



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA**

**DISEÑO DE UN CONTROLADOR DIFUSO PARA COMPENSAR CARGAS DE  
COMUNICACIÓN EN TIEMPO REAL**

**TESIS**

**QUE PARA OPTAR POR EL GRADO DE**

**MAESTRO EN INGENIERÍA**  
**ELÉCTRICA - CONTROL**

**P R E S E N T A**

**JOSÉ ALBERTO APARICIO SANTOS**

**TUTOR:**

**DR. HÉCTOR BENÍTEZ PÉREZ, IIMAS**

**MÉXICO, CIUDAD DE MÉXICO, AGOSTO 2017**

DISEÑO DE UN CONTROLADOR  
DIFUSO PARA COMPENSAR CARGAS  
DE COMUNICACIÓN EN TIEMPO  
REAL

JOSÉ ALBERTO APARICIO SANTOS

AGOSTO 2017

*Dedicado a  
mi padre Simón Aparicio,  
mi madre Basilisa Santos  
y a mis hermanos Simón, Sandra y Leonardo.*

# Agradecimientos

A mi familia, la cual me ha apoyado desde que llegué al mundo.

A mis compañeros, en especial a aquellos que se han convertido en amigos.

A mis profesores, que me enseñaron varias lecciones dentro y fuera del aula, en especial al Dr. Héctor Benítez por dirigir esta tesis y a mis sinodales por sus valiosas observaciones.

Agradezco al proyecto PAPIIT IN104516 por apoyarme con el equipo de cómputo y la licencia del programa MATLAB, de igual manera quiero agradecer al CONACYT y al proyecto CONACYT 176556 por la beca otorgada.

A la Universidad Nacional Autónoma de México, que me apoyó con sus clases, conferencias, bibliotecas y su gente maravillosa. De igual manera agradezco al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas por el espacio y equipo brindado.

A la Universidad Autónoma de la Ciudad de México, a la que le debo grandes enseñanzas y que me dio buenos amigos.

Muchas gracias a todos.

**José Alberto Aparicio Santos**

*”Por mi raza hablará el espíritu”*

*”Nada humano me es ajeno”*

# Índice

<b>Agradecimientos</b>	<b>II</b>
<b>1 Introducción</b>	<b>2</b>
1.1 Objetivos . . . . .	3
1.1.1 Objetivos específicos . . . . .	3
1.2 Metodología . . . . .	4
1.3 Alcances . . . . .	4
1.4 Estructura de la tesis . . . . .	4
<b>2 Antecedentes</b>	<b>6</b>
2.1 Recursos . . . . .	6
2.2 Plataforma virtual . . . . .	7
2.3 Calidad de servicio ( <i>Quality of service</i> -QoS) . . . . .	7
2.4 Nivel de servicio ( <i>service level</i> ) . . . . .	7
2.5 Manejador de recursos (RM) . . . . .	8
2.5.1 Esquemas centralizados . . . . .	8
2.5.2 Esquemas distribuidos . . . . .	9
2.6 Sistemas difusos . . . . .	10
2.6.1 Modelo Takagi-Sugeno . . . . .	11
2.7 Controlabilidad y observabilidad de un sistema difuso . . . . .	13
2.8 Estabilidad de un sistema difuso . . . . .	14
2.8.1 Condiciones de estabilidad de un sistema difuso continuo . . . . .	15
2.8.2 Condiciones de estabilidad de un sistema difuso discreto . . . . .	15
2.9 Desigualdades matriciales lineales ( <i>LMI</i> ) . . . . .	15
2.9.1 Diseño de un controlador difuso estable continuo . . . . .	16
2.9.2 Diseño de un controlador difuso estable discreto . . . . .	17
2.10 Resumen del capítulo . . . . .	17

<b>3</b>	<b>Estado del arte</b>	<b>19</b>
3.1	Trabajos relacionados . . . . .	19
3.1.1	Planificador basado en funciones de utilidad . . . . .	19
3.1.2	Planificador basado en tiempos de cálculos . . . . .	20
3.1.3	Planificador basado en gráficas . . . . .	20
3.1.4	Planificador adaptable basado en realimentación . . . . .	20
3.1.5	Reparto de recurso por búsqueda local subóptima . . . . .	21
3.1.6	Distribución por medio de middleware . . . . .	21
3.1.7	Basado en computación en malla . . . . .	22
3.1.8	Trabajos basados en teoría de juego . . . . .	22
3.1.9	Distribución utilizando lazos de retroalimentación anidados . . . . .	24
3.1.10	Distribución de recursos basados en la calidad de servicio . . . . .	25
3.1.11	Manejador de recursos distribuido . . . . .	25
3.2	Manejador de recursos y aplicaciones . . . . .	26
3.3	Dinámica de ajuste . . . . .	30
3.4	Convergencia . . . . .	31
3.4.1	Convergencia síncrona . . . . .	33
3.4.2	Convergencia asíncrona . . . . .	35
3.5	Resumen del capítulo . . . . .	36
<b>4</b>	<b>Manejador de recursos difuso</b>	<b>37</b>
4.1	Difusión ( <i>Fuzzification</i> ) del modelo síncrono . . . . .	37
4.2	Estabilidad del sistema . . . . .	46
4.3	Resumen del capítulo . . . . .	49
<b>5</b>	<b>Modelo y control del Giroscopio</b>	<b>50</b>
5.1	Giroscopio de 3 grados de libertad . . . . .	50
5.1.1	Modelo no lineal en espacio de estados . . . . .	59
5.2	Variables premisa . . . . .	60
5.3	Observabilidad y controlabilidad . . . . .	62
5.4	Controlador . . . . .	62
5.5	Resumen del capítulo . . . . .	63
<b>6</b>	<b>Resultados</b>	<b>66</b>
6.1	Experimento 1 . . . . .	67
6.2	Experimento 2 . . . . .	71
6.3	Experimento 3 . . . . .	74

<b>7 Conclusiones</b>	<b>77</b>
7.1 Trabajo futuro . . . . .	77

# Lista de figuras

2.1	Manejador de recursos centralizado, donde $app_1, app_2, app_n$ son aplicaciones. $s_1, s_2, s_n$ es el nivel de servicio. $v_1, v_2, v_n$ es la plataforma virtual. $f_1, f_2, f_n$ es el valor de la función de emparejamiento, los subíndices indican el número de aplicación a la que pertenece cierto parámetro. RM es el manejador de recursos. . . . .	9
2.2	Esquema de optimización distribuido, donde $app_1, app_2, app_n$ son aplicaciones. $s_1, s_n$ es el nivel de servicio. $v_1, v_2, v_n$ es la plataforma virtual. $f_1, f_2, f_n$ es el valor de la función de emparejamiento. $\lambda_1, \lambda_2, \lambda_n$ son los niveles de prioridad, los subíndices indican el número de aplicación a la que pertenece el parámetro. RM es el manejador de recursos. . . . .	10
3.1	Relación entre los recursos y $\delta$ , en rojo se encuentra la zona de recursos insuficientes (emparejamiento escaso), en verde la zona de recursos correcta (perfecto emparejamiento) y en azul la zona de recursos abundantes (emparejamiento abundante). . . . .	28
3.2	Visualización de aplicaciones asíncronas, donde RM es la línea del tiempo del manejador de recursos, $app_i$ es la línea del tiempo de una aplicación $i$ . . . . .	35
4.1	Función de membresía triangular. . . . .	42
4.2	Esquema del sistema difuso del manejador de recursos . . . . .	46
4.3	Nivel de servicio de la prueba. . . . .	48
4.4	Plataforma virtual de la prueba. . . . .	48
4.5	Nivel de prioridad de la prueba. . . . .	49
5.1	Partes del giroscopio de 3 grados de libertad [Quanser (2012a)]. . . . .	51
5.2	Marco de coordenadas del giroscopio. . . . .	53



