacio de estados | 26 |
| 3.2 Diseño del controlador usando Fuzzy Logic | 27 |
| CAPÍTULO IV | |
| SIMULACIONES | 33 |
| CONCLUSIONES | 53 |
| ANEXOS | |

| | |
|----------------------------------------------------|-----------|
| ANEXO 1: LISTADO DEL PROGRAMA “PROGRAMA1.M” | 55 |
| ANEXO 2: LISTADO DEL PROGRAMA “GETK.M” | 59 |
| BIBLIOGRAFÍA | 63 |

PRÓLOGO

La primera idea acerca de fuzzy logic (lógica difusa) fue presentada por Lofti A. Zadeh en 1965 y las ideas fundamentales sobre controles fuzzy logic fueron sugeridas en diversas notas publicadas entre 1968 y 1973. La primera implementación fue hecha por Mamdani y Assilian en 1974 en Inglaterra: la regulación de un generador a vapor. A partir de entonces, y una vez que las ideas fundamentales del control fuzzy logic fueron comprendidas mejor, las aplicaciones fueron multiplicándose, principalmente en Japón.

Si bien la lógica difusa ha sido usada exitosamente en muchas aplicaciones, aún se hacen cuestionamientos a sus fundamentos. Esto puede deberse a dos razones:

- El diseño de un control fuzzy es difícil, debido a la ausencia de una base teórica.
- El desempeño de un control fuzzy puede ser inconsistente debido a su dependencia de la experiencia práctica de un operador humano.

También es de notar que en casi todas sus aplicaciones exitosas, los controladores no son pensantes sino del tipo llamado incorporados, mientras que la mayoría de los artículos desarrollados sobre lógica difusa tratan con la representación del conocimiento y del razonamiento.

El presente trabajo mostrará la aplicación de la teoría de la lógica difusa en el control de un sistema bastante conocido como es el del péndulo invertido.

El problema planteado básicamente consiste en mantener en equilibrio una varilla (péndulo) colocada verticalmente hacia arriba, cuyo extremo base está unido a un eje. Este eje está en un carrito móvil, el cual puede desplazarse sobre un riel horizontal mediante un sistema de poleas accionado por un servomotor DC. La varilla puede girar libremente según el eje de su base, en el plano vertical que pasa por el riel y el carrito.

En una primera parte daré una breve base teórica sobre lógica difusa que ayudará a comprender mejor la filosofía en el diseño del sistema de control. Posteriormente, haré el modelamiento matemático del sistema propuesto, haciendo el análisis del subsistema mecánico separado del subsistema eléctrico y luego uniéndolos.

Una vez modelado el sistema, aplicaré los conceptos de lógica difusa para diseñar un controlador, así como otro basado en la teoría convencional de espacio de estados.

Finalmente, presentaré el resultado de las simulaciones realizadas con MATLAB, para ambos tipos de controladores diseñados, haciendo una comparación entre ellos y presentando las conclusiones respectivas del análisis.

CAPÍTULO I BASE TEÓRICA (LÓGICA DIFUSA Y CONTROL DIFUSO)

1.1 Conceptos sobre lógica difusa (“fuzzy logic”)

En [1] se hace una clasificación de los términos lingüísticos y los valores numéricos, colocándolos en tres categorías de acuerdo a su significado, los cuales son definidos por funciones características:

1) Palabras determinísticas, por ejemplo "hombre", "muerto" o "Juan".

Todas ellas tienen valores lógicos 0 ó 1 que corresponden a 'sí' o 'no', es decir, si a uno le preguntan ¿es Ud. hombre? o ¿está Ud. vivo? o ¿su nombre es Juan?, la respuesta es única y exacta y solo puede ser 'sí' o 'no'. Es similar para el caso de los valores numéricos, cuando se especifica "exactamente 37°C" o "47,25Km". Vemos pues que esta categoría no tiene ni flexibilidad ni intervalos y en un gráfico significaría solo un punto en el dominio; por eso a este tipo de término o valor se le llama “singleton” (Fig. I-1).

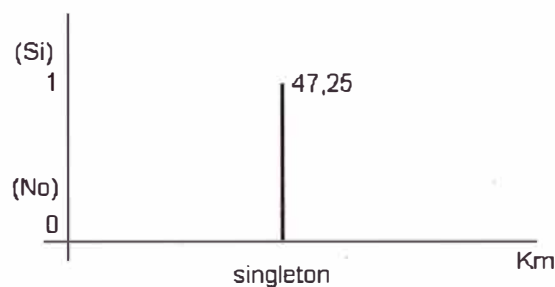


Fig. I-1. Un singleton.

2) Conjunto "crisp" (preciso): Aún en la ciencia donde se prefiere el uso de valores exactos, a veces se usan intervalos numéricos para representar valores con cierta flexibilidad. Por ejemplo: "una temperatura confortable en una habitación es entre 20°C y 28°C". Esto podría significar que una temperatura de 19,999°C y otra de 28.0001°C no son confortables. Al igual que en la primera categoría, los valores lógicos que puede tomar la temperatura dado un intervalo es sí o no ("¿la temperatura está en el rango 'confortable'?"), por lo que podemos ver este intervalo como un conjunto numeroso de singletons. A este intervalo determinístico se le denomina conjunto crisp (preciso), y se muestra en la Fig. I-2.

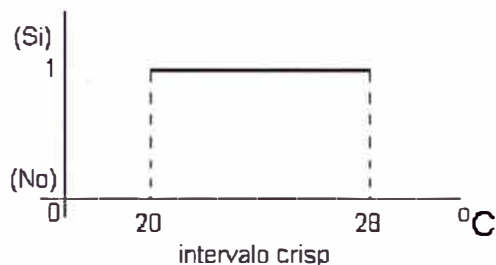


Fig. I-2. Un intervalo crisp.

3) Valores "fuzzy": son los que generalmente se usan en la vida real para una comunicación fácil y eficiente. Por ejemplo, el concepto "alto" o "frío", en donde los valores lógicos no son solamente 0 ó 1, "sí" o "no", sino que pueden tomar un conjunto de valores como "un poco" alto o "más o menos" frío, que se encuentran entre los valores lógicos 0 y 1. Como se verá más adelante, las funciones que describen detalladamente estos valores se

denominan funciones de membresía (“membership function”). Un conjunto que representa un valor fuzzy se muestra en la Fig. I-3.



Fig. I-3. Un conjunto fuzzy.

Un ejemplo aclarará este punto: ¿Que calificación (0-20) usted cree que es buena?. Una forma de responder esta pregunta es establecer que todas las calificaciones superiores a 14 son buenas y las inferiores no lo son; esto es, se coloca una marca o umbral de concordancia/discordancia con respecto a lo que se considera una calificación buena. Una calificación de 13.99 no sería buena y una de 14.01 sí.

Otra forma de hacerlo es colocar una marca en 19 y decir que una calificación es buena mientras más cercana esté a ella.

Una tercera forma de responder sería colocando marcas (como en el anterior caso) tanto en los lados 'buena' y 'mala' calificación, sin colocar una en donde uno esté indeciso. Así, las calificaciones estarán agrupadas en 3 subgrupos: 'buena', 'mala' y 'ni buena ni mala' calificación.

Como se ve, tenemos diferentes maneras de representar el conocimiento de "buena calificación" y esto implica que la propiedad "buena calificación" es inherentemente vaga, significado que quiere decir que el

conjunto de calificaciones a las que se le aplica no tiene límites abruptos. Por consiguiente, lo difuso (o vago) de la propiedad radica en no tener unos límites claramente definidos.

En este ejemplo, el intervalo $U=\{x/x\in [0,20]\}$ es el campo de referencia o "universe of discourse" (universo en discusión), el cual cubre todo el rango de calificaciones y B es el subconjunto de buenas calificaciones en U, donde la transición entre 'buena' y 'mala' calificación no es abrupta sino gradual. Generalmente, dada una calificación, podremos definirla como buena o mala; sin embargo, las que estén en el límite no serán tan fáciles de definir. Luego, podremos asignar ciertos grados de pertenencia o membresía al subconjunto: aquellos con grado 1 pertenecerán totalmente y los de grado 0 estarán excluidos. En los límites, no obstante, los grados de membresía estarán entre 0 y 1, siendo los más cercanos a la unidad los que califican mejor como miembros del subconjunto.

Tres cosas deben de quedar claras aquí:

1. No existen grados de membresía precisos. Estos son asignados según un índice de tendencia, en forma subjetiva, por un individuo o grupo de individuos.

2. Los grados de membresía no están absolutamente definidos. Así, una calificación de 16 sería considerada buena por un estudiante promedio pero no lo sería tanto por uno exigente.

3. Lo difuso difiere de lo impreciso en que este último se refiere a una carencia de conocimiento acerca del valor de un parámetro, por lo que es expresado como un intervalo de tolerancia en donde pueden caer todos los

posibles valores de tal parámetro. En cambio, lo difuso ocurre cuando el intervalo no tiene límites definidos.

En el ejemplo anterior identifiquemos los conceptos fuzzy relacionados con la variable lingüística "calificación": el nombre de la variable es "calificación", los valores lingüísticos que puede tomar podrían ser "buena", "regular" y "mala" y el universo de la variable ("universe of discourse") es el intervalo numérico $[0,20]$. Cada valor lingüístico tiene asociado mediante una función semántica una función de membresía, es decir una función matemática que asigna a cada calificación dada, un valor entre 0 y 1, y que indica cuán perteneciente al conjunto es: mientras más próximo a la unidad sea el valor asignado por la función de membresía, más "perteneciente" es al conjunto y mientras más cercano sea dicho valor a 0, es más "no perteneciente".

Así, la función de membresía (μ_F) es una generalización de la función característica de los conjuntos críps, en donde a cada elemento 'u' perteneciente al universe of discourse U se le asigna un valor dentro del intervalo unidad $[0;1]$ en lugar de los dos elementos $\{0,1\}$, el cual indica el grado de pertenencia de 'u' al conjunto difuso F en U, significando 1 una total pertenencia y 0 total exclusión. Esto es:

$$\mu_F : U \rightarrow [0,1]$$

El conjunto F queda totalmente definido por el conjunto de duplas:

$$F = \{(u, \mu_F(u)) | u \in U\}$$

existiendo varias otras formas de definirlo; por ejemplo, describese el concepto "país extenso" sobre el universe of discourse $P = \{\text{Uruguay, Perú, Rusia, Brasil, Suiza, Líbano, India}\}$:

"país extenso" =

= $\{(\text{Uruguay}, 0.3), (\text{Perú}, 0.7), (\text{Rusia}, 1), (\text{Brasil}, 1), (\text{Suiza}, 0.2), (\text{Líbano}, 0.1), (\text{India}, 1)\}$ =

= $0.3/\text{Uruguay} + 0.7/\text{Perú} + 1/\text{Rusia} + 1/\text{Brasil} + 0.2/\text{Suiza} + 0.1/\text{Líbano} + 1/\text{India}$ =

= $\sum_i \mu_F(p_i)/p_i = \sum_{p \in P} \mu_F(p_i)/p_i$

siendo en el caso continuo: $F = \int_P \mu_F(p_i)/p_i \cdot$

Para el caso continuo, las funciones de membresía normalmente tienen la propiedad de ser curvas convexas, es decir:

$$\forall u_1, u_2 \in U \text{ se cumple : } f_A(\lambda u_1 + (1-\lambda)u_2) \geq \min(f_A(u_1), f_A(u_2)) \quad \lambda \in [0, 1]$$

La razón que se menciona en [2] es que, si bien la teoría fuzzy es capaz de tratar con funciones de membresía arbitrarias, la mente humana tiende a formular los conceptos en forma de ideas convexas. No hacerlo, lleva a resultados paradójicos y por eso intuitivamente las personas lo evitan.

Una excepción a esta propiedad se da cuando se define una función de membresía, como el complemento de otra cuya forma se asemeje a ' π '.

En el gráfico de la Fig. I-4, vemos la forma de las funciones de membresía más comunes usadas en ingeniería:

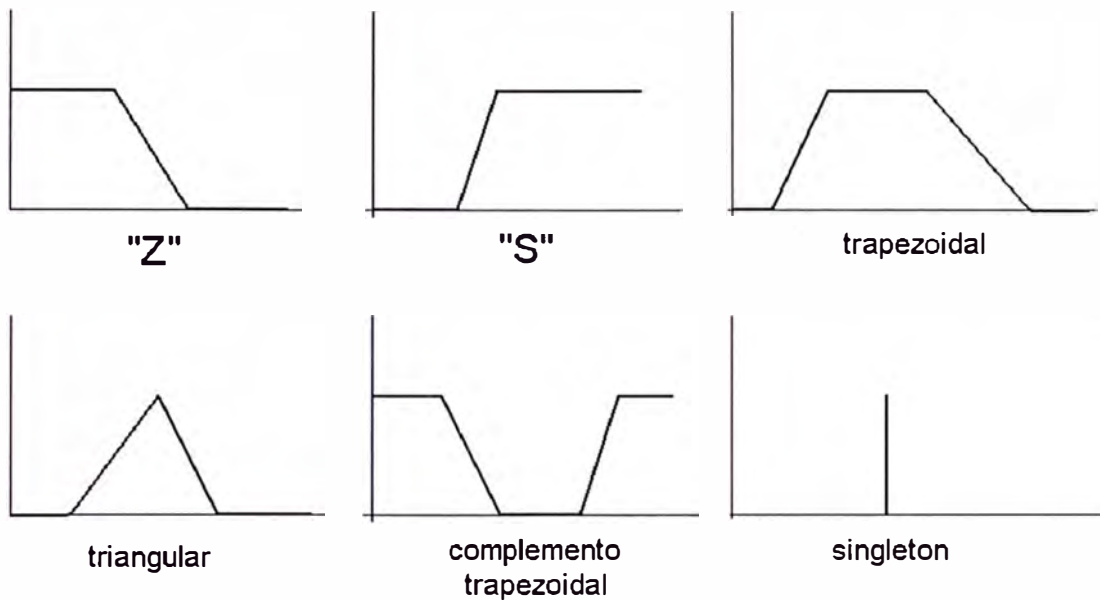


Fig. I-4. Funciones de membresía usuales.