5.3	Esquema del controlador difuso del Giroscopio, los bloque de controlador dependen de los pesos $h_i$ calculados al extremo izquierdo del diagrama, el bloque subsistemas contiene todas la matrices $A_i$ y $B_i$ de la parte consecuente. . . . .	64
5.4	Posición deseada y posición de giroscopio . . . . .	65
5.5	Señales de error del giroscopio. . . . .	65
6.1	Señal deseada del giroscopio en todos los experimentos. . . . .	68
6.2	Señal deseada del Resource Manager experimento 1 . . . . .	68
6.3	Plataformas virtuales del experimento 1 . . . . .	69
6.4	Prioridades del experimento 1 . . . . .	69
6.5	Niveles de servicio del experimento 1 . . . . .	70
6.6	Señal de error del giroscopio experimento 1 . . . . .	70
6.7	Señal deseada del Resource Manager experimento 2 . . . . .	72
6.8	Prioridades del experimento 2 . . . . .	72
6.9	Plataformas virtuales del experimento 2 . . . . .	73
6.10	Señal de error del giroscopio experimento 2 . . . . .	73
6.11	Señal deseada del Resource Manager experimento 3 . . . . .	75
6.12	Prioridades del experimento 3 . . . . .	75
6.13	Plataformas virtuales del experimento 3 . . . . .	76
6.14	Señal de error del giroscopio experimento 3 . . . . .	76

# Lista de tablas

4.1	Combinaciones entre las funciones de membresía $M_{j1}(z_j), M_{j2}(z_j)$ ; $j = 1, 2, 3$ , ( $\cdot$ ) denota la función lógica <i>AND</i> y $w_i, i = 1, 2, 3 \dots 64$ es la cuantificación de cada combinación. . . . .	43
5.1	Características de los motores del giroscopio [Quanser (2012b)] . . .	51
5.2	Combinaciones entre las funciones de membresía para el giroscopio	61

# Capítulo 1

## Introducción

En la computación embebida, el número de aplicaciones que comparten la misma plataforma de ejecución ha ido en aumento debido al incremento en la capacidad de las nuevas plataformas de hardware. Por ello se requieren mejores mecanismos para controlar la tasa de ejecución de cada aplicación.

Un manejador de recursos, también conocido como gestor de recursos (*Resource manager*, RM) se encarga de asignar a los procesos en ejecución recursos de memoria y procesamiento adecuados. La gestión de recursos es la capacidad de minimizar los compromisos de rendimiento de varias cargas de trabajo, junto con las funciones que supervisan el uso de los recursos [Sommerville (2011)].

En las técnicas de reservación de recursos, cada reservación es vista como un procesador virtual (o plataforma) ejecutada a una fracción de la velocidad de procesador físico. Se asume que una aplicación es capaz de ser ejecutada a diferentes niveles de servicio (*service levels*), mayor nivel de servicio implica mayor calidad de servicio (*quality of service*, QoS).

Un sistema sincronizado denota un sistema de computación en el que los resultados del proceso están inmediatamente disponibles con rango estable de certeza para el usuario. No obstante, desde el punto de vista técnico, el proceso en tiempo real denota un sistema de computación en el que la información relativa a una transacción u otro evento se actualiza en los ficheros del sistema simultáneamente con el proceso de la transacción. Es decir, que los ficheros de datos de una aplicación en tiempo real están actualizados en todo momento, por lo tanto la gestión de recursos juega un papel crucial en el desempeño de este tipo de sistema.

Este trabajo se centra en un sistema de control en tiempo real [Saffady (1986)] y sus objetivos principales son demostrar que es posible distribuir recursos de manera subóptima al desempeño del sistema físico y probar experimentalmente que el sistema de control de un sistema electromecánico con el RM da como resultado un ahorro de tiempo. Se construye un modelo difuso de compensación de comunicación entre diversos nodos que usan un medio común en tiempo real, así como un modelo de control difuso para el propio sistema electromecánico. El algoritmo para compensar las cargas de comunicación usado en esta tesis está basado en el artículo [Chasparis et al. (2016)].

## **1.1 Objetivos**

Construir un sistema de compensación de comunicación entre diversos nodos que usan un medio común en tiempo real. El sistema se descompone en dos procesos recursivos:

1. La adaptación de recursos ejecutada RM.
2. La adaptación del nivel de servicio ejecutada por cada aplicación.

Estos procesos llevan al sistema a una distribución de recursos equitativa, lo que significa que todas las aplicaciones tendrán recursos necesarios para su funcionamiento.

Para demostrar la eficacia del RM, se le aplica a un sistema de Giroscopio de 3 grados de libertad [Apkarian and Javid (2013)].

### **1.1.1 Objetivos específicos**

- Como las ecuaciones que describen al RM presentan no linealidades, se recurre a los sistemas difusos para lidiar con ellas, el modelo que se usa es el de Takagi-Sugeno (TS).
- Se diseña un controlador para el sistema electromecánico, el cual toma en cuenta retardos en la red.

- Cada sensor y actuador del sistema electromecánico se toma como una aplicación del compensador de carga, el RM busca que la distribución de recursos sea óptima y no se vea afectado el desempeño del controlador del sistema electromecánico de manera significativa.

Los resultados obtenidos podrían llevar a una predicción o una generalización para sistemas en condiciones similares.

## 1.2 Metodología

Debido a que los datos de interés son observables o medibles de alguna manera se usarán metodologías cuantitativas.

Se busca una relación causa y efecto, en el sistema de control del sistema electromecánico en el cuál se prueba con el RM y sin él. A su vez, se realizan pruebas variando parámetros y algoritmos de control.

## 1.3 Alcances

Para que el sistema pueda implementarse es necesario el conocimiento de algunas constantes o su estimación, las cuales son propias de la aplicación.

El sistema no puede asignar el 100% de los recursos a las aplicaciones, debido a que se requiere cierta cantidad de recursos para la ejecución del sistema operativo y otros programas propios de la computadora.

## 1.4 Estructura de la tesis

La tesis tiene la siguiente estructura. En el capítulo 2 se presentan algunos conceptos básicos relacionados con la asignación de recursos, así como algunas herramientas matemáticas que se utilizan a lo largo de la tesis entre ellas destaca el proceso para difundir un modelo matemático y él como diseñar un controlador estable para dicho modelo por medio de desigualdades matriciales lineales (*Linear Matrix Inequality-LMI*).

El algoritmo del RM utilizado en esta tesis se presenta en el capítulo 3, en el mismo capítulo citan una serie de trabajos relacionados que tratan de resolver el

problema de la distribución de recursos utilizando diversos enfoques.

En el capítulo 4 se lleva a cabo la difusión del modelo del RM y se encuentran las ganancias que garantizan el control y la estabilidad de dicho modelo, de manera similar en el capítulo 5 se llevo a cabo la difusión del modelo de giroscopio y por medio de LMI se encuentran las ganancias que garantizan la estabilidad del sistema.

En el capítulo 6 se muestran los resultados de ambos sistemas acoplados, así como su interpretación. Por último en el capítulo 7 se establecen las conclusiones y se plantean recomendaciones para futuros trabajos.

# Capítulo 2

## Antecedentes

### 2.1 Recursos

En el contexto de sistemas informáticos, un recurso puede incluir entre otras cosas, memoria, primitivas de sincronización, manejadores de archivos, conexiones de red, tokens de seguridad, sesiones de base de datos y tanto servicios locales como distribuidos.

Hay numerosas forma de clasificar los recursos, una de las más simples es ver los recursos como reutilizables o no reutilizables. Los recursos utilizables son típicamente adquiridos por un proveedor de recursos, es usado y después liberado. Una vez que los recursos han sido liberados pueden ser adquiridos y usados nuevamente. Un ejemplo de un recurso reutilizable es la memoria que es asignada y liberada por el sistema operativo. Los recursos reutilizables son la más importante forma de recursos, porque el proveedor de recursos generalmente tiene sólo un número limitado de recursos, por lo tanto reutilizar los recursos tiene sentido. En contraste, los recursos no reutilizables son consumidos, y en consecuencia una vez adquiridos no son liberados, o su liberación es implícita. Un ejemplo de un recursos no reutilizable es el tiempo de procesamiento en una red informática, una vez que el tiempo de procesamiento es adquirido y utilizado, éste se pierde y no puede regresar [Kircher and Jain (2004)].

## 2.2 Plataforma virtual

La plataforma virtual es una abstracción que nace de la abstracción "maquina privada virtual" presentada en [Nesbit et al. (2008)]. Maquina privada virtual, considera todos los recursos físicos (procesadores, ancho de banda de memoria y cachés). Por otro lado abstracción plataforma virtual utilizada en ésta tesis es la propuesta en el proyecto *Adaptivity and Control of Resources in Embedded Systems* (Adaptabilidad y control de recursos en sistemas embebidos - ACTORS), la cual puede ser interpretada como velocidad (relativa a la velocidad completa del procesador) a la que cierta aplicación está siendo ejecutada, ésta abstracción provee un control más fino del tiempo del procesador con lo que permite garantizar los requerimientos de las aplicaciones en tiempo real. Abstractar sólo el tiempo del procesador introduce un grado de incertidumbre porque otros tipos de recursos no fueron modelados. En el proyecto ACTORS para lidiar con ésta incertidumbre se utilizó un bucle de retroalimentación que ajusta el tiempo del procesador conforme la calidad de servicio entregada al usuario [Bini et al. (2011)].

## 2.3 Calidad de servicio (*Quality of service -QoS*)

La QoS es el efecto colectivo del desempeño del servicio, el cual determina el grado de satisfacción del usuario [E.800 (1994)]. La abstracción QoS proporciona un medio para razonar y hacer frente a los requisitos de sincronización de flexible heterogéneos con fluctuación de carga de una manera unificada [Wüst et al. (2005)].

## 2.4 Nivel de servicio (*service level*)

El nivel de servicio mide el desempeño de un sistema, es una medida muy dependiente de la naturaleza del sistema por ejemplo: en una empresa de entregas el desempeño puede medirse en la cantidad de entregas exitosas al día. Por otro lado el nivel de servicio puede depender de lo que un solicitante requiera del sistema, por ejemplo una editorial no reclama la misma velocidad en la reparación de una impresora que un despacho de arquitectos [Bowersox et al. (2002)].

En esta tesis se asume que cada aplicación es capaz de ser ejecutada a diferentes niveles de servicio, donde un mayor nivel de servicio implica una mayor calidad



de servicio (*Quality of service* - QoS) que a su vez deriva en un mayor consumo de recursos, ejemplos de éstos son: la resolución de video ajustable, el monto de datos enviados a través de un canal para mostrar una página de Internet, y la posibilidad de ejecutar un controlador a diferentes tiempos de muestreo [Chasparis et al. (2013)].

## **2.5 Manejador de recursos (RM)**

Un manejador de recursos, se encarga de asignar a los procesos en ejecución recursos de memoria y procesamiento adecuados. La asignación de recursos es la capacidad de minimizar los compromisos de rendimiento de varias cargas de trabajo, junto con las funciones que supervisan el uso de los recursos [Sommerville (2011)].

Existen dos tipos principales de esquemas para un RM; centralizados y distribuidos.

### **2.5.1 Esquemas centralizados**

Un esquema clásico para RM es el esquema centralizado (Figura 2.1), en este el RM se encarga de la asignación de los procesadores virtuales a las aplicaciones, a su vez monitorea el uso de recursos y asigna el nivel de servicio de cada aplicación [Maggio et al. (2013)].

Sin embargo los esquemas centralizados tienen varias debilidades. Primero, la complejidad de los algoritmos usados para implementar el RM crece significativamente con el número de aplicaciones, lo que lo vuelve impráctico, debido a que puede llegar a consumir una mayor cantidad de recursos que los que debe distribuir de manera óptima. Segundo, es difícil implementar una función de costos para estos problemas, el RM debe comparar la calidad de diferentes aplicaciones, pero esta comparación no tiene sentido, pues el concepto de calidad es dependiente de cada aplicación. Por último, para una apropiada asignación de los niveles de servicio el RM debe tener conocimiento de las aplicaciones. Las aplicaciones deben informar al RM sobre el nivel de servicio disponible y los recursos que se consumirán por cada nivel de servicio, esto incrementa significativamente la complejidad de la comunicación [Maggio et al. (2013)].

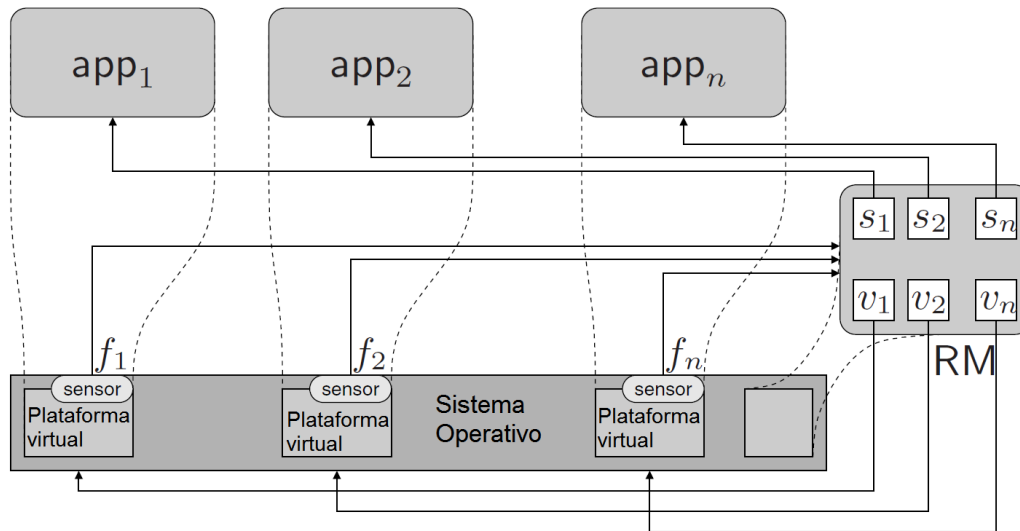


Figura 2.1: Manejador de recursos centralizado, donde  $app_1$ ,  $app_2$ ,  $app_n$  son aplicaciones.  $s_1, s_2, s_n$  es el nivel de servicio.  $v_1, v_2, v_n$  es la plataforma virtual.  $f_1, f_2, f_n$  es el valor de la función de emparejamiento, los subíndices indican el número de aplicación a la que pertenece cierto parámetro. RM es el manejador de recursos.