1.2 Operaciones fuzzy

Para dos conjuntos fuzzy, las nociones de igualdad e inclusión pueden ser derivadas directamente de la teoría clásica de conjuntos. Dos conjuntos fuzzy A y B son iguales si todos los elementos del universo tienen el mismo grado de membresía en cada uno de ellos:

$$A = B \Leftrightarrow \forall x \in S : \mu_A(x) = \mu_B(x)$$

Un conjunto fuzzy A es un subconjunto de otro B si cada elemento del universo tiene un grado de membresía menor en A que en B :

$$A \subseteq B \Leftrightarrow \forall x \in S : \mu_A(x) \leq \mu_B(x)$$

Sin embargo, a diferencia de la teoría clásica de conjuntos, las definiciones de las operaciones unión, intersección y complemento de conjuntos (que corresponden a las operaciones lógicas and, or y not) presentan ambigüedades. En la teoría clásica, "a and b" es verdadero

solamente si ambos lo son y "a or b" es falso si ambos lo son, pero en la teoría fuzzy la interpretación no es tan simple, debido a los valores escalonados que pueden tomar las variables. Zadeh propuso las siguientes operaciones:

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

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

$$\forall x \in X : \mu_{A'}(x) = 1 - \mu_A(x)$$

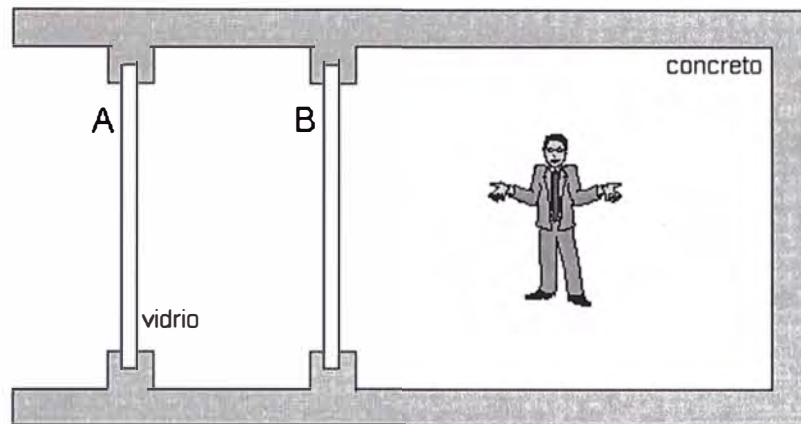
que son una extensión simple de las operaciones clásicas. Otras operaciones son también posibles:

$$\forall x \in X : \mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x)$$

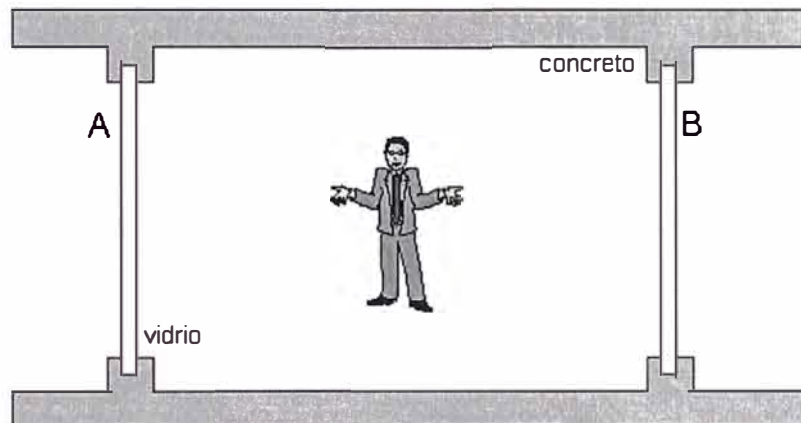
$$\forall x \in X : \mu_{A \cup B}(x) = \min(1, \mu_A(x) + \mu_B(x))$$

¿Cuál usar?. En [3] se ilustra esto con un ejemplo: considérese una persona en una prisión de concreto, donde las paredes A y B son de vidrio resistente (Fig. I-5). Aquí planteamos dos preguntas:

- 1.-¿Cuan fácil es escapar de la prisión?
- 2.-¿Cuan fácil es para la luz solar entrar a la celda?



(a) operación "and"



(b) operación "or"

Fig. I-5. Dos tipos diferentes de celdas:

(a) representa la operación and, (b) la operación or.

Asumamos que para la pared A es (0,1 fácil) de escapar a través de ella y (0,6 fácil) de ser penetrada por la luz solar. Para la pared B estos valores son (0,3 fácil) y (0,9 fácil).

En la primera 1, la facilidad de escape viene dada por la pared menos fácil, es decir A (0,1 fácil), si no consideramos que el reo pueda cansarse, en cuyo caso este valor sería menor. Para el caso de la pregunta 2, la luz penetraría a través de ambas paredes y la facilidad sería el producto

$0,6 * 0,9 = (0,54 \text{ fácil})$. Como se ve, dependiendo de la aplicación, el operador "and" puede ser descrito por el mínimo o por el producto.

En la Fig. I-5(b) se representa el caso para el operador "or". La primera pregunta se responde usando la operación "máximo" y la segunda por una operación como $0,6 + 0,9 - 0,6 * 0,9 = (0,96 \text{ fácil})$ que puede considerarse algo arbitraria.

Para generalizar la definición de estas operaciones se definen familias de operaciones denominadas T-norm y S-norm, las cuales se usan para representar las operaciones de intersección, unión y complemento.

T-norm: $T(a,b)$ denota una clase de función binaria que representa la operación de intersección y que debe de cumplir los siguientes criterios:

$$\text{i) } T(a,b) = T(b,a)$$

$$\text{ii) } T(T(a,b),c) = T(a,T(b,c))$$

$$\text{iii) } a \leq c \text{ and } b \leq d \text{ implica } T(a,b) \leq T(c,d)$$

$$\text{iv) } T(a,1) = a$$

S-norm: $S(a,b)$ denota una clase de función binaria que representa la operación de unión y que debe de satisfacer los tres primeros criterios de las T-norms además de:

$$\text{iv) } S(a,0) = a$$

Complemento: la operación complemento "c" debe al menos de satisfacer:

$$\text{i) } c(0) = 1$$

$$\text{ii) } a < b \text{ implica } c(a) > c(b)$$

$$\text{iii) } c(c(a)) = a$$

Con estas definiciones es fácil percibir que en fuzzy logic no se cumple la propiedad del tercero excluido, es decir:

$$A \cap A' \neq \emptyset$$

$$A \cup A' \neq U$$

donde A es un conjunto fuzzy y A' su complemento.

Un estudio más profundo sobre las matemáticas involucradas en la lógica difusa puede encontrarse en [3] (Capítulo 2).

Uno de los motivos de la teoría fuzzy es evitar cierto tipo de paradojas que surgen en la teoría clásica de conjuntos, por ejemplo:

* Un montículo que contiene una piedra es pequeño

* Si le agregas una piedra a un montículo pequeño, este sigue siendo pequeño

∴ Luego, (por inducción) todos los montículos son pequeños

1.3 Teoría fuzzy versus teoría probabilística

Desde el punto de vista frecuentista, la probabilidad de un evento es la proporción de ocurrencias del evento en una larga serie de experimentos. Sin embargo, existe otro tipo de probabilidad llamada subjetiva, la cual no tiene una interpretación frecuentista sino que es descrita por una cantidad de creencia subjetiva de que un evento ocurra. Aquí, la probabilidad numérica solo refleja el conocimiento incompleto del observador y es este concepto de probabilidad el relevante en la representación del conocimiento y la inferencia bajo incertidumbre.

Zadeh da un interesante estudio de la diferencia entre la teoría de conjuntos difusos y la de probabilidades. Él establece que un conjunto difuso induce una distribución de posibilidad en el universo en discusión; así, el significado del conjunto difuso 'buena calificación' induce la siguiente distribución de posibilidad:

Es imposible que una calificación del 0 al 10 sea buena.

Es 0.1 posible que una calificación de 11 sea buena.

Es 0.2 posible que una calificación de 12 sea buena.

Es 0.3 posible que una calificación de 13 sea buena.

Es 0.5 posible que una calificación de 14 sea buena.

Es 0.7 posible que una calificación de 15 sea buena.

Es 0.9 posible que una calificación de 16 sea buena.

Es totalmente posible que una calificación del 17 al 20 sea buena.

Obsérvese que a diferencia de la teoría de probabilidades no hay necesidad que la suma de posibilidades (o el área bajo la función de distribución de posibilidad) sea la unidad. Consideremos la sentencia "Hans como x huevos en el desayuno", siendo el universe of discourse $U=\{1,2,3,\dots\}$. Asociemos la distribución de posibilidad $\pi_x(u)$ como el grado de facilidad con el que Hans puede comer ' u ' huevos. También podemos asociar una distribución de probabilidad $P_x(u)$ como la probabilidad que Hans coma ' u ' huevos en el desayuno. Asumiendo que empleamos algún criterio explícito o implícito para valorar la facilidad con la que Hans puede

comer 'u' huevos en el desayuno, los valores de $\pi_X(u)$ y $P_X(u)$ podrían ser los de la siguiente tabla:

Tabla I-1. Distribución de posibilidad (π_X) y probabilidad (P_X) asociada a x.

| U | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------------|-----|-----|-----|---|-----|-----|-----|-----|
| $\pi_X(u)$ | 1 | 1 | 1 | 1 | 0,8 | 0,6 | 0,4 | 0,2 |
| $P_X(u)$ | 0,1 | 0,8 | 0,1 | 0 | 0 | 0 | 0 | 0 |

Obsérvese que mientras la posibilidad de que Hans coma 3 huevos es 1, su probabilidad es completamente baja (0,1); así, un alto grado de posibilidad no implica un alto grado de probabilidad. Sin embargo, si un evento es probable, este debe ser también posible.

Para investigar la relación entre los procesos aleatorios y los difusos, se han realizado diversos trabajos proponiendo una interpretación probabilística de los conjuntos difusos; sin embargo, en todos ellos es necesario que la obtención de la información sea hecha manteniendo el carácter vago de la misma. Un estudio del mismo puede encontrarse en [4].

1.4 Control Fuzzy

Al realizar el diseño de un controlador difuso propiamente dicho, generalmente se hace una diferenciación en el tipo de controlador a diseñar. Normalmente son clasificados como controladores tipo Mamdani y controladores tipo Sugeno, aunque realmente fuzzy logic puede aplicarse y combinarse con muchas otras teorías y obtenerse varios tipos de

controladores. Como se verá más adelante, la diferencia en ellos radica en la parte consecuente de las reglas difusas.

Un controlador fuzzy se compone de 4 módulos:

1) Módulo de fuzzificación: el cual realiza las siguientes funciones:

- El escalamiento o normalización de la entrada, donde los valores físicos de las variables son mapeadas en un universe of discourse normalizado que normalmente será el intervalo $[-1,1]$. Sin embargo, a veces es mejor variar tal rango; por ejemplo para el caso de un PLC es preferible un rango entero de $[0,4095]$ (codificándolo en bits). El valor de las constantes de normalización las obtenemos del rango de funcionamiento del actuador (para la variable de salida) y de los sensores (para las variables de entrada). Sin embargo, esto no nos da una garantía de estabilidad del sistema. Lo aconsejable es obtener esta información del operador humano o de simulaciones.
- La fuzzificación propiamente dicha, es decir, halla la concordancia de un valor crisp con un conjunto fuzzy.

2) Base de conocimiento: consiste en una base de datos y una base de reglas.

La base de datos provee información necesaria para el módulo de fuzzificación y defuzzificación y para la base de reglas. Esta información incluye a las funciones de membresía y a los valores de normalización/desnormalización.

La base de reglas representa en una manera estructurada las políticas de control de un operador experimentado formulado como un conjunto de reglas como sigue:

If <process state> then <control input>

donde <process state> es el antecedente y <control input> el consecuente de la regla.

Dependiendo de la forma del consecuente, podemos identificar dos tipos de controladores:

a) Tipo Mamdani: cuyas reglas son del tipo:

If x es A and y es B then u es C

donde A, B y C son conjuntos fuzzy y (x), (y) y (u) son variables físicas del proceso.

b) Tipo Sugeno: las reglas de este controlador son del tipo:

If x es A and y es B then $u=p*x+q*y+r$

donde (p),(q) y (r) son constantes.

Nótese que a diferencia de las reglas del controlador tipo Mamdani, la salida de las reglas de este controlador no es un conjunto difuso sino un valor preciso (crisp).

Es fácil intuir que la definición de una cantidad grande de reglas hace que el controlador sea menos rápido debido a la cantidad de operaciones que debe de realizar, sin mencionar lo difícil de su sintonización.

El diseño del módulo que contiene la base de conocimiento comprende la elección de los estados del proceso (variables de entrada) y de las variables de control de salida, sus valores lingüísticos y funciones de

membresía correspondientes, la elección de sus factores de normalización y la derivación del conjunto de reglas. Hay que tener en cuenta que cada punto debe pertenecer por lo menos a un conjunto de membresía y que casi todos los puntos deben pertenecer por lo menos a dos conjuntos. Además un punto debería tener valor 1 solamente para un conjunto.

También en el diseño hay que decidir si se van a usar reglas del tipo Mamdani o del tipo Sugeno.

Como se mencionó inicialmente, la lógica difusa hace que el diseño de un controlador pueda hacerse de una forma intuitiva, obteniendo información del conocimiento heurístico de un operador humano que pueda controlar el sistema en forma manual. Este conocimiento puede plasmarse fácilmente en la base de reglas del controlador. Cuando carecemos de tal operador que nos suministre dicha información, lo que se puede hacer es obtener la información desde las simulaciones realizadas con controladores de otro tipo, de fórmulas analíticas derivadas del conocimiento del sistema, de redes neuronales, de metareglas, etc. La sintonización de este tipo de controlador generalmente es hecha mediante prueba y error.

3) Motor de inferencia: su función básica consiste en calcular el valor total de la variable de control de salida basada en la contribución de cada regla en la base de reglas. La salida del módulo de fuzzificación, que representa el estado actual preciso (crisp) del proceso, se compara con el antecedente de cada regla obteniéndose un grado de concordancia. Basado en este grado de concordancia, el valor de control de salida en el consecuente de la regla

es modificado mediante el operador de "implicación". El conjunto de todas estas salidas modificadas es el valor total de salida del controlador.

El diseño de este módulo generalmente consiste en la elección del operador implicación a usar. Generalmente se usa la definición de Mamdani, aunque la de Larsen y la de Gödel también son populares (Fig. I-6):

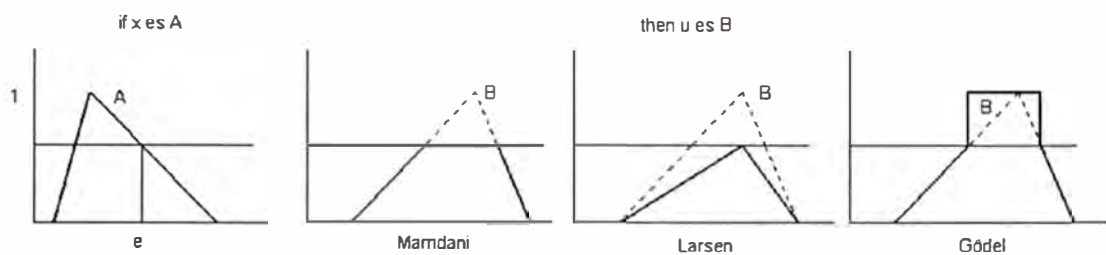


Fig. I-6. Representación gráfica de la implicación: Mamdani, Larsen y Gödel.