## 2.5.2 Esquemas distribuidos

Debido a estos problemas, los esquemas distribuidos han ganado mayor atención en estos últimos años, el esquema de la Figura 2.2 es de este tipo, asigna el ancho de banda de procesamiento de las aplicaciones (adaptación de recursos), y el nivel de servicio de cada aplicación es ejecutada por cada aplicación independientemente (adaptación del nivel de servicio). Se proveen condiciones bajo las cuales el esquema recursivo distribuido muestra convergencia a distribuciones equitativas (*fair allocations*). Distribuciones equitativas es donde los recursos están distribuidos de forma tal que las aplicaciones tienen recursos suficientes para ejecutarse sin que alguna aplicación acapare todos recursos o tenga insuficientes. [Subrata et al. (2008)].

El esquema distribuido de la Figura 2.2 asume las siguientes propiedades:

- Complejidad lineal con el número de aplicaciones.

- El RM tiene conocimiento de los detalles de cada aplicación.
- Robustez ante el número y naturaleza de las aplicaciones.

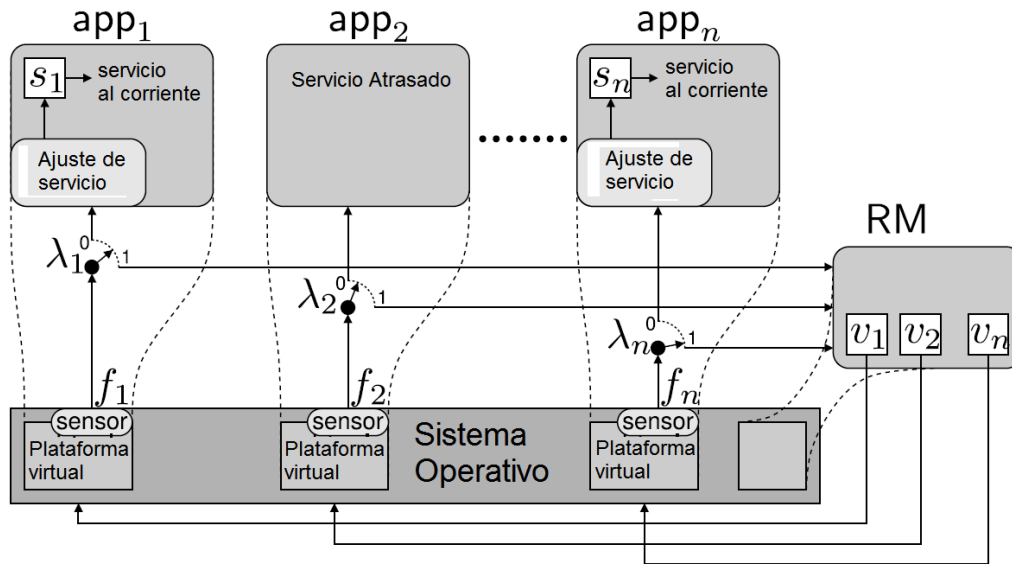


Figura 2.2: Esquema de optimización distribuido, donde  $app_1, app_2, app_n$  son aplicaciones.  $s_1, s_n$  es el nivel de servicio.  $v_1, v_2, v_n$  es la plataforma virtual.  $f_1, f_2, f_n$  es el valor de la función de emparejamiento.  $\lambda_1, \lambda_2, \lambda_n$  son los niveles de prioridad, los subíndices indican el número de aplicación a la que pertenece el parámetro. RM es el manejador de recursos.

## 2.6 Sistemas difusos

Los esquemas de RM están caracterizados por no linealidades, la complejidad de estas obstaculiza el diseño de adecuadas técnicas de control. Las redes neuronales artificiales y sistemas difusos adaptables constituyen una elección confiable para modelar sistemas desconocidos, pues son considerados aproximadores universales. En este sentido, ellos pueden aproximar una función no lineal suave a cualquier exactitud prescrita en una región convexa, se cuenta con suficientes neuronas ocultas y datos de entrenamiento o reglas difusas [Boutalis et al. (2014)].

## 2.6.1 Modelo Takagi-Sugeno

Las técnicas de control difuso representan un medio de recolección de conocimiento humano y habilidad [Fang et al. (2006)]. A pesar de que el método ha sido prácticamente exitoso, se ha probado que es extremadamente difícil desarrollar un análisis general y diseñar una teoría para sistemas de control difusos convencionales. Los sistemas difusos basado en el modelo de Takagi-Sugeno [Tanaka and Wang (2001)] han aparecido recientemente en la literatura, mostrando un gran número de resultados sobre análisis de estabilidad y diseño.

Se usa un modelo difuso Takagi-Sugeno para implementar un RM que se describe en la sección 3.2. El modelo difuso discreto a implementar es el Takagi-Sugeno, el cual está descrito por una serie de reglas **IF – THEN** con la siguiente estructura [Tanaka et al. (1998)]:

**Regla modelo i:**

$$\begin{aligned}
 & \text{IF } z_1 \text{ es } M_{i1} \text{ and } \dots \text{ and } z_p(t) \text{ es } M_{ip} \\
 \text{THEN } & \begin{cases} \nabla x(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \quad (2.1)
 \end{aligned}$$

$M_{ij}$  Funciones de membresía, entregan un valor de membresía en el intervalo de  $[0, 1]$  a cierta variable premisa  $z_i$  asociada.

$r$  número de reglas premisa.

$\nabla x(t)$  representa un operador:

- Para sistemas difusos continuos  $\nabla x(t) = \dot{x}(t)$ .
- Para sistemas difusos discretos  $\nabla x(t) = x(t + 1)$ .

$u(t)$  vector de entrada.

$y(t)$  vector de salida.

- $z_1(t), \dots, z_p(t)$  Variables premisas, la variables premisas son dependientes de los estados y contienen las no linealidades.

El estado se puede reconstruir con las siguientes ecuaciones.

$$\nabla x(t) = \sum_{i=1}^r h_i(z(t)) \{A_i x(t) + B_i u(t)\}, \quad (2.2)$$

$$y(t) = \sum_{i=1}^r h_i(z(t)) C_i x(t), \quad (2.3)$$

donde

$$z(t) = [z_1(t) \ z_2(t) \ \dots \ z_p(t)] \quad (2.4)$$

$$w_i(z(t)) = \prod_{j=1}^p M_{ij}(z_j(t)) \quad (2.5)$$

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{j=1}^r w_j(z(t))} \quad (2.6)$$

$w_i$  Es la cuantificación de cada antecedente  $i$ .

$h_i$  Es el peso de cada antecedente comparado con la suma de todos los antecedentes.

$A_i$  Matriz de estados del subsistema consecuencia  $i$ .

$B_i$  Vector de entrada del subsistema consecuencia  $i$ .

$C_i$  Vector de salida del subsistema consecuencia  $i$ .

**Regla control i:**

$$\begin{aligned} & \text{IF } z_1 \text{ es } M_{i1} \text{ and } \dots \text{ and } z_p(t) \text{ es } M_{ip} \\ & \text{THEN } u(t) = -F_i x(t), \quad i = 1, 2, \dots, r \end{aligned} \quad (2.7)$$

donde la regla de control nominal  $u(t)$  es:

$$u(t) = -\frac{\sum_{i=1}^r w_i(z(t)) F_i x(t)}{\sum_{i=1}^r w_i(z(t))} = -\sum_{i=1}^r h_i(z(t)) F_i x(t) \quad (2.8)$$

Sustituyendo (2.8) en (2.2), el sistema difuso en lazo cerrado puede ser representado como

$$\nabla x(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t)) h_j(z(t)) (A_i - B_i F_j) x(t) \quad (2.9)$$

## 2.7 Controlabilidad y observabilidad de un sistema difuso

Los conceptos de controlabilidad y observabilidad, introducidos por primera vez en [Kalman (1959)], juegan un rol importante en aspectos teóricos y prácticos del control moderno. La controlabilidad de un sistema está cercanamente relacionado a la existencia de soluciones de la realimentación de estados por asignación de ganancias a los valores propios el sistema arbitrario. El concepto de observabilidad está se relaciona a la condición de observar o estimar las variables de estado desde las variables de salida, las cuales son generalmente medibles [Golnaraghi and Kuo (2010)].

Considerar el sistema lineal en espacio de estados (2.10).

$$\dot{x} = Ax + Bu \quad (2.10)$$

donde  $x$  es el vector de estados,  $A$ , la matriz de estado,  $B$  el vector de entrada y  $u$  la entrada del sistema.

La controlabilidad para un sistema lineal en espacio de estados (2.10) equivale a que la matriz  $\Gamma_c$  (ecuación (2.11)) sea de rango completo.

$$\Gamma_c[A, B] \triangleq [B \quad AB \quad A^2B \dots A^{n-1}B] \quad (2.11)$$

De manera similar la observabilidad depende que la matriz  $\Gamma_o$  (ecuación (2.12)) sea de rango completo [Goodwin et al. (2000)].

$$\Gamma_o[A, C] \triangleq \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (2.12)$$

La estabilidad de la salida del sistema difuso está dado por sólo una de la ecuaciones implicación a la vez, dependiendo de sus pesos, la estabilidad nominal del sistema simplemente dependerá de sus subsistemas. En otras palabras si todos los subsistemas son estables el sistema nominal lo será.

El mismo argumento aplica para la controlabilidad y observabilidad del sistema nominal. Si todos los subsistemas nominales son controlables y observables, entonces el sistema completo y observable también, es decir las matrices  $\Gamma_{ci}$  (ecuación 2.13) y  $\Gamma_{oi}$  (ecuación 2.14) deben ser de rango completo para cualquier  $i = 1, 2 \dots r$  [Kachroo (1995)].

$$\Gamma_{ci}[A_i, B_i] \triangleq [B_i \quad A_i B_i \quad A_i^2 B_i \dots A_i^{n-1} B_i] \quad (2.13)$$

$$\Gamma_{oi}[A_i, C_i] \triangleq \begin{bmatrix} C_i \\ C_i A_i \\ \vdots \\ C_i A_i^{n-1} \end{bmatrix} \quad (2.14)$$

## 2.8 Estabilidad de un sistema difuso

**Definición 2.8.1. (Estabilidad cuadrática de un sistema difuso)**[Tanaka and Wang (2001)] *El sistema difuso (2.2) se dice cuadráticamente estabilizable si existe un controlador como en (2.8) tal que el sistema en lazo cerrado (2.9) sea cuadráticamente estabilizable.*

Se define a partir de (2.9)

$$G_{ij} = A_i - B_i F_j \quad (2.15)$$

Entonces sustituyendo (2.15) en (2.9)

$$\nabla x(t) = \sum_{i=1}^r \sum_{j=1}^r h_i(z(t)) h_j(z(t)) G_{ij} x(t) \quad (2.16)$$

Que puede ser reescrita como

$$\nabla x(t) = \sum_{i=1}^r h_i(z(t)) h_i(z(t)) G_{ii} x(t) + 2 \sum_{i < j}^r h_i(z(t)) h_j(z(t)) \left\{ \frac{G_{ij} + G_{ji}}{2} \right\} x(t) \quad (2.17)$$

Las condiciones de estabilidad para un sistema difuso continuo y uno discreto no son iguales, por lo que se presentan por separado.

### 2.8.1 Condiciones de estabilidad de un sistema difuso continuo

Un sistema difuso continuo en lazo cerrado es estable si los términos de (2.17) cumplen con las desigualdades (2.18) y (2.19) [Seidi et al. (2012)].

$$G_{ii}^T P + P G_{ii} < 0 \quad (2.18)$$

$$\left( \frac{G_{ij} + G_{ji}}{2} \right)^T P + P \left( \frac{G_{ij} + G_{ji}}{2} \right) \leq 0 \quad (2.19)$$

En el caso de que  $B_1 = B_2 = \dots B_r$  sólo se debe cumplir la desigualdad (2.18) para garantizar estabilidad asintótica.

### 2.8.2 Condiciones de estabilidad de un sistema difuso discreto

Un sistema difuso discreto en lazo cerrado es estable si los términos de (2.17) cumplen con las desigualdades (2.20) y (2.21).

$$G_{ii}^T P G_{ii} - P < 0 \quad (2.20)$$

$$\left( \frac{G_{ij} + G_{ji}}{2} \right)^T P \left( \frac{G_{ij} + G_{ji}}{2} \right) - P \leq 0 \quad (2.21)$$

En el caso de que  $B_1 = B_2 = \dots B_r$  sólo se debe cumplir la desigualdad (2.20) para garantizar estabilidad asintótica.

El problema de diseño consiste en encontrar la matriz  $P$  y las ganancias  $F_i$ , tales que cumplan las condiciones (2.18)-(2.21) o (2.20)-(2.21) según sea el caso, para encontrar dichos valores se recurre al método por LMI.

## 2.9 Desigualdades matriciales lineales (LMI)

Las LMIs han emergido recientemente como una herramienta útil para resolver un número de problemas de control. La idea básica del método LMI en control es interpretar un problema de control dado como un problema de programación semidefinido.



Las LMIs son muy importantes por muchas razones. Primero, escribiendo un problema en esta forma lleva a una eficiente solución numérica. También, la aproximación es particularmente adecuada a problema con "incertidumbres" de datos y múltiples especificaciones. Finalmente, esta aproximación parece ampliamente aplicable, no sólo en control, si no en otras áreas donde surjan las incertidumbre. [Ghaoui and Niculescu (2000)].

Una LMI tiene la siguiente forma

$$F(x) \triangleq F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.22)$$

donde  $x \in \mathbb{R}^m$  es la variable y las matrices simétricas  $F_i = F_i^T \in \mathbb{R}^{n \times n}$ ,  $i = 0, \dots, m$  son dadas. El símbolo de desigualdad significa que  $F(x)$  es positiva definida. Por supuesto la LMI (2.22) es equivalente a un conjunto de  $n$  desigualdades polinomiales en  $x$ .

También existen LMIs no estrictas, que tiene la forma

$$F(x) \geq 0 \quad (2.23)$$

La LMI estricta (2.22) y la no estricta (2.23) están muy relacionadas [Boyd et al. (1994)].

Las condiciones de estabilidad para un sistema difuso Takagi-Sugeno discreto dependen de encontrar una  $P$  para satisfacer la desigualdad de Lyapunov para todas las reglas. Si el número de reglas  $r$  de un sistema difuso es grande, el problema de encontrar una  $P$  común para satisfacer  $r$  desigualdades no será sencillo, incluso usando LMIs [Wang and Sun (2002)].

En [Tanaka et al. (1998)] se hallan condiciones relajadas de estabilidad con diseño basado en LMI.