En muchos casos, el número de las reglas que definen las diferentes condiciones de entrada es limitado y hay una buena posibilidad que no exista reglas para ciertas entradas. Tales situaciones indefinidas reducen la eficiencia del controlador. En [5] se menciona que para superar esto, el efecto de todas las reglas podría ser extendido alrededor de las situaciones de entrada en todas las direcciones a una distancia determinada por el usuario. Así, para una entrada dada, el algoritmo de control verá si existe una regla correspondiente; si no existe, entonces la regla en la inmediata vecindad será considerada.

En otras ocasiones es posible dejar reglas sin definir cuando en la definición de las funciones de membresía exista "overlapping", es decir,

superposiciones entre las funciones. Si no fuese ese el caso, hay que verificar que la base de reglas cubra todo el rango posible de entradas.

4) Módulo de defuzzificación: el cual presenta las siguientes funciones:

- Realiza la defuzzificación propiamente dicha, es decir, convierte el conjunto de salidas modificadas obtenido del motor de inferencia en un simple valor puntual. Como la defuzzificación (por ejemplo mediante el método del centroide) es un proceso complejo y por consiguiente lento, se puede preferir acondicionar las funciones de membresía a un determinado número de elementos verticales (discretización de las funciones de membresía).
- Realiza la desnormalización de la variable de salida.

El diseño del módulo consiste en la elección del método de defuzzificación. Entre los principales tenemos el del centro de gravedad, el del centro de sumas, el de la altura, el del primer máximo, el de la media de los máximos y el del centro de la más grande área. Por definición, para los controladores tipo Sugeno ciertos métodos de defuzzificación no pueden aplicarse.

En la Fig. 7 se muestra el proceso de fuzzificación, disparo de reglas y defuzzificación para un controlador tipo Mamdani. El antecedente cuenta con dos variables de entrada (θ y ω) y el consecuente de una sola salida (u). Cada una de estas variables tiene definido tres valores

lingüísticos, teniendo asociadas funciones de membresía tipo S, Z y triangular. El método de defuzzificación usado es el del centro de gravedad.

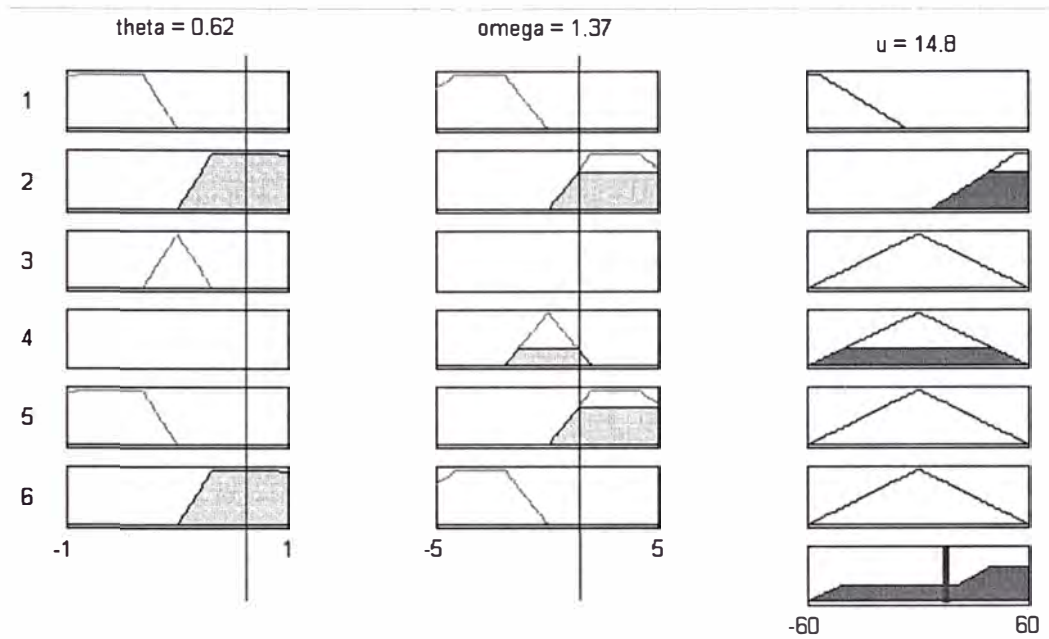


Fig. I-7. Representación gráfica del proceso de fuzzificación, disparo de reglas y defuzzificación en un controlador tipo Mamdani.

CAPÍTULO II MODELO MATEMÁTICO DEL SISTEMA DEL PÉNDULO INVERTIDO

En el sistema del péndulo invertido una varilla está engoznada a un carrito, el cual puede moverse en un riel horizontal en una dimensión. La varilla puede moverse solo en el plano vertical definido por el carro y el riel. El controlador puede aplicar una fuerza sobre el carrito mediante un servomotor DC y a través de un sistema de poleas para ponerlo en movimiento. El diagrama del sistema se muestra en la Fig. II-1. desde donde definimos las siguientes variables y magnitudes:

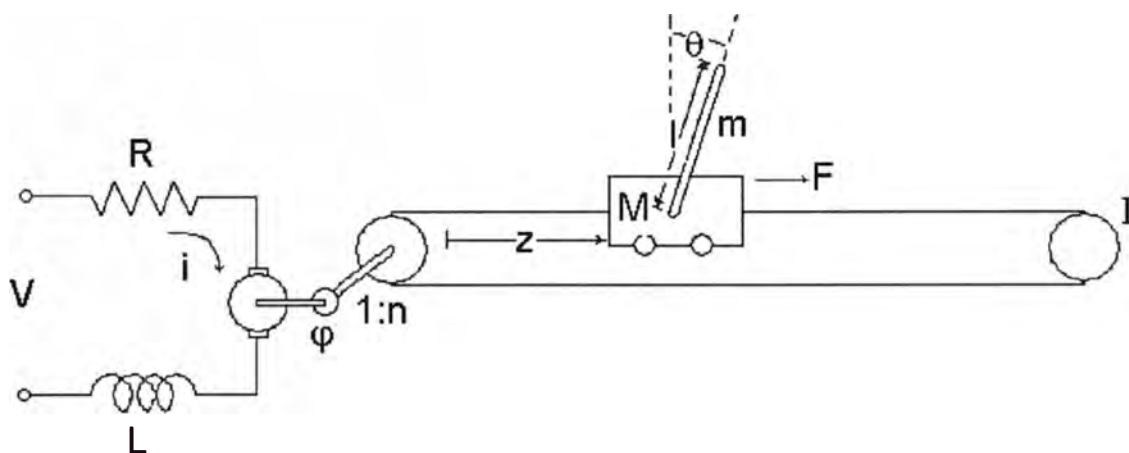


Fig. II-1. Esquema físico del sistema del péndulo invertido.

R: resistencia de armadura del servomotor DC

L: inductancia de armadura del servomotor DC

K_b : constante contra-electromotriz del servomotor

K: constante del servomotor

i: corriente en el servomotor

V: voltaje de control del servomotor

φ : ángulo de giro del eje del servomotor

n: factor de reducción de velocidad

J_m : momento de inercia del eje del servomotor

J_d : momento de inercia de una polea

B_m : fricción viscosa del eje del servomotor

B_d : fricción viscosa en una polea

F: fuerza aplicada al carrito

r: radio de las poleas

m_d : masa de la polea

M: masa del carrito

l: longitud de la varilla

m: masa de la varilla

θ : desplazamiento angular de la varilla respecto a la vertical

z: desplazamiento horizontal del carrito en el riel

I: momento de inercia de la varilla

Para el análisis del subsistema eléctrico del servomotor, asumiremos que la inductancia de armadura es despreciable. Generalmente esta inductancia, en conjunto a la resistencia de armadura, se comporta como un filtro pasabajos para la corriente. Para el caso en que la tensión de control se establezca mediante el uso de la técnica PWM (Pulse Width Modulation) o

modulación por ancho de pulsos, este filtro eliminaría los armónicos. Para el caso en que el valor de la inductancia de armadura del servomotor es 0, se tiene:

$$V = R \cdot i + K_b \cdot \varphi'$$

y el torque eléctrico τ_e resulta:

$$\tau_e = K \cdot i$$

Haciendo el análisis del subsistema mecánico, se tiene que el torque mecánico en el eje del motor está dado por:

$$\tau_m = J_m \varphi'' + 2 \frac{J_d \varphi''}{n^2} + \frac{F \cdot r}{n} + B_m \varphi' + 2 \frac{B_d \varphi'}{n^2}$$

donde:

$$J_d = \frac{m_d r^2}{2}$$

Nótese que los términos relacionados con los torques de las poleas se cuentan 2 veces debido a que el sistema tiene 2 poleas. Además, los torques han sido "reflejados" hacia el lado del motor.

El cálculo de la fuerza de reacción del carrito (tercera Ley de Newton) lo hacemos usando la segunda Ley de Newton respecto al centro de masa del sistema carrito-varilla:

$$F = (M + m) \frac{d^2}{dt^2} \left[\frac{M \cdot z + m \left(\frac{l}{2} \sin \theta + z \right)}{M + m} \right]$$

Si asumimos que no existen zonas muertas producidas por fricciones de tipo estático, tenemos que el torque mecánico y el torque eléctrico son iguales en valor: $\tau_m = \tau_e$, y sumando torques en el eje de la varilla:

$$I.\theta'' = \frac{m.g.l}{2} \sin\theta - \frac{m.z''.l}{2} \cos\theta$$

donde: $I = \frac{m.l^2}{3}$ y existe la relación: $z = \frac{\varphi.r}{n}$

Escogiendo las variables de estado:

$$\begin{aligned} x_1 &= z, \\ x_2 &= z', \\ x_3 &= \varphi, \\ x_4 &= \varphi', \\ u &= V, \end{aligned}$$

se obtienen las siguientes relaciones:

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \frac{u - kajap.x_2 - sajap.D.g.\sin(x_3).\cos(x_3) + sajap.\sin(x_3).x_4^2}{ajap - sajap.D.\cos^2(x_3)}, \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= \frac{u.\cos(x_3) - kajap.x_2.\cos(x_3) + sajap.\sin(x_3).\cos(x_3).x_4^2 - ajap.g.\sin(x_3)}{sajap.\cos^2(x_3) - \frac{ajap}{D}} \end{aligned}$$

donde:

$$\begin{aligned} ajap &= \frac{R.r}{n.k} \left(M + m + m_d + J_m \frac{n^2}{r^2} \right) \\ kajap &= \frac{n}{r} \left[K_b + \frac{R}{K} \left(B_m + 2 \frac{B_d}{n^2} \right) \right] \\ sajap &= \frac{R.r}{K.n} \left(\frac{m.l}{2} \right) \\ D &= \frac{3}{2l} \end{aligned}$$

CAPÍTULO III DISEÑO DEL CONTROLADOR

El diseño del controlador para el sistema del péndulo invertido generalmente es hecho basándose en la teoría del espacio de estados. En el presente trabajo, se realiza un primer diseño usando esta teoría y luego también se le emplea en el diseño que aplica la teoría de la lógica difusa.

3.1 Diseño del Controlador usando la Teoría del Espacio de Estados

Para poder aplicar la teoría del espacio de estados en el diseño del controlador para el sistema del péndulo invertido, se debe de linealizar en primer lugar las relaciones halladas en la sección anterior:

$$X' = A.X + B.u = \begin{bmatrix} 0 & \frac{1}{sajap.D - ajap} & \frac{0}{sajap.D - ajap} & 0 \\ 0 & \frac{kajap}{sajap.D - ajap} & \frac{sajap.D.g}{sajap.D - ajap} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-D.kajap}{sajap.D - ajap} & \frac{-ajap.g.D}{sajap.D - ajap} & 0 \end{bmatrix} .X + \begin{bmatrix} 0 \\ 1 \\ \frac{ajap - sajap.D}{D} \\ \frac{sajap.D - ajap}{sajap.D - ajap} \end{bmatrix} .u$$

en donde la linealización se ha hecho alrededor del origen $X=[0 \ 0 \ 0 \ 0]$.

El programa "programa1.m" escrito en MATLAB, y cuyo listado se encuentra en el Anexo 1, calcula los parámetros de la matriz de control. Los valores físicos utilizados en tal programa, fueron obtenidos de la hoja de especificación técnica de un motor DC del laboratorio de investigación de la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de

Ingeniería, así como de mediciones físicas del sistema del péndulo invertido existente en dicho laboratorio. La elección de la matriz de polos es un tanto arbitraria, de acuerdo a la variable que más nos interesa regular.

El tiempo de muestro se ha establecido en 1 ms. Si se escoge un tiempo de muestreo lo suficientemente pequeño, es prácticamente indiferente utilizar la matriz de control continua (KKS) o la discreta (KKds). En nuestro caso, trabajaremos con la segunda. Así, obtenemos los siguientes valores:

$$KKs=[-6.9083 \ -20.9369 \ -66.1406 \ -15.0739]$$

$$KKds=[-6.8819 \ -20.8969 \ -66.0299 \ -15.0492]$$

los cuales difieren en menos de 1%.

3.2 Diseño del controlador usando Fuzzy Logic

La estrategia de control a seguir será la de dividir el universo de las variables de entrada en regiones, en cada una de las cuales efectuaremos la linealización de la planta. Luego, cada una de estas plantas linealizadas nos dará una regla de control y todas ellas alimentarán la base de reglas. Entonces, el tipo de controlador a diseñar será el de Sugeno. Una demostración de la estabilidad de este tipo de estrategia puede encontrarse en [6].

Al igual que en el diseño del controlador mediante la teoría de espacio de estados, las variables de entrada son la posición y velocidad lineal del carro y la posición y velocidad angular de la varilla. La variable de salida es la tensión de armadura del motor DC.

De la relación general obtenida para el sistema del péndulo invertido en el capítulo II, se ve que ésta es lineal respecto a la posición y velocidad del carro, es decir, no es necesario linealizar el proceso respecto a ellos. Por consiguiente, estas variables no definirán las regiones en las que se dividirá el universo de entrada; esto es, no intervendrán en el antecedente de las reglas fuzzy. En realidad, la linealización se hace en la vecindad del plano definido por la posición y velocidad nulas del carro. Las expresiones generales linealizadas no dependen de los valores de la posición pero sí de la velocidad del carro, como se verá más adelante.

A diferencia de [6], que considera solo la posición angular de la varilla para dividir el “universe of discourse” en regiones, en este trabajo se considerará además la velocidad angular de la misma. Para estas dos variables de entrada definamos tres valores lingüísticos: NEGative, ZERO y POSitive cuyas funciones de membresía respectivas serán del tipo Z, π y S. MATLAB nos ofrece varias otras posibilidades de elección. Los cortes para la posición angular lo definiremos en $\pm\pi/3$ rad (un valor alto) y ± 2 rad/seg para la velocidad angular. Este último valor se eligió mediante un proceso de prueba y error.