### 2.9.1 Diseño de un controlador difuso estable continuo

El diseño del controlador difuso estable para un sistema difuso continuo debe cumplir la siguiente LMI.

$$-XA_i^T - A_iX + M_i^T B_i^T + B_i M_i > 0 \quad (2.24)$$

$$-XA_i^T - A_iX - XA_j^T - A_jX + M_j^T B_i^T + B_i M_j + M_i^T B_j^T + B_j M_i \geq 0 \quad (2.25)$$

$$i < j : h_i \cap h_j \neq 0 \quad (2.26)$$

con  $X = P^{-1}$ ,  $M_i = F_i X$ .

Las condiciones (2.24)-(2.26) son LMIs con respecto a  $X$  y  $M_i$ . Se puede calcular una matriz positiva definida  $X$  y  $M_i$  satisfaciendo las LMIs o determinar que tales  $X$  y  $M_i$  no existen.

## 2.9.2 Diseño de un controlador difuso estable discreto

Para encontrar la matriz  $P > 0$ , y las ganancias  $F_i$  que hagan cumplir la desigualdad se recurre a las siguientes LMIs.

$$\begin{bmatrix} X & XA_i^T - M_i^T B_i^T \\ A_i X - B_i M_i & X \end{bmatrix} > 0 \quad (2.27)$$

$$\begin{bmatrix} X & \left\{ \frac{A_i X + A_j X - B_i M_j - B_j M_i}{2} \right\}^T \\ \left\{ \frac{A_i X + A_j X - B_i M_j - B_j M_i}{2} \right\} & X \end{bmatrix} > 0 \quad (2.28)$$

donde  $X = P^{-1}$ ,  $M_i = F_i X$ .

Tanto en para el caso continuo como el discreto las ganancias  $F_i$  y  $P$  común puede ser obtenida con las ecuaciones (2.29):

$$P = X^{-1}, \quad F_i = M_i X^{-1} \quad (2.29)$$

## 2.10 Resumen del capítulo

En este capítulo se enlistan conceptos básicos utilizados en la distribución de recursos, así como los esquemas centralizado y distribuido para un RM. Se muestran las ecuaciones que describen un modelo difuso Takagi-Sugeno, y se menciona

como evaluar sus propiedades de controlabilidad y observabilidad. De igual manera se indican las condiciones y el método (LMI) para lograr estabilidad en lazo cerrado.

# Capítulo 3

## Estado del arte

### 3.1 Trabajos relacionados

Han existido una extensa cantidad de trabajos acerca de la asignación de recursos equitativa, es de notar que los trabajos difieren en cuanto a abstracciones usadas para la asignación de recursos, las utilizadas en ésta tesis son plataforma virtual, trabajo, actividad y reserva. Sin embargo puede haber otras dependiendo de los recursos que son modelados por dicha abstracción.

A continuación se citan algunos trabajos en los que se trató de resolver el problema de la asignación de recursos.

#### 3.1.1 Planificador basado en funciones de utilidad

Las funciones de utilidad han sido de utilizadas por los economistas durante décadas en los intentos por modelar el comportamiento humano. [Jensen et al. (1985)] fue el primero en aplicar estas funciones en el contexto de sistemas en tiempo real, partiendo de la idea de que la característica distintiva principal de un sistema en tiempo real es el concepto de que la finalización de un proceso o conjunto de procesos tiene un valor para el sistema que puede expresarse como una función del tiempo. Esta noción se describe en términos de un modelo planificación para sistemas operativos en tiempo real y proporciona una herramienta para medir la efectividad de los planificadores de procesos en sistemas de tiempo real más utilizados. Esta medición se lleva a cabo para producir un planificador de procesos en tiempo real de tal manera que sus tiempos de ejecución maximice el valor colec-

tivo del sistema, ya sea en un ambiente multiproceso de memoria compartida o en nodos múltiples de un ambiente de procesos distribuidos.

### **3.1.2 Planificador basado en tiempos de cálculos**

[Mercer et al. (1994)] propone un marco de planificador basado en razones de cálculos expresadas como tiempo de cálculo por duración, proporcionando una forma efectiva para especificar los requerimientos del procesador, manteniendo la flexibilidad de un sistema de tiempo compartido. Diseña un mecanismo con capacidad de reservación que aísla los programas del tiempo y la ejecución características de los otros programas de la misma manera que un sistema de protección de memoria los aísla de acceso de memoria externos. Este marco planificador soporta reservación y control, además introduce una nueva abstracción del núcleo llamada "reserva" (*reserve*), que se encarga de controlar la reservación y mide el uso del procesador por cada programa.

### **3.1.3 Planificador basado en gráficas**

En el artículo [Jones and Rosu (1997)], se presenta un sistema que puede planificar múltiples actividades independientes utilizando una gráfica de planificación precalculada. Una actividad es una abstracción a la cual se le asignan recursos, normalmente cada programa o aplicación en ejecución es asociada con una actividad separada [Jones et al. (1995)]. Las actividades pueden obtener un mínimo garantizado de tasas de ejecución por medio de reservaciones del procesador.

### **3.1.4 Planificador adaptable basado en realimentación**

En [Steere et al. (1999)], se propone asignar a cada subprocesso un porcentaje de ciclos del procesador sobre un periodo de tiempo, y se usa un planificador adaptable basado en realimentación para asignar automáticamente ambos, proporción y periodo. Las aplicaciones con requerimientos conocidos, pueden ignorar el planificador especificando la proporción y el periodo deseados. Como resultado el esquema provee reservaciones a las aplicaciones que la necesiten, y los beneficios de la proporción y el periodo a las que no necesiten reservación. El planificador adaptable usando proporción y periodo ofrecen ciertos beneficios en comparación con los esquemas basados en prioridad adaptable o fija: muy pequeño control de

asignación, baja varianza en el monto de ciclos asignados a un subproceso, y la anulación de inversión de prioridad accidental e inanición, incluyendo defensa contra inversión de ataque de negación de servicio. El planificador asigna la proporción basado en la tasa de progreso, el progreso se mide utilizando un *fill-level* en un buffer acotado, un controlador basado en realimentación dinámicamente ajusta las asignaciones del procesador y el periodo de los subprocesos en el sistema, y un planificador basado en reservación proporcional fundamental. Como resultado el sistema dinámicamente adapta asignaciones para atender recursos recurrentes necesarios para las aplicaciones, sin requerir la participación de un humano experto.

### **3.1.5 Reparto de recurso por búsqueda local subóptima**

[Lee et al. (1999)] considera un problema de reparto múltiple de recursos finitos para satisfacer las necesidades de QoS de multiples aplicaciones a diferente niveles de QoS. Cada aplicación, como video conferencia, necesita múltiples recursos para satisfacer sus requerimientos de QoS. Se evalúan y comparan tres estrategias para resolver este problema. Muestra que la programación dinámica e integración mixta calcula soluciones óptimas para este problema pero presenta tiempos de ejecución muy largos. Se adapta el problema de integración mixta para producir resultados subóptimos con pequeños tiempos de ejecución. Se presenta un algoritmo de enfoque basado en una búsqueda local que está un 5% lejos de la solución óptima pero que es más de dos órdenes más rápido. Esta técnica de búsqueda local resulta ser muy escalable y robusta conforme el número de recursos es requerido por cada aplicación.

### **3.1.6 Distribución por medio de middleware**

Por otro lado, [Li and Nahrstedt (1999)] presenta un marco de control de lógica de intercambio de información entre aplicaciones (*middleware*) para mejorar la efectividad de las decisiones de adaptación de calidad de servicio mediante un control dinámico y la reconfiguración interna de parámetros y funcionalidades de una aplicación multimedia distribuida. El modelo propuesto es modelado por el control de tareas y el modelo de control difuso, basado en resultados rigurosos de la teoría de control, y es verificado por la controlabilidad y adaptabilidad de una aplicación de seguimiento visual distribuido.