Como la variable de salida está determinada mediante el consecuente de reglas tipo Sugeno, ninguna función de membresía fuzzy estará asociada a ella. En su lugar, existe una matriz de control por cada regla fuzzy. Estas las denotaremos como KNN, KNZ, KNP, KZN, KZZ, KZP, KPN, KPZ y KPP donde KZP es la matriz de control para la regla “x3 is **Z**ero and x4 is **P**os”.

Es de notar que no emplearemos normalización de variables. Para las magnitudes lineales no es necesario puesto que no intervienen en la base de reglas. Para la posición angular no tiene mucho sentido y para la salida no se obtiene mucho beneficio, puesto que ésta es calculada directamente de una expresión matemática. La velocidad angular podría ser la única variable a normalizar, pero mantendremos la simplicidad en el diseño.

El universe of discourse para la posición lineal viene dado por la longitud del riel; la de la salida, por el rango de funcionamiento del motor DC. La de la posición angular será $\pm\pi/2$ y la de las velocidades se determinaron por prueba y error.

En cuanto a las operaciones relacionadas con los operadores AND, OR e IF/THEN, elegiremos $\text{prod}()$, $\text{probor}()$ (or probabilístico: $\text{probor}(a,b)=a+b-ab$) y $\text{wtaver}()$ (weight average: promedio ponderado). Normalmente esta elección no es decisiva en el éxito del controlador.

El número de reglas difusas a establecer será el mismo que el número de regiones en las que se divide el universo de entrada, es decir, $3*3=9$ regiones:

If x_3 es NEG AND x_4 es NEG THEN $u = -KNN * X$

If x_3 es NEG AND x_4 es ZERO THEN $u = -KNZ * X$

If x_3 es NEG AND x_4 es POS THEN $u = -KNP * X$

If x_3 es ZERO AND x_4 es NEG THEN $u = -KZN * X$

If x_3 es ZERO AND x_4 es ZERO THEN $u = -KZZ * X$

If x_3 es ZERO AND x_4 es POS THEN $u = -KZP * X$

If x_3 es POS AND x_4 es NEG THEN $u = -KPN * X$

If x_3 es POS AND x_4 es ZERO THEN $u = -KPZ \cdot X$

If x_3 es POS AND x_4 es POS THEN $u = -KPP \cdot X$

donde:

x_3 : posición angular de la varilla

x_4 : velocidad angular de la varilla

u : salida de control

X : vector de estado

La expresión general de la linealización del sistema del péndulo invertido sería:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= \left(\frac{\kappa}{\sigma D \cos^2(x_{3s}) - \alpha} \right) x_2 + \\
 &+ \left(\frac{\sigma [Dg[\alpha + \cos^2(x_{3s})](\sigma D - 2\alpha)] + x_4^2 \cos(x_{3s}) [\alpha - \sigma D(1 + \sin^2(x_{3s}))]}{(\sigma D \cos^2(x_{3s}) - \alpha)^2} \right) x_3 + \\
 &+ \left(\frac{-2\sigma x_{4s} \cdot \sin(x_{3s})}{\sigma D \cos^2(x_{3s}) - \alpha} \right) x_{4s} + \left(\frac{-1}{\sigma D \cos^2(x_{3s}) - \alpha} \right) u \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \left(\frac{-D\kappa \cos(x_{3s})}{\sigma D \cos^2(x_{3s}) - \alpha} \right) x_2 + \\
 &+ \left(\frac{-\alpha Dg \cos(x_{3s}) [\sigma D(1 + \sin^2(x_{3s})) - \alpha] + \sigma D x_{4s}^2 (\sigma D \cos^2(x_{3s}) - \alpha \cos(2x_{3s}))}{(\sigma D \cos^2(x_{3s}) - \alpha)^2} \right) x_3 + \\
 &+ \left(\frac{2D\sigma x_{4s} \sin(x_{3s}) \cos(x_{3s})}{\sigma D \cos^2(x_{3s}) - \alpha} \right) x_{4s} + \left(\frac{D \cos(x_{3s})}{\sigma D \cos^2(x_{3s}) - \alpha} \right) u
 \end{aligned}$$

donde $\alpha = a_j a_p$, $\kappa = k_j a_p$ y $\sigma = s_j a_p$.

El programa "getk.m" escrito en MATLAB, es una función que calcula la matriz de control para una región, dada la posición y velocidad angular de

la varilla alrededor de las cuales se hace la linealización. El listado de este programa se encuentra en el Anexo 2.

Ejecutando este programa se obtienen las siguientes matrices de control:

$$KNN=[-14.3204 \ -33.1209 \ -131.6022 \ -43.2900],$$

$$KNZ=[-14.4767 \ -33.3779 \ -131.6870 \ -43.6220],$$

$$KNP=[-14.3195 \ -33.1195 \ -131.5952 \ -43.4147],$$

$$KZN=[-6.8819 \ -20.8969 \ -65.9564 \ -15.0492],$$

$$KZZ=[-6.8819 \ -20.8969 \ -66.0299 \ -15.0492],$$

$$KZP=[-6.8819 \ -20.8969 \ -65.9564 \ -15.0492],$$

$$KPN=[-14.3195 \ -33.1195 \ -131.5952 \ -43.4147],$$

$$KPZ=[-14.4767 \ -33.3779 \ -131.6870 \ -43.6220],$$

$$KPP=[-14.3204 \ -33.1209 \ -131.6022 \ -43.2900]$$

de donde se ve que no era necesario dividir en regiones según la velocidad angular, ya que la variación de la matriz de control respecto a esta variable es insignificante. En tal caso, al dividir el universo de entrada solamente según la variable x_3 (posición angular), el número de reglas baja a tres. Compárese con el número de reglas que habría que definir para un controlador tipo Mamdani: $3*3*3*3=81$ reglas (asumiendo tres variables lingüísticas para todas las variables de entrada). La diferencia estriba en que si bien las variables lingüísticas dividen el universo de entrada en 81 regiones, en el controlador tipo Sugeno estas regiones determinan una matriz de control, la cual es independiente de tres de las cuatro variables, es decir, que con solo tres reglas se cubre todo el universo. Esto no sucede con

el controlador tipo Mamdani, donde cada región determina una salida de control que obviamente depende de las cuatro variables de entrada.

CAPÍTULO IV SIMULACIONES

Las simulaciones fueron hechas usando MATLAB versión 5.3.0.10183 para Windows, SIMULINK versión 3.0 y el toolbox Fuzzy Logic versión 2.0.1. El diagrama en SIMULINK empleado se aprecia en las Figs. IV-1 y IV-2:

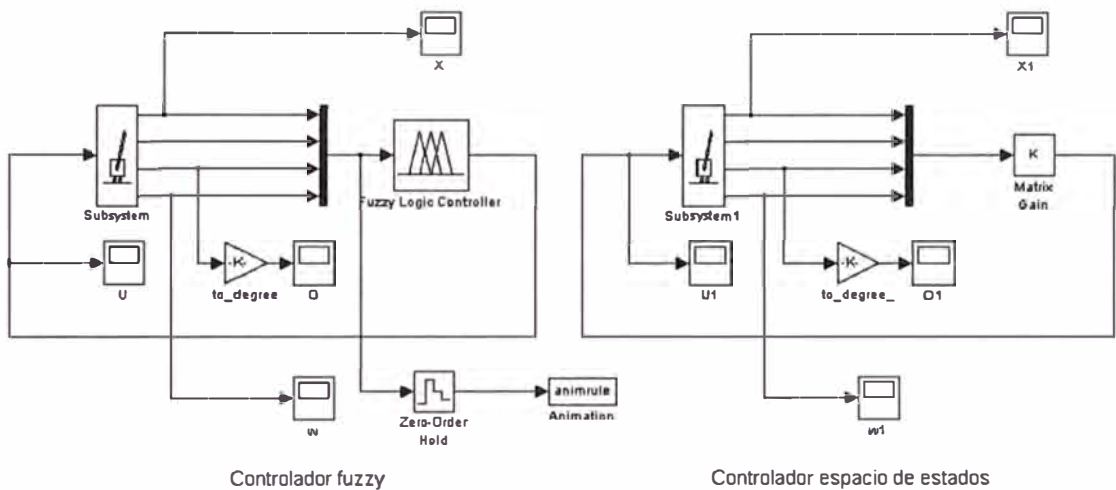
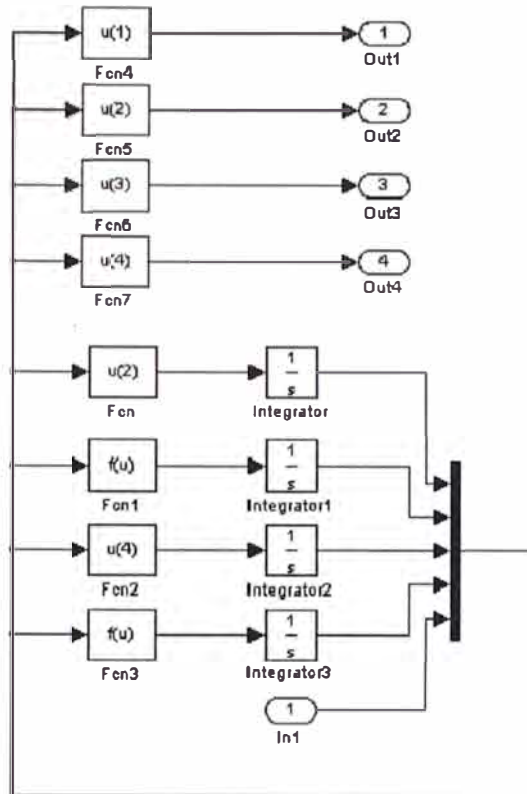


Fig. IV-1. Esquema del diagrama de bloques del sistema en SIMULINK



Modelamiento del sistema del péndulo invertido

Fig.IV-2. Modelamiento del subsistema correspondiente a la planta (péndulo)

Como se observa en las Figs. IV-1 y IV-2, en las simulaciones se ha preferido mostrar el desempeño para una regulación del sistema en lugar de un control de posición. De esta manera, se aprecia mejor la ventaja del control fuzzy. Las condiciones iniciales son establecidas en los integradores del diagrama de bloques de la Fig. IV-2.

Los bloques de ganancias escalares (bloques triangulares) sirven para obtener el gráfico de la posición angular de la varilla en grados sexagesimales. El resto de las variables se visualizan en unidades del Sistema Internacional: metros, metros/segundo, radianes/segundo y voltios

para las variables x_1 (posición lineal del carro), x_2 (velocidad lineal del carro), x_4 (velocidad angular de la varilla) y U (voltaje de armadura del servomotor DC) respectivamente. El retenedor de orden cero y el bloque de animación son para visualizar el disparo de las reglas en el tiempo.

Respecto a la matriz de control K del controlador diseñado por el espacio de estados, el cual se observa a la derecha en la Fig. IV-1, como ha sido diseñada alrededor del punto $X=[0 \ 0 \ 0 \ 0]$, tiene un valor igual a la matriz de control KZZ del controlador fuzzy (en realidad, se ha tomado el negativo de los valores de todas las matrices de control en la simulación, para evitar colocar un bloque con ganancia -1 en el lazo de control).

A pesar de que en la etapa de diseño se vio que no es necesario dividir el universo de entrada respecto a la velocidad angular, esta variable va a ser considerada en las simulaciones. Las funciones de membresía escogidas en MATLAB para las variables lingüísticas y sus “universe of discourse” son:

X_1 y X_2 :

$ZERO = \text{trapmf} [-45 \ -40 \ 40 \ 45]$ sobre el universo $[-50 \ 50]$.

es decir, una única función trapezoidal para simular una total pertenencia sobre todo el dominio de la variable, puesto que las reglas no dependen de estas variables, pero la salida de control sí (Fig. IV-3).

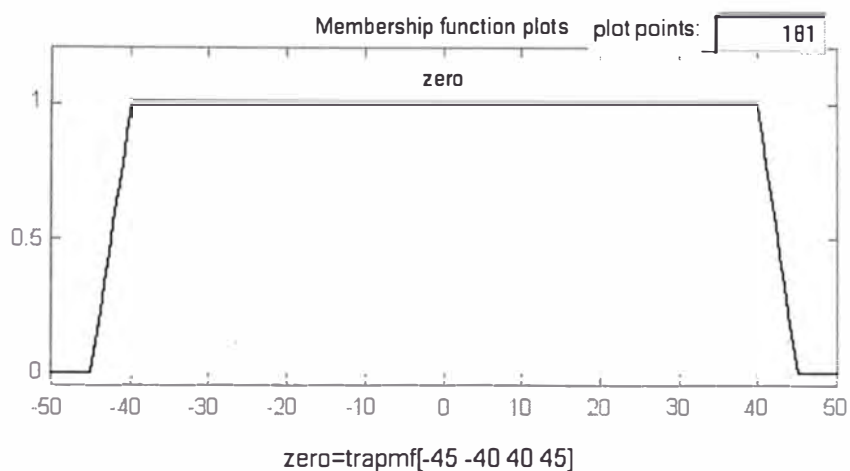


Fig. IV-3. Función de membresía para las variables x_1 y x_2 .

x_3 :

$$\text{NEG}=\text{zmf}[-\pi/3 \ 0]$$

$$\text{ZERO}=\text{pimf}[-\pi/3 \ 0 \ 0 \ \pi/3]$$

$$\text{POS}=\text{smf}[0 \ \pi/3]$$

sobre el universo $[-2 \ 2]$ (Fig. IV-4).

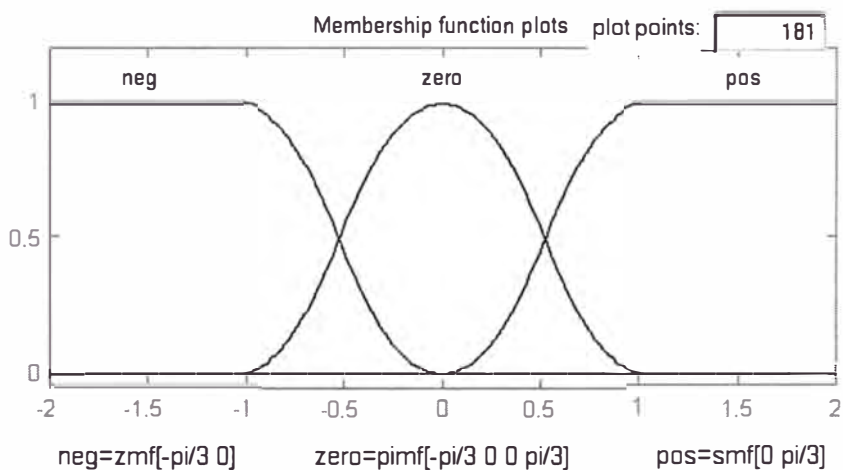


Fig. IV-4. Funciones de membresía para la variable x_3 .

x_4 :

NEG=zmf[-2 0]

ZERO=pimf[-2 0 2]

POS=smf[0 2]

sobre el universo [-5 5] (Fig. IV-5).