### 3.1.7 Basado en computación en malla

[Al-Ali et al. (2004)] describe una arquitectura de manejador de recursos para la computación en malla (*Grid*) QoS. En computación en malla la QoS es vista tres aspectos: red, computación y medios de almacenamiento, de tal forma que se satisfaga el nivel de QoS requerido.

### 3.1.8 Trabajos basados en teoría de juego

[Mahajan et al. (2004)] aplica técnicas de la teoría de juegos para ayudar a formular y analizar soluciones a dos problemas de los sistemas: desalentar la individualidad en redes inalámbricas de múltiples saltos y permitir la cooperación entre ISP en Internet.

En [Roughgarden (2004)] se estudia el problema de optimizar el rendimiento de un sistema compartido por usuarios individuales no cooperativos. Se considera el ajuste de la planificación de pequeños trabajos (*jobs*) en un conjunto compartido de máquinas que poseen funciones de latencia (retardos) que especifican el monto de tiempo necesario para completar un trabajo, dada la carga de máquina. Se mide el rendimiento del sistema por medio de la latencia total, es decir, la suma de los retardos.

Asignando trabajos de acuerdo a los intereses individuales de los usuarios, quienes desean minimizar sólo la latencia de su propia experiencia de trabajo, típicamente resultando en desempeño subóptimo. En muchos sistemas de este tipo hay una mezcla entre trabajos "individualmente controlados" y "centralmente controlados". La congestión debido a los trabajos centralmente controlados influenciará las acciones de los usuarios individuales, y por lo tanto se aspira a contener la degradación del rendimiento del sistema debido al comportamiento individual mediante la planificación de los trabajos centralmente controlados de la mejor manera posible. Se formula este objetivo como un problema de optimización vía los juegos de Stackelberg, juego en los cuales un jugador actúa como líder (es la autoridad centralizada interesada en la optimización del desempeño del sistema) y los demás son los seguidores (los usuarios individuales). El problema es entonces calcular una estrategia para el líder (estrategia Stackelberg) que induce a los seguidores a reaccionar de una manera que aproximadamente minimiza la latencia en el sistema. Con base a esto se desarrolló un algoritmo simple que calcula una es-

trategia provocando una asignación de trabajo con un latencia total no mayor a la de unos tiempos constantes que de la asignación óptima de todos los trabajos; en la ausencia de trabajos centralmente controlados y una estrategia Stackelberg, ningún resultado de este tipo es posible. También se prueba unas fuertes garantías de rendimiento en el caso especial donde cada máquina con funciones de latencia es lineal en la carga de la máquina.

En el artículo [Wüst et al. (2005)], se enfoca en la asignación de recursos aplicado al procesamiento de video, debido a que éste se caracteriza por sus enormes fluctuaciones de contenido dependiente de los tiempos de procesamiento, y una tolerancia limitada para errar en los plazos límite. Se presenta un enfoque cercano al caso promedio que permite una asignación de recursos para una simple tarea de procesamiento de video, basado en un procesamiento asíncrono y escalable, y una adaptación de la calidad de servicio, ésta adaptación equilibra diferentes parámetros de la calidad de servicio que pueden ser sintonizados, basados en experimentos de percepción del usuario: Calidad de imagen, errores en los plazos límites y cambios de calidad. Se modela el problema de equilibrio como un problema de decisión discreto estocástico, y se proponen dos estrategias de solución, basadas en aprendizaje reforzado y en un proceso de decisión de Markov (*Markov decision process* - MDP), ésta última provee de un marco matemático para modelar la toma de decisiones secuencial cuando los resultados son inciertos [Puterman (1994)]. Ambas estrategias se mejoran con una compensación para las fluctuaciones de carga estructural (no estáticas).

[Grosu and Chronopoulos (2005)] presenta un marco de teoría de juegos, se formula el problema de equilibrio de carga en sistemas distribuidos heterogéneos como un juego no cooperativo entre los usuarios, se presenta una estructura de equilibrio de Nash (*Nash equilibrium*), la cuál es un concepto de solución en teoría de juegos donde un jugador no gana si sólo vela por sus propios intereses. Basado en esta estructura se llega un nuevo algoritmo de balanceo de cargas distribuido. La principal ventaja de éste esquema es la estructura distribuida, baja complejidad y optimalidad en la asignación de recursos.

[Subrata et al. (2008)] propone una solución de teoría de juegos para el problema de la asignación de aplicaciones sensibles a la QoS. Modela la asignación de la red basado en la QoS como un juego cooperativo y se presenta la estructura de solución de negociación de Nash (*Nash bargaining Solution*). El algoritmo propuesto es justo para todos los usuarios y representa una solución óptima de Pareto



al QoS objetivo, es decir que se llega a un punto de equilibrio de la asignación de recursos en el cual ninguna aplicación puede mejorar su QoS sin reducir la QoS de otra aplicación. Una ventaja de este esquema es el bajo nivel de sobrecarga y desempeño robusto contra las imprecisiones en la información de predicción del rendimiento.

[Wei et al. (2010)] utiliza la teoría de juegos para resolver un problema de asignación de recursos con restricciones de QoS, en el cual los demandantes de servicios pretenden resolver sofisticados problemas de computación paralela solicitando el uso de recursos a través de una red basada en nube y el costo de cada servicio computacional depende de la cantidad de computo. La teoría de los juegos se utiliza para resolver el problema de la asignación de recursos. Se propone una solución aproximada práctica con los dos pasos siguientes. En primer lugar, cada participante resuelve su problema óptimo independientemente, sin tener en cuenta la multiplexación de las asignaciones de recursos. Se propone un método de programación de enteros binarios para resolver la optimización independiente. En segundo lugar, se diseña un mecanismo evolutivo que cambia las estrategias multiplexadas de las soluciones óptimas iniciales de los diferentes participantes, minimizando sus pérdidas de eficiencia. Los algoritmos en el mecanismo evolutivo tienen en cuenta tanto la optimización como la equidad. Se demuestra que el equilibrio de Nash siempre existe si el juego de asignación de recursos tiene soluciones factibles.

### **3.1.9 Distribución utilizando lazos de retroalimentación anidados**

En [Cucinotta et al. (2010)] donde se considera un conjunto dinámico de aplicaciones en tiempo real usando un conjunto de recursos compartidos. Cada aplicación se puede ejecutar en diferentes modalidades, cada una asociada con un nivel de calidad de servicio. Los recursos, en su naturaleza, tienen diferentes modalidades, cada una con una velocidad y potencia de consumo, y son manejadas por un planificador basado en reservaciones permitiendo una asignación dinámica de la fracción de recursos (ancho de banda) asignado a cada aplicación.

Para lidiar con los cambios dinámicos de la aplicación, el sistema se apoya en una política de recursos adaptable organizada en dos lazos realimentados anidados. El lazo interno opera en el parámetro de planificación para obtener una asignación de

recursos que conoce las restricciones temporales de las aplicaciones. El lazo externo opere en el nivel de servicio de las aplicaciones y en el nivel de potencia de los recursos para lograr un buen intercambio entre la calidad de servicio global y el consumo de energía. Este lazo viene a juego siempre que la carga de trabajo de la aplicación excede las cotas que permiten el lazo interno operar correctamente, o cuando la carga de trabajo decrece abajo de un nivel que permita elecciones más agresivas para la calidad de servicio o un ahorro de energía substancial.