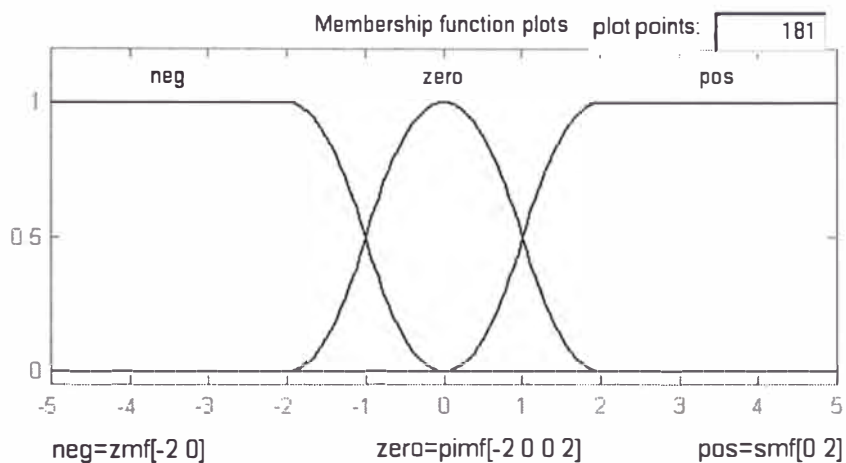


Fig. IV-5. Funciones de membresía para la variable x_4 .

u:

kpp, kpz, kpn, kzp, kzz, kzn, knp, knz, knn = lineal[K 0]

donde K es la matriz de control diseñada en el capítulo anterior para la región específica.

La base de reglas viene dada por:

If (x_3 is NEG) and (x_4 is NEG) then (u is knn)

If (x_3 is NEG) and (x_4 is ZERO) then (u is knz)

If (x_3 is NEG) and (x_4 is POS) then (u is knp)

If (x_3 is ZERO) and (x_4 is NEG) then (u is kzn)

If (x_3 is ZERO) and (x_4 is ZERO) then (u is kzz)

If (x3 is ZERO) and (x4 is POS) then (u is kzp)

If (x3 is POS) and (x4 is NEG) then (u is kpn)

If (x3 is POS) and (x4 is ZERO) then (u is kpz)

If (x3 is POS) and (x4 is POS) then (u is kpp)

todas con el mismo peso. En la Fig. IV-6 se muestra la base de reglas en forma gráfica.

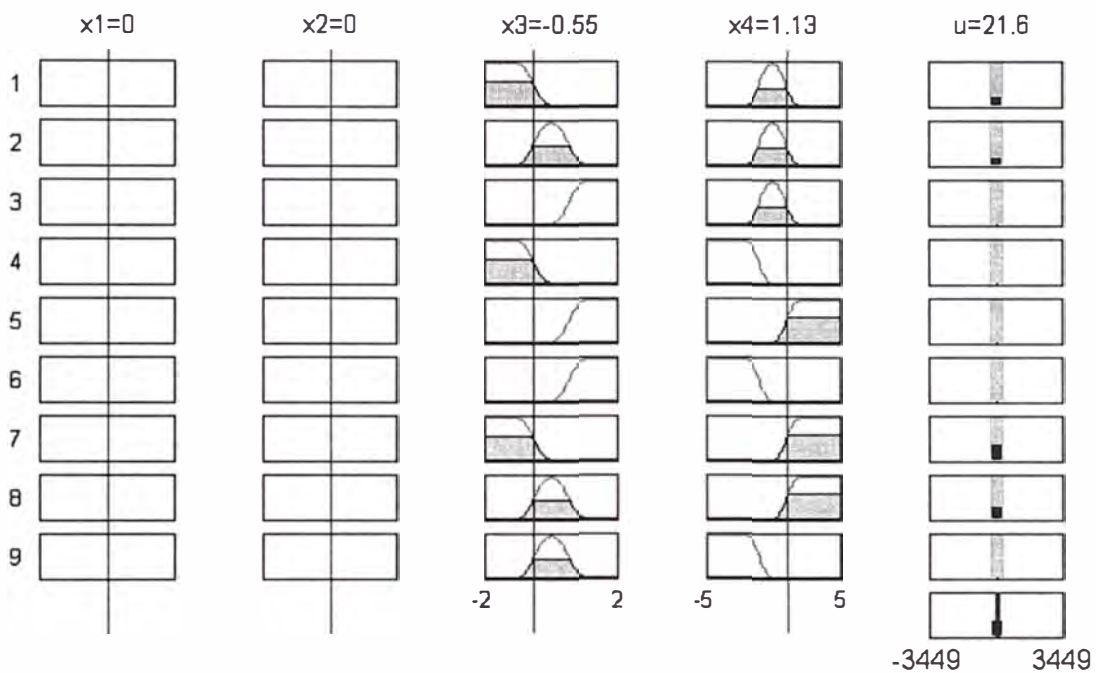


Fig. IV-6. Representación gráfica de la base de reglas y disparo de las mismas.

donde se ha elegido las operaciones “producto”, “or probabilístico” ($\text{probor}(a,b)=a+b-ab$) y “promedio ponderado” para los operadores “and”, “or” y el proceso de defuzzificación.

En [7] puede encontrarse la definición dada por MATLAB para las funciones de membresía utilizadas en este trabajo, así como el uso del toolbox Fuzzy Logic.

Para una condición inicial $X=[0 \ 0 \ \pi/3 \ 0]$ se obtuvieron los resultados mostrados en las Figs. 15 y 16. En todos los gráficos obtenidos mediante simulaciones, el nombre de la variable se encuentra en la esquina superior izquierda del subgráfico correspondiente. Las unidades de medida para estas variables corresponden al establecido en el Sistema Internacional, a excepción de la posición angular de la varilla que se expresa en grados sexagesimales. Yendo de arriba hacia abajo y luego de izquierda a derecha: metros, metros/segundo, radianes/segundo, grados sexagesimales y voltios versus segundos para las variables x_1 , x_2 , x_4 , x_3 y U respectivamente.

Comparativamente, el controlador diseñado por teoría del espacio de estados llega a regular la planta en un tiempo menor que el controlador fuzzy, pero el valor de su segundo sobreimpulso es mucho mayor. Así, para la posición angular de la varilla (variable x_3 , subgráfico superior derecho en las Figs. 15 y 16) el controlador por espacio de estados llega a regular la variable en dos segundos mientras que el controlador fuzzy en tres, pero el segundo sobreimpulso toma un valor muchísimo mayor (35° versus menos de 7°). Este comportamiento también se aprecia en los otros subgráficos: todas las variables son reguladas más rápidamente por el controlador diseñado por el espacio de estados, pero presentan sobreimpulsos mayores.

Valores altos de sobreimpulso se obtienen generalmente cuando un controlador genera señales de control que logran una regulación rápida.

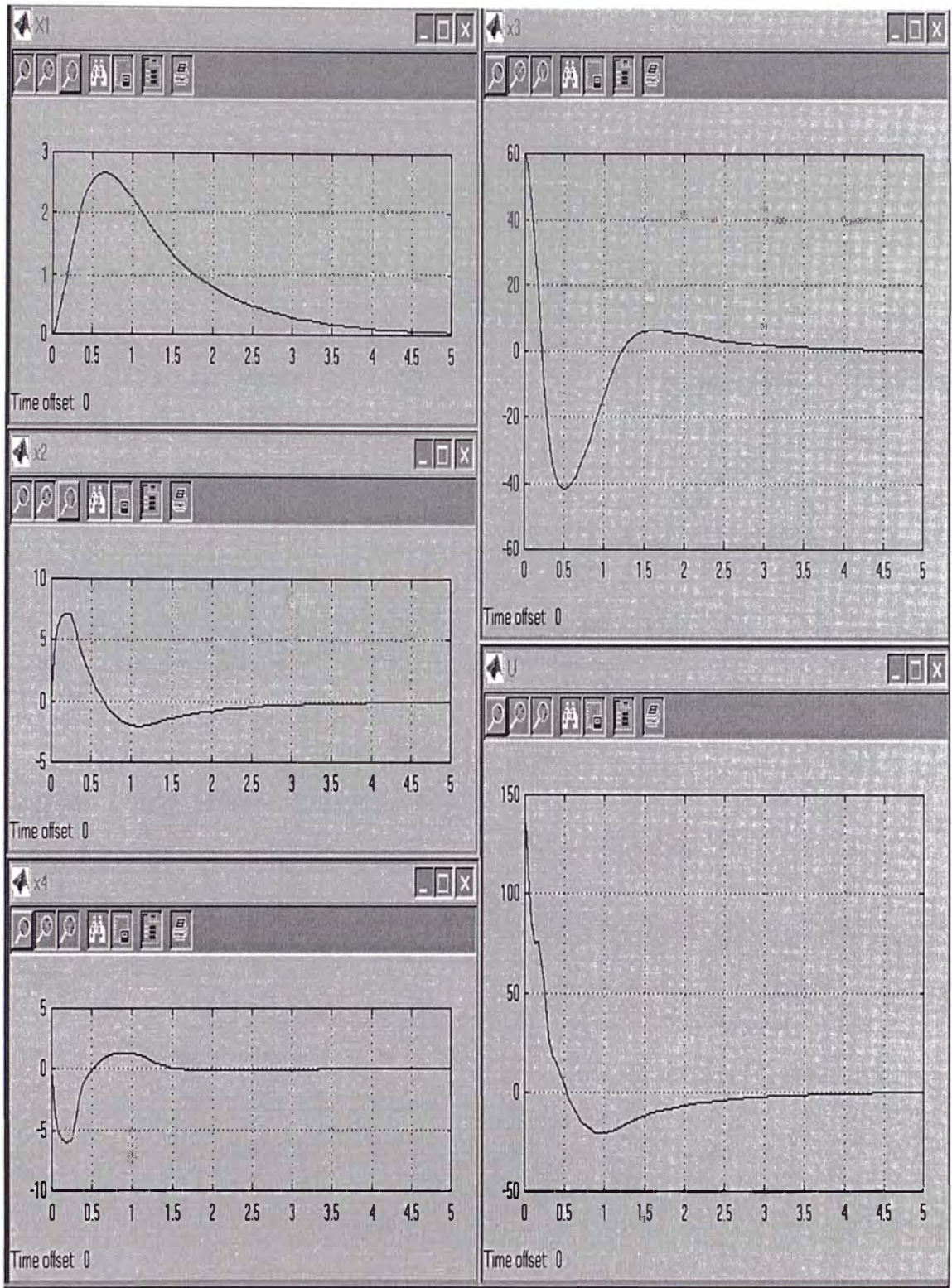


Fig.IV-7. Gráficas de las variables del sistema del péndulo invertido para el controlador fuzzy y condiciones iniciales $X=[0 \ 0 \ \pi/3 \ 0]$.

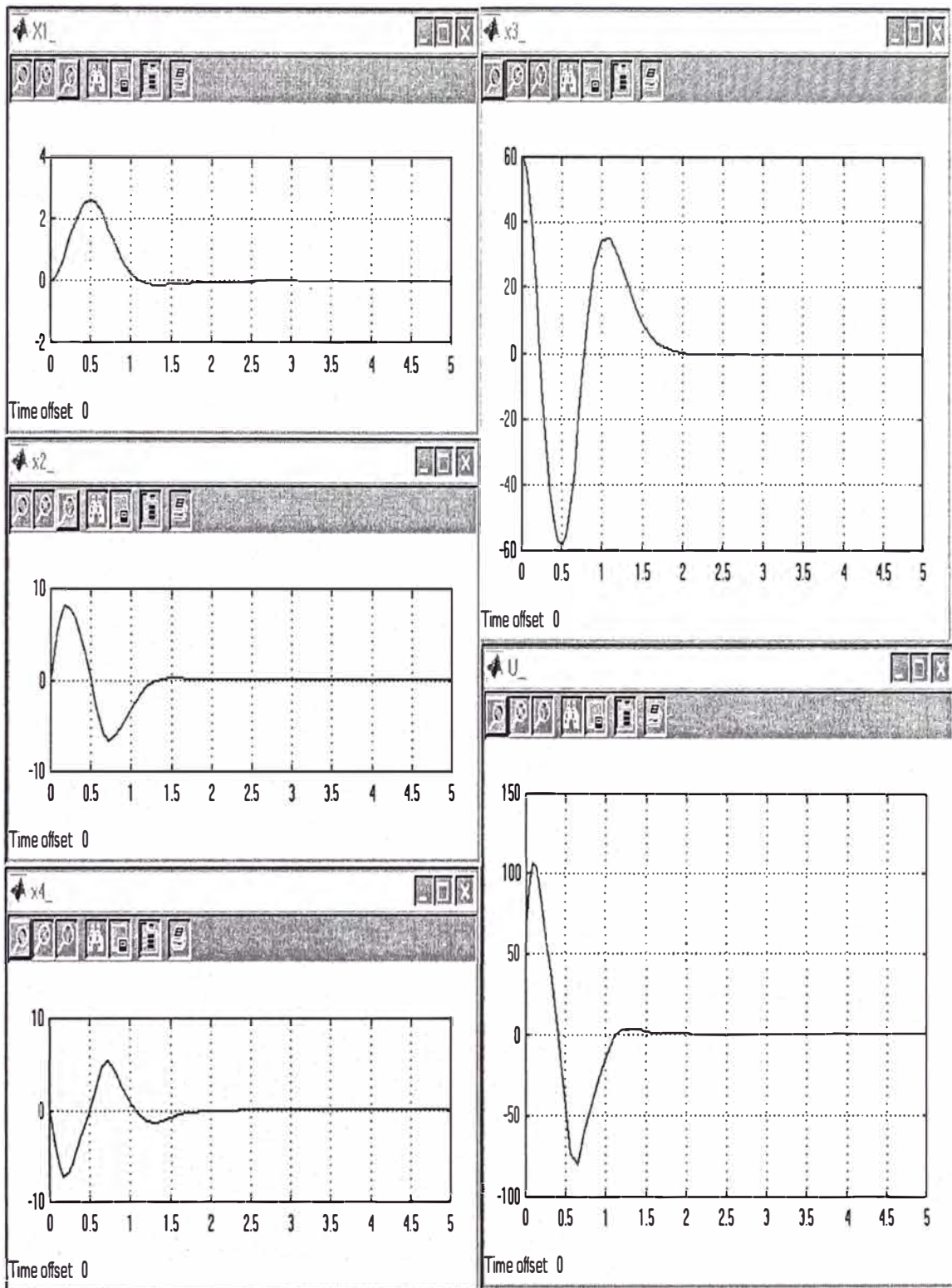


Fig.IV-8. Gráficas de las variables del sistema del péndulo invertido para el controlador por espacio de estado y condiciones iniciales $X=[0 \ 0 \ \pi/3 \ 0]$.

Para hacer una mejor comparación entre el controlador fuzzy y el controlador por el espacio de estados, rediseñemos la matriz de control de este último, multiplicando su matriz de polos J (ver listado del programa “programa1.m” en el Anexo 1) por 0,75, es decir, haciéndolo más lento. Para este segundo controlador por espacio de estados, se obtienen las matrices de control siguientes:

$$KKs=[-2.1858 \ -14.3755 \ -40.8669 \ -9.3565]$$

$$KKds=[-2.1821 \ -14.3684 \ -40.8852 \ -9.3610]$$

y los resultados que vemos en la Fig. IV-9.

Con el resultado de la simulación de este nuevo controlador es más fácil hacer una comparación con el controlador fuzzy. Las gráficas son similares, pero el controlador fuzzy obtiene mejores resultados:

- 1) Posición lineal del carro (x_1 , subgráfico superior izquierdo): en el controlador fuzzy, el tiempo de asentamiento es ligeramente menor y el sobreimpulso es 2,7 metros en comparación con 3,4 metros del otro.
- 2) Velocidad lineal del carro (x_2 , subgráfico medio izquierdo): el tiempo de asentamiento es practicamente el mismo para ambos controladores, pero el primer pico ocurre primero para el controlador fuzzy. El valor del primer sobreimpulso es de aproximadamente 7m/s para ambos (un poco mayor el del controlador fuzzy) y del segundo de -2m/s (un poco mayor en valor absoluto el del controlador por espacio de estados).
- 3) Posición angular de la varilla (x_3 , subgráfico superior derecho): el tiempo de asentamiento del controlador fuzzy es ligeramente menor y el primer sobreimpulso ocurre 0,25 segundos antes y es ligeramente menor.

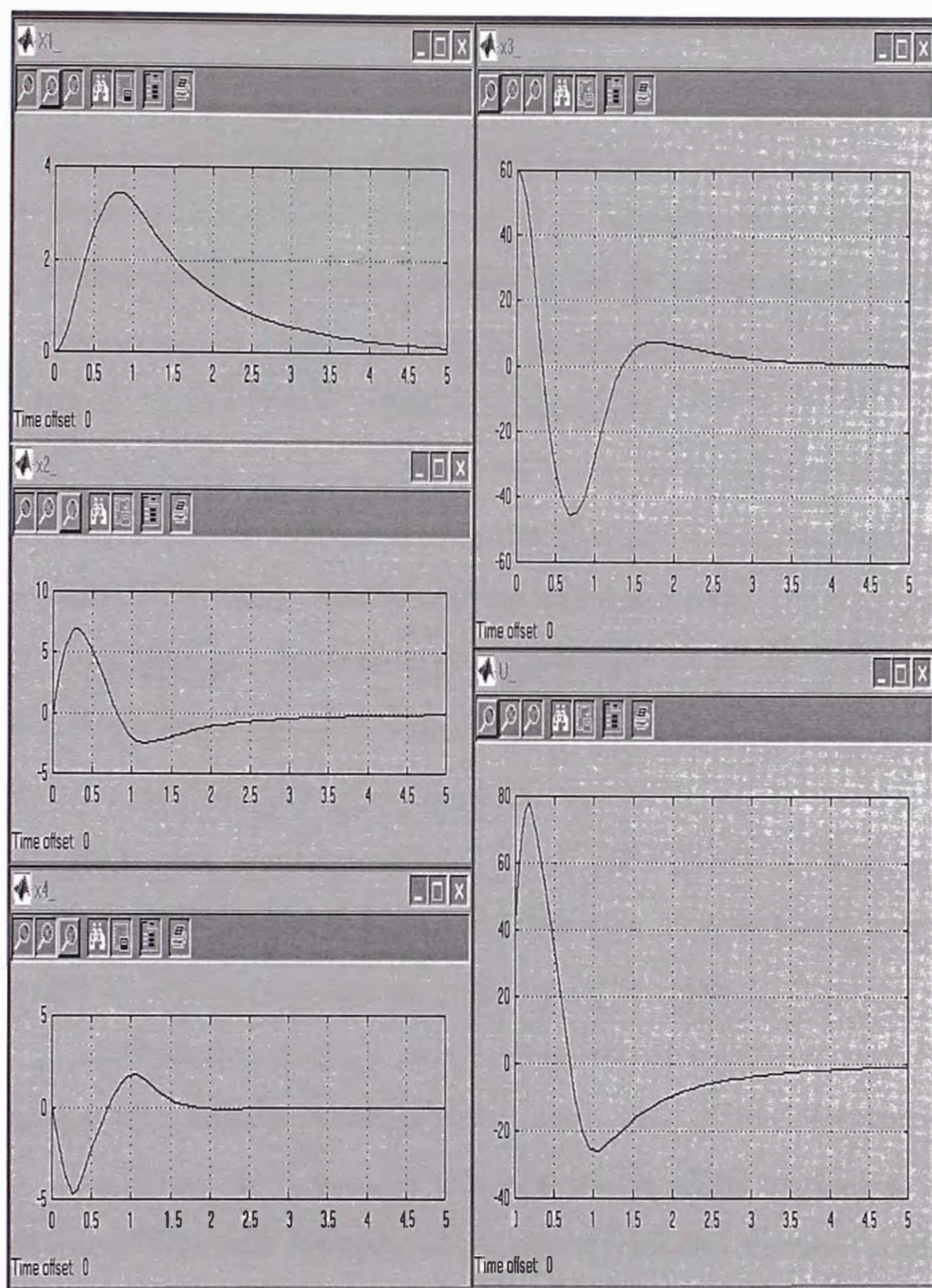


Fig.IV-9. Gráficas de las variables del sistema del péndulo invertido para el segundo diseño del controlador por espacio de estado y condiciones iniciales $X=[0 \ 0 \ \pi/3 \ 0]$.

- 4) Velocidad angular de la varilla (x_4 , subgráfico inferior izquierdo): en el controlador fuzzy los sobreimpulsos ocurren antes y su segundo sobreimpulso es menor, pero el primero es mayor.
- 5) Voltaje de armadura del servomotor DC (U , subgráfico inferior derecho): el controlador fuzzy genera un valor inicial de control mayor que el controlador por espacio de estados. Su sobreimpulso es ligeramente inferior, lo mismo que su tiempo de asentamiento.

Resumiendo, se ve un desempeño similar en ambos controladores siendo el controlador fuzzy un poco superior en cuanto a comportamiento en casi todos los puntos, a excepción del valor inicial de control.

Sin embargo, la diferencia en el desempeño se aprecia cuando se hace la simulación para una condición inicial $X=[0 \ 0 \ 70\pi/180 \ 0]$ (notar que $70\pi/180\text{rad}=70^\circ$) tal como se muestra en las Figs. IV-10 y IV-11.

Para estas condiciones iniciales vemos que el controlador fuzzy aún puede estabilizar el sistema para un ángulo grande, mientras que el control basado en la teoría del espacio de estados no.

¿Qué pasaría si para el control por espacio de estados en lugar de definir $K=K_{ZZ}$ definiese $K=K_{NN}$? ¿sería estable para $x_3=70^\circ$?. Realizando las simulaciones se comprueba que variando la matriz de control, el controlador diseñado mediante el espacio de estados también puede estabilizar una desviación de 70° de la varilla, pero a costa de un menor desempeño para las regiones correspondientes a desviaciones angulares pequeñas. Así, para una condición inicial $X=[0 \ 0 \ \pi/5 \ 0]$ se obtienen los resultados mostrados en las Figs. IV-12 y IV-13.

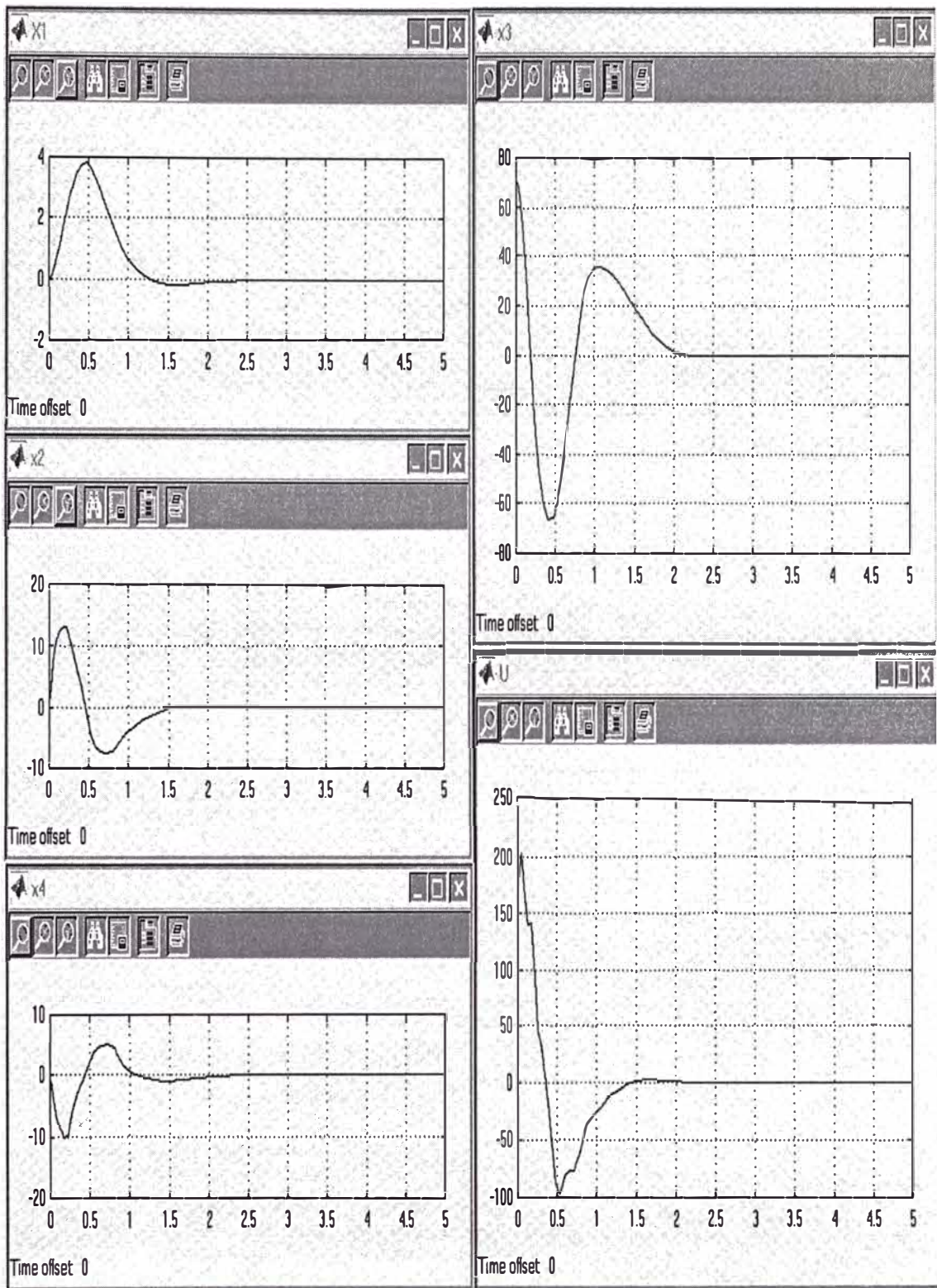


Fig.IV-10. Gráficas de las variables del sistema del péndulo invertido para un controlador fuzzy y condiciones iniciales $X=[0 \ 0 \ 70^\circ \ 0]$.

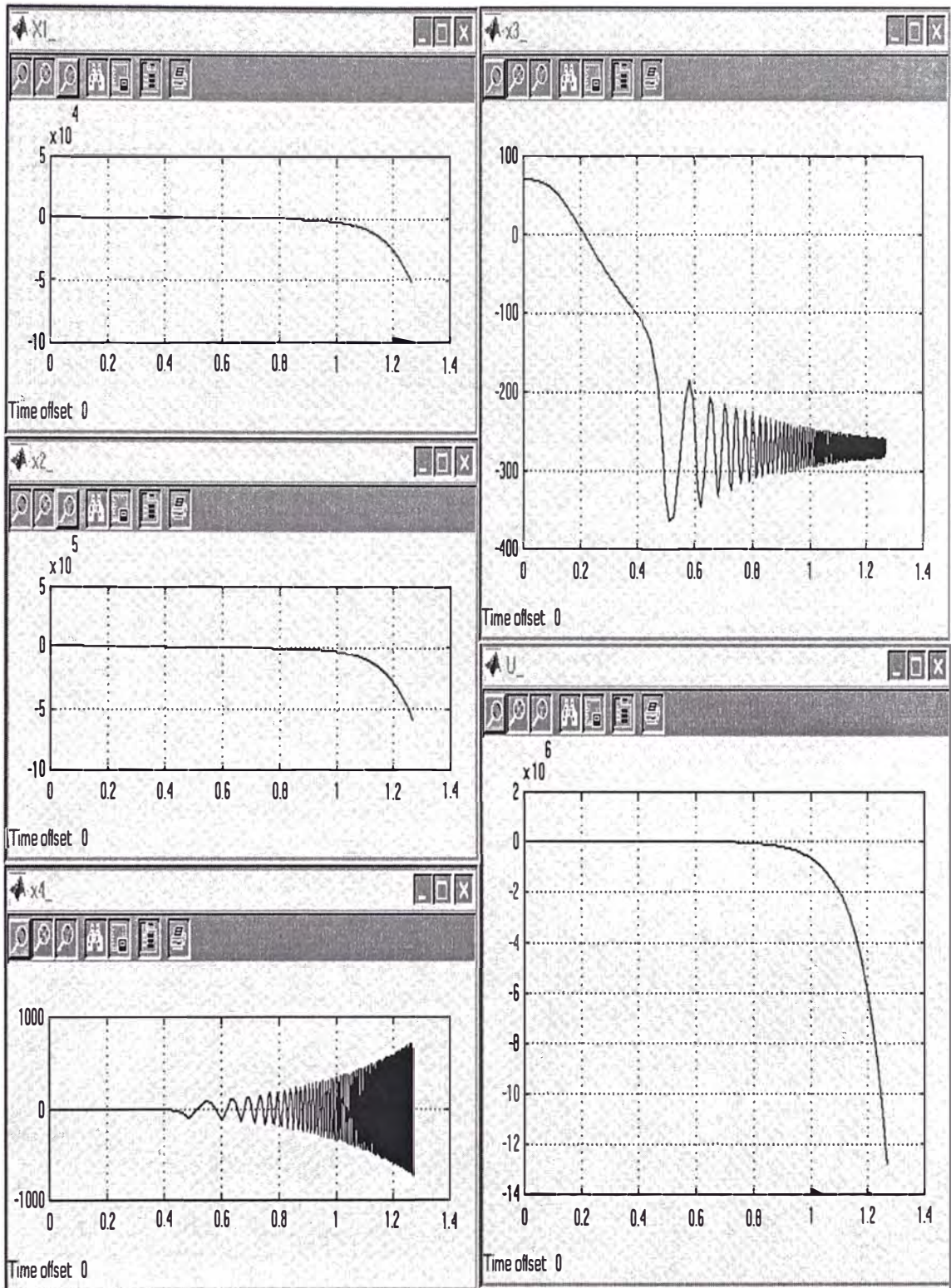


Fig.IV-11. Gráficas de las variables del sistema del péndulo invertido para un controlador por espacio de estado y condiciones iniciales $X=[0 \ 0 \ 70^\circ \ 0]$.

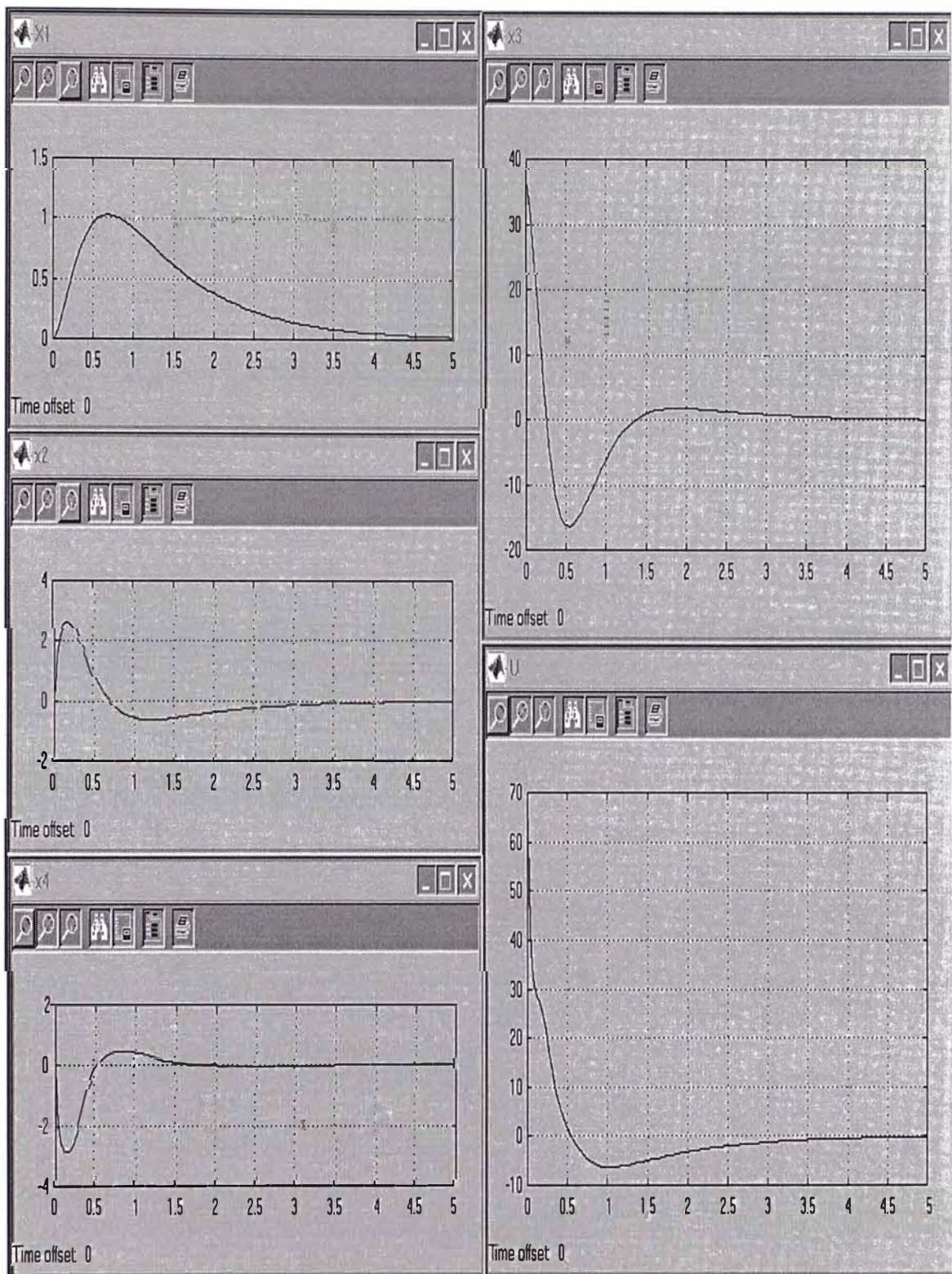


Fig.IV-12. Gráficas de las variables del sistema del péndulo invertido para un controlador fuzzy y condiciones iniciales $X=[0 \ 0 \ \pi/5 \ 0]$.

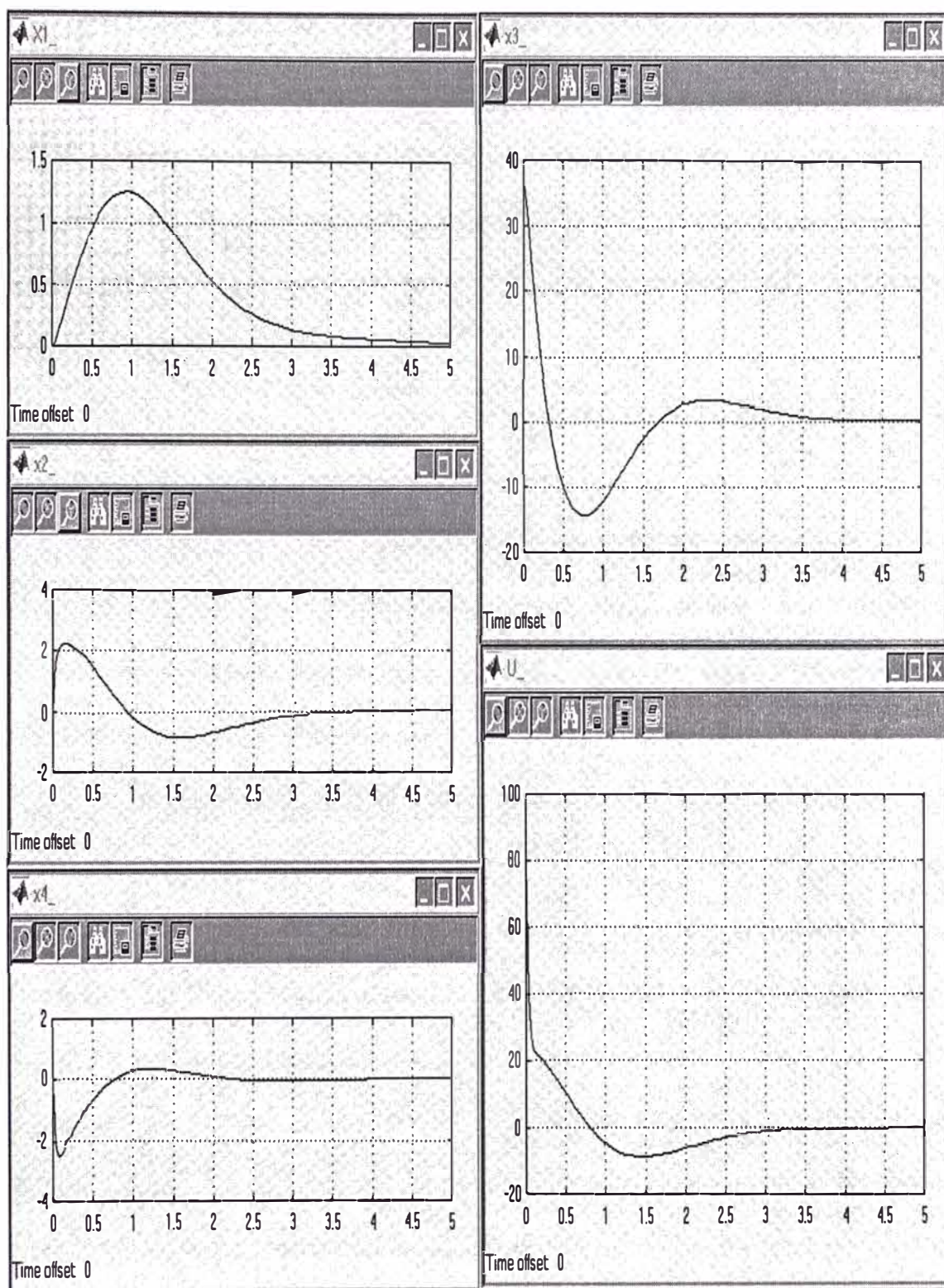


Fig.IV-13. Gráficas de las variables del sistema del péndulo invertido para un controlador por espacio de estado y condiciones iniciales $X=[0 \ 0 \ \pi/5 \ 0]$.

Para estas condiciones iniciales ($X=[0 \ 0 \ \pi/5 \ 0]$), el controlador fuzzy ha logrado menores tiempos de asentamiento para todas las variables y sobreimpulso menor para la variable x_1 . En el resto de las variables, ha logrado un segundo sobreimpulso menor pero un primero mayor. La ley de control es también menor que en el caso del controlador diseñado por el espacio de estados.

En la Fig. IV-14 se muestra el resultado de la simulación para este controlador con condiciones iniciales $X=[0 \ 0 \ \pi/3 \ 0]$.

Si comparamos estos resultados con los mostrados en las Figs. IV-7 y IV-8, las cuales son los resultados de simulaciones con idénticas condiciones iniciales, vemos que el desempeño de este controlador es ligeramente inferior al obtenido por el controlador fuzzy.

Es fácil entonces darse cuenta que la ventaja del controlador fuzzy viene dada por su flexibilidad en especificar los parámetros de control de acuerdo a las regiones de entrada. Para el caso de los controladores diseñados por el espacio de estados, estos tienen un buen desempeño en la región alrededor del punto en el cual fue diseñado, pero su desempeño baja al alejarse de él. Con el controlador fuzzy no sucede esto, pues cuando el estado del sistema se aleja del punto alrededor del cual se hizo el diseño de una regla, esta deja de tener preponderancia en la salida final de control, mientras que otra regla –aquella diseñada alrededor del punto al cual nos acercamos- comienza a afectar más a la salida de control.

La mayor flexibilidad del diseño fuzzy se aprecia mejor observando la superficie de control para cada controlador (Figs. IV-15 y IV-16).

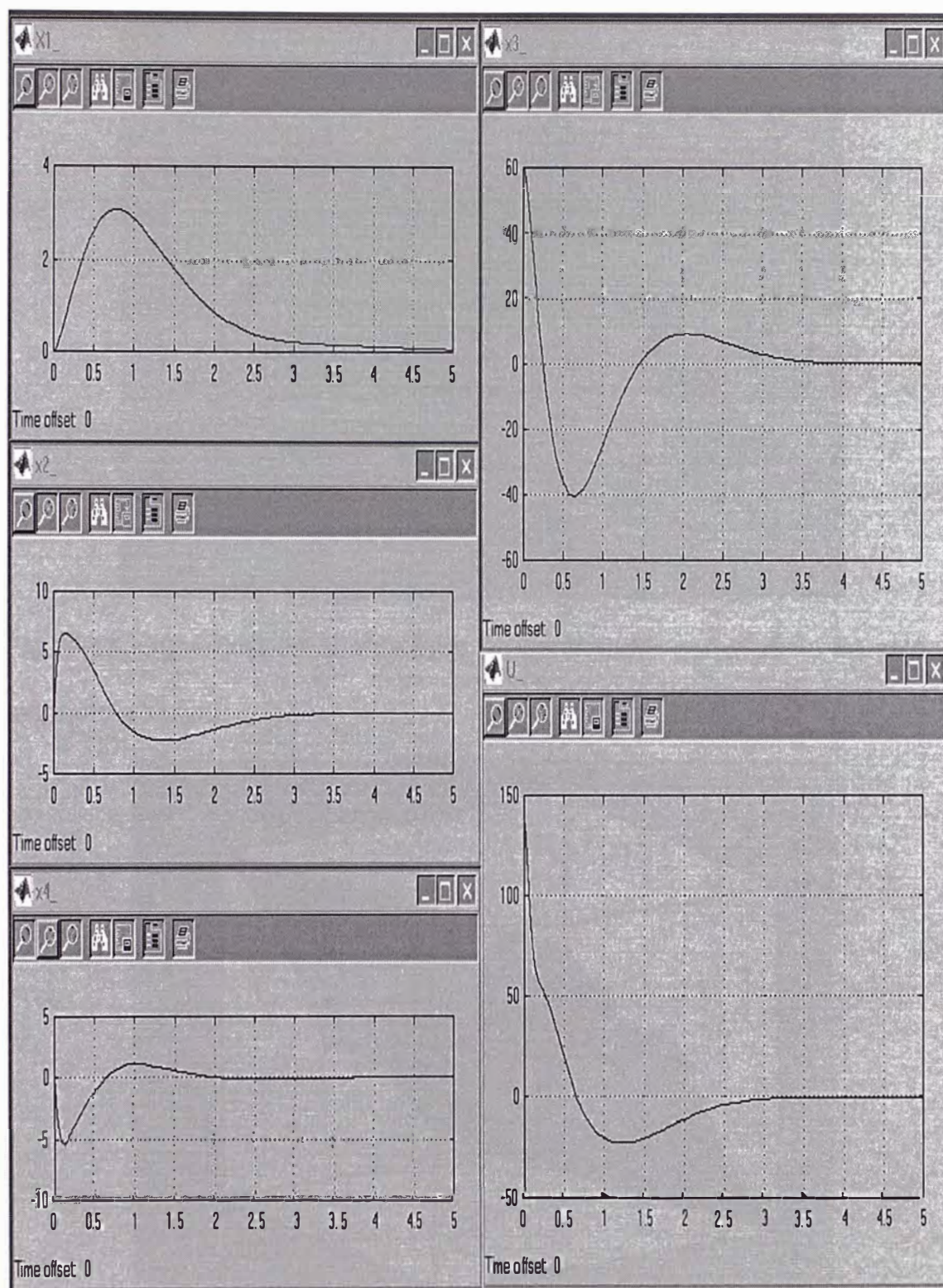


Fig.IV-14. Gráficas de las variables del sistema del péndulo invertido para el controlador por espacio de estado y condiciones iniciales $X=[0 \ 0 \ \pi/3 \ 0]$.

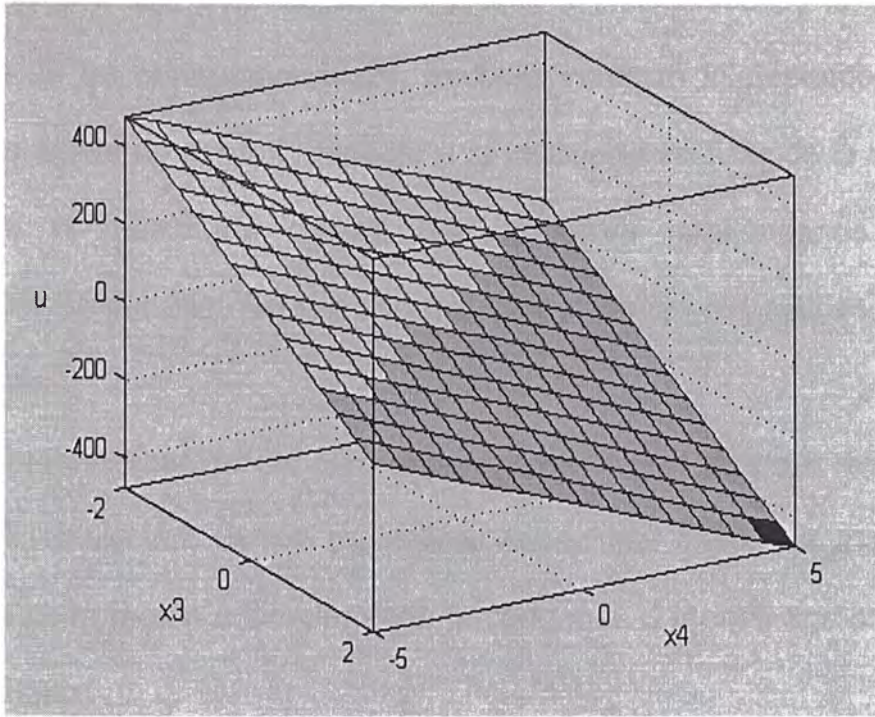


Fig.IV-15. Superficie de control para un controlador por espacio de estado respecto a x_3 y x_4 .

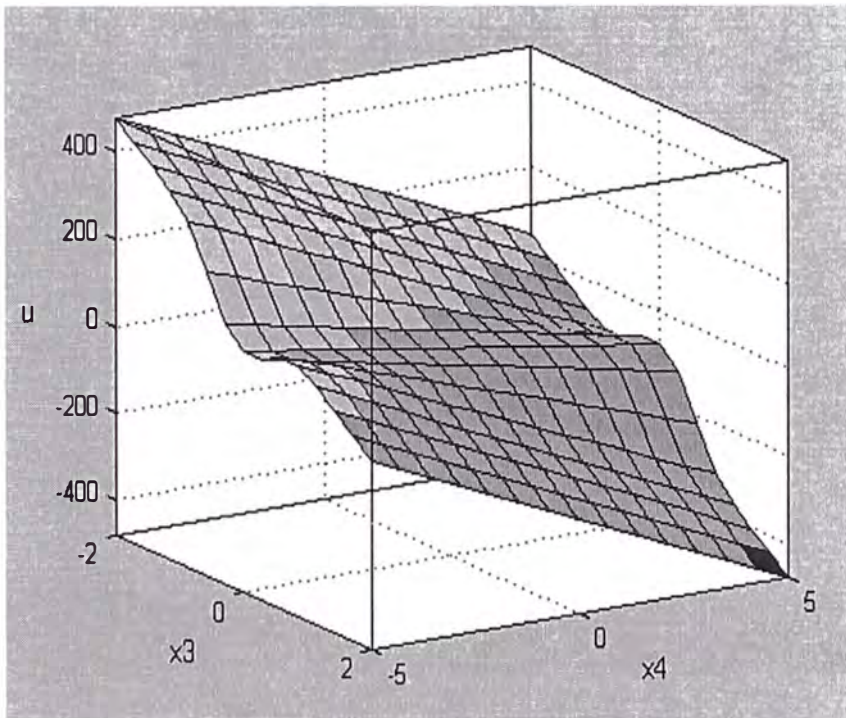


Fig.IV-16. Superficie de control para un controlador fuzzy respecto a x_3 y x_4 .

Estas gráficas, además de comprobar la flexibilidad de la estrategia de control de un controlador fuzzy, también verifican la dependencia casi lineal de la salida de control respecto a la velocidad angular de la varilla, es decir, que la matriz de control casi no varia moviéndonos en esa coordenada. Es por eso que esa variable bien pudo ser obviada en el diseño de las reglas.

La personalización del control para cada región puede ir más allá de la simple linealización, ya que podríamos especificar diferentes matrices de polos para cada región o simplemente hacerlo más o menos agresivo según nuestro criterio.

CONCLUSIONES

Luego de la exposición del método de diseño de un controlador fuzzy y de las simulaciones respectivas para el sistema del péndulo invertido, podemos establecer lo siguiente:

1. La teoría de conjuntos fuzzy involucra teoremas complicados pero la mayoría de ellos no tienen relación con el desarrollo de algoritmos de control fuzzy; estos generalmente son aplicados en sistemas de inferencia, manejo de incertidumbres, computación con palabras, etc.
2. El motivo de usar la lógica difusa en control de procesos es que si bien el desempeño difuso puede no ser superior al convencional, la lógica difusa permite ser aplicada con éxito en procesos típicamente controlados por un operador con cierto éxito. Así, se puede establecer un conjunto de reglas que pueden ser utilizadas directamente en la formulación del algoritmo fuzzy que será aplicado luego al control del proceso, logrando la ventaja de reducir el requerimiento de una constante intervención del operador, y de esta manera se mejora el desempeño del sistema debido a que en su implementación se está empleando la mejor práctica de operación.
3. La elección de la forma de los conjuntos fuzzy en la etapa de diseño es completamente arbitraria y depende de las preferencias del usuario,

aunque existen ciertas guías de elección. Características tales como la altura, simetría, ancho y grado de solapamiento influyen en cierto modo el desempeño y características de operación del controlador.

4. En la etapa de diseño, al hacer un primer análisis para diseñar un controlador difuso tipo Mamdani, llegamos a la conclusión que el diseño tiende a volverse dificultoso cuando interviene un número alto de variables, ya que el número de reglas difusas a definir crece en forma exponencial con ellas, dependiendo de la cantidad de etiquetas elegida.
5. El empleo de un control fuzzy nos da la flexibilidad de personalizar el control para regiones específicas del dominio de entrada, logrando superficies de control más complejas.
6. Los controladores con lógica difusa también se emplean en procesos cuyo modelamiento es demasiado complejo, o en casos extremos, cuando no se pueden modelar convencionalmente.

ANEXO 1
LISTADO DEL PROGRAMA “PROGRAMA1.M”

```

% programa1.m

% Script que calcula la matriz de control para el sistema del péndulo
% invertido usando la teoría de espacios de estados. Establece los valores
% de las variables en el Matlab workspace para poder ser utilizados por el
% modelo hecho en simulink.

G=9.8;                %aceleracion debido a la gravedad

conversion=7.06155e-3;

                    %conversion ingles -> MKS=0.453592/16*2.54e-2*g
                    %onzas_fuerza.inches -> Newtons.metros

R=7.38;                %resistencia de armadura
r=0.0648;              %radio de la polea
M=0.92;                %masa del carrito
m=0.060;               %masa de la varilla
md=0.2;                %masa de la polea
Jm=(2.775e-4)*conversion; %momento de inercia del motor
n=19.741;              %reduccion del motor (>1)
K=4.4*conversion;      %constante de armadura del motor
Kb=31.0352e-3;         %constante de la fuerza contraelectromotriz del motor
Bm=(2.5974e-4)*conversion; %friccion viscosa del eje del motor
Bd=0;                  %friccion viscosa en el eje de la polea
l=0.785;               %longitud de la varilla

% Matriz de polos continuos
J = [ -6    0    0    0

```

```

    0    -1    0    0
    0    0    -7    0
    0    0    0    -3];

JJ = poly(J);

% Matriz de polos discretos

Ts=1/1000;                %tiempo de muestreo

Jd = [exp(J(1,1)*Ts)      0      0      0
      0      exp(J(2,2)*Ts)  0      0
      0      0      exp(J(3,3)*Ts)  0
      0      0      0      exp(J(4,4)*Ts)];

JJd = poly(Jd);

ajap=R*r*(M+m+ md+Jm*n*n/(r*r))/(n*K);
kajap=n*(Kb+R*(Bm+2*Bd/(n*n))/K)/r;
sajap=R*r*(m*I/2)/(K*n);

D=3/2I;

A=zeros(4); B=zeros(4,1); C=[1 0 0 0];
A(1,2)=1; A(3,4)=1;
var1=1/(sajap*D-ajap);
A(2,2)=kajap*var1;
A(2,3)=sajap*D*g*var1;
B(2,1)=-var1;
A(4,2)=-D*A(2,2);

```

```
A(4,3)=-ajap*g*D*var1;
```

```
B(4,1)=D*var1;
```

```
MMs=[Bs As*Bs As*As*Bs As*As*As*Bs];
```

```
rangoMs=rank(MMs);
```

```
Phis = polyvalm(JJ,As);
```

```
KKs = [0 0 0 1]*(inv(MMs))*Phis;
```

```
%Parte Discreta del análisis
```

```
[Ads,Bds,Cds,Dds]=c2dm(As,Bs,C,0,Ts,'zoh');
```

```
MMds=[Bds Ads*Bds Ads*Ads*Bds Ads*Ads*Ads*Bds];
```

```
rangoMds=rank(MMds);
```

```
Phids = polyvalm(JJd,Ads);
```

```
KKds = [0 0 0 1]*(inv(MMds))*Phids;
```


ANEXO 2
LISTADO DEL PROGRAMA "GETK.M"

```

function [KKds]=getk(x3s,x4s);

% getk.m

% Funcion que calcula la matriz de control continua y discreta para el
% sistema del péndulo invertido.

% X3s y x4s son la posición y velocidad angular alrededor de la cual se
% linealiza el sistema.

g=9.8;                %aceleracion debido a la gravedad

conversion=7.06155e-3;

                    %conversion ingles -> MKS=0.453592/16*2.54e-2*g
                    %onzas_fuerza.inches -> Newtons.metros

R=7.38;              %resistencia de armadura
r=0.0648;            %radio de la polea
M=0.92;              %masa del carrito
m=0.060;             %masa de la varilla
md=0.2;              %masa de la polea
Jm=(2.775e-4)*conversion; %momento de inercia del motor
n=19.741;            %reduccion del motor (>1)
K=4.4*conversion;    %constante de armadura del motor
Kb=31.0352e-3;       %constante de la fuerza contraelectromotriz del motor
Bm=(2.5974e-4)*conversion; %friccion viscosa del eje del motor
Bd=0;                %friccion viscosa en el eje de la polea
l=0.785;              %longitud de la varilla

% Matriz de polos continuos
J = [ -6    0    0    0

```

```

    0    -1    0    0
    0    0    -7    0
    0    0    0    -3];

JJ = poly(J);

% Matriz de polos discretos

Ts=1/1000;                %tiempo de muestreo

Jd = [exp(J(1,1)*Ts)    0    0    0
      0    exp(J(2,2)*Ts)    0    0
      0    0    exp(J(3,3)*Ts)    0
      0    0    0    exp(J(4,4)*Ts)];

JJd = poly(Jd);

ajap=R*r*(M+m+md+Jm*n*n/(r*r))/(n*K);
kajap=n*(Kb+R*(Bm+2*Bd/(n*n))/K)/r;
sajap=R*r*(m*l/2)/(K*n);
D=3/2l;

% Alrededor de x3 =x3s y x4=x4s

As=zeros(4); Bs=zeros(4,1); C=[1 0 0 0];
As(1,2)=1; As(3,4)=1;
var1=1/(sajap*D*cos(x3s)*cos(x3s)-ajap);
As(2,2)=kajap*var1;
As(2,3)=(sajap*D*g*(ajap+cos(x3s)*cos(x3s)*(sajap*D-2*ajap))+
sajap*x4s*x4s*cos(x3s)*(ajap-sajap*D*(1+sin(x3s)*sin(x3s))) ) ) *var1*var1;

```

$$As(2,4)=-2*sajap*x4s*\sin(x3s)*var1;$$

$$Bs(2,1)=-var1;$$

$$As(4,2)=-D*As(2,2)*\cos(x3s);$$

$$As(4,3)=var1*var1*(-ajap*g*D*\cos(x3s)*(sajap*D*(1+\sin(x3s)*\sin(x3s))-ajap) \\ +sajap*D*x4s*x4s*(sajap*D*\cos(x3s)*\cos(x3s)-ajap*\cos(2*x3s)));$$

$$As(4,4)=-D*As(2,4)*\cos(x3s);$$

$$Bs(4,1)=D*var1*\cos(x3s);$$

$$MMs=[Bs \ As*Bs \ As*As*Bs \ As*As*As*Bs];$$

$$rangoMs=rank(MMs);$$

$$Phis = polyvalm(JJ,As);$$

$$KKs = [0 \ 0 \ 0 \ 1]*(inv(MMs))*Phis;$$

%Parte Discreta del análisis

$$[Ads,Bds,Cds,Dds]=c2dm(As,Bs,C,0,Ts,'zoh');$$

$$MMds=[Bds \ Ads*Bds \ Ads*Ads*Bds \ Ads*Ads*Ads*Bds];$$

$$rangoMds=rank(MMds);$$

$$Phids = polyvalm(JJd,Ads);$$

$$KKds = [0 \ 0 \ 0 \ 1]*(inv(MMds))*Phids;$$

BIBLIOGRAFIA

- [1] "A fuzzy inference engine in nonlinear analog mode and its application to a fuzzy logic control". Takeshi Yamakawa. IEEE Transactions on Neural Networks, vol. 4 No. 3 pag 496-521. May 1993.
- [2] "Fuzzy if...then rule models and their transformation into one another". László T. Kóczy. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 26 No. 5 pag 621-637. Sept 1996.
- [3] "An introduction to fuzzy control". Dimiter Driankov, Hans Hellendoorn, Michael Reinfrank. Springer. Second revised edition. 1995.
- [4] "What does a probabilistic interpretation of fuzzy sets mean?". Ping Liang and Fengming Song. IEEE Transactions on Fuzzy Systems, vol. 4 No. 2 pag 200-205. May 1996.
- [5] "Development of fuzzy algorithms for servo systems". Y. F. Li and C. C. Lau. IEEE Control Systems Magazine, pag 65-71. Apr 1989.
- [6] "Analysis and design of fuzzy controller and fuzzy observer". Xiao-Jun Ma, Zeng-Qi Sun and Yan-Yan He. IEEE Transactions on Fuzzy Systems, vol. 6, No.1 pag 41-51. Feb 1998.
- [7] "Fuzzy logic toolbox for use with Matlab. User's guide". Version 2. Mathworks. 1999.