### **3.1.10 Distribución de recursos basados en la calidad de servicio**

[Bini et al. (2011)] dice que las claves para lograr un alto rendimiento en los sistemas embebidos que requieren la ejecución de múltiple aplicaciones en plataformas de múltiples núcleos son: aislamiento entre aplicaciones y asignar apropiados montos de recursos para cada aplicación. El enfoque desarrollado en el proyecto ACTORS provee aislamiento temporal a través de reservaciones de recursos sobre plataformas de múltiples núcleos, adaptando los recursos disponibles basado en el requerimiento de la calidad general.

[Årzén et al. (2011)] se desarrolla un RM que colabora con un nuevo planificador de Linux que proporciona soporte para reservas de servidor de ancho de banda constante con el fin de adaptar las aplicaciones a los cambios en la disponibilidad de recursos y adaptar la asignación de recursos a los cambios en los requerimientos de la aplicación. Se hacen tres pruebas al RM: Calidad de video ajustable, control realimentado y decodificación de video.

### **3.1.11 Manejador de recursos distribuido**

[Maggio et al. (2013)] presenta un RM basado en la teoría de juegos, donde trata de resolver dos problemas de manera desacoplada; la asignación de recursos y el ajuste en la calidad, resolviendo cada uno en el dominio donde naturalmente pertenecen. El desempeño de cada función es medida por su función de emparejamiento. El RM ajusta la distribución de recursos dependiendo del peso dependiente de cada aplicación. La aplicación ajusta su propio nivel de servicio. En [Chasparis et al. (2013)] se propone un marco de RM donde las aplicaciones

y el RM actúan independientemente tratando de maximizar su propia medida de desempeño de acuerdo al proceso de ajuste basado en utilidad. A diferencia de los esquemas de optimización centralizada, el marco propuesto manifiesta propiedades de adaptabilidad y robustez a cambios en el número y la naturaleza de las aplicaciones, esto se logra con poca información disponible tanto por las aplicaciones como por el RM. En otras palabras, el RM se encarga de asignar las plataformas virtuales para las aplicaciones, tratando de alcanzar una distribución equitativa, mientras cada aplicación es responsable de ajustar su propio nivel de servicio, buscando lograr el mejor emparejamiento entre su nivel de servicio y el ancho de banda provisto. El esquema propuesto muestra complejidad lineal en el número de aplicaciones.

[Chasparis et al. (2016)] es una extensión de los artículos [Maggio et al. (2013)] que analiza la convergencia global y asíncrona, y [Chasparis et al. (2013)] el cual presenta una implementación completa del RM en *Linux*. Este artículo propone un esquema distribuido que garantiza una distribución equitativa de recursos, mientras incorpora medición en el desempeño de las aplicaciones y ajuste en el nivel de servicio de las mismas.

El RM que se utiliza en ésta tesis está basado principalmente en los artículos [Chasparis et al. (2013)], [Maggio et al. (2013)] y [Chasparis et al. (2016)], los cuales utilizan las mismas bases y definiciones. A continuación se profundiza en dicho RM.

## 3.2 Manejador de recursos y aplicaciones

El RM asigna recursos a través de un servidor de banda ancha constante (CBS) con periodo  $P_i$  y presupuesto  $Q_i$ , por lo tanto la aplicación  $i$  es asignada a una plataforma virtual con ancho de banda  $v_i$  la cual corresponde a una fracción de la potencia de procesamiento (o velocidad) de un simple CPU [Abeni and Buttazzo (1998)].

La relación entre el presupuesto, el periodo y el ancho de banda está descrita por la ecuación (3.1) [Chasparis et al. (2013)].

$$v_i = \frac{Q_i}{P_i} \tag{3.1}$$

- $v_i$  ancho de banda de la plataforma virtual,

- $Q_i$  presupuesto,
- $P_i$  periodo.

No obstante esta idea se puede aplicar para el caso donde se cuenta con varios procesadores, para facilitar cálculos se puede recurrir a una normalización de las plataformas virtuales (ecuación 3.2) [Chasparis et al. (2013)].

$$\bar{v}_i = \frac{v_i}{\kappa} = \frac{Q_i}{\kappa P_i} \quad (3.2)$$

Donde  $\bar{v}_i$  representa la plataforma virtual de la aplicación  $i$  normalizada y  $\kappa$  es el número de CPUs disponibles.

No todas las plataformas virtuales  $v_i$  son factibles, pues su suma no puede exceder el número  $\kappa$ . Formalmente se define un conjunto de **plataformas virtuales factibles**,  $(v_1, \dots, v_n)$  como [Chasparis et al. (2016)]:

$$\mathbf{v} \doteq \left\{ \mathbf{v} = (v_1, \dots, v_\ell) \in [0, 1]^\ell : \sum_{i=1}^{\ell} v_i \leq \kappa \right\} \quad (3.3)$$

donde  $\ell$  es el número de aplicaciones.

Para que sea posible evaluar el desempeño de una aplicación sensible al tiempo, se necesita una función de desempeño, el RM debe ser capaz de medir esta función en cualquier tiempo  $t \geq 0$ .

La ecuación (3.4), a la cual se le conoce como función de emparejamiento, se encarga de medir el desempeño [Maggio et al. (2013)].

$$f_i(t) \doteq \frac{D_i(t)}{R_i(t)} - 1 \quad (3.4)$$

con

$D_i(t)$  como el plazo flexible (*soft-deadline*) de la aplicación  $i$ , que es la duración en tiempo de su última reservación de CPU.

$R_i(t)$  como el tiempo de respuesta (*job-response*), que es el lapso de tiempo desde el tiempo inicial hasta el tiempo final de un trabajo durante su última reservación.

El valor que toma  $f_i(t)$  le indica al RM si los recursos asignados a la aplicación son suficientes, escasos o demasiados, la relación entre  $f_i(t)$  se describe a continuación:

Para algún pequeño  $\delta > 0$

- Si  $|f_i| \leq \delta$ , la aplicación  $i$  tiene la cantidad correcta de recursos.
- Si  $f_i < -\delta$ , indica que la aplicación  $i$  no tiene suficientes recursos.
- Si  $f_i > \delta$ , la aplicación  $i$  tiene más que suficientes recursos.

Estas zonas de distribución de recursos se pueden observar gráficamente en la Figura 3.1.

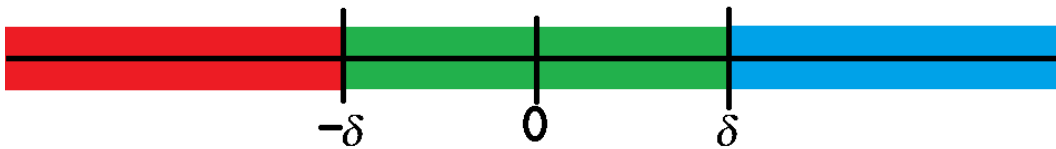


Figura 3.1: Relación entre los recursos y  $\delta$ , en rojo se encuentra la zona de recursos insuficientes (emparejamiento escaso), en verde la zona de recursos correcta (perfecto emparejamiento) y en azul la zona de recursos abundantes (emparejamiento abundante).

En esta tesis  $\delta = 0$ , de esta forma sólo se tienen dos zonas de recursos; suficientes e insuficientes.

La función de emparejamiento depende indirectamente de la plataforma virtual  $v_i$  y el nivel de servicio  $s_i$  de la aplicación  $i$ . Para muchas clases de aplicaciones, se llega a una representación nominal de la función de emparejamiento [Chasparis et al. (2013)].

- Para algunas aplicaciones multimedia,  $D_i$  puede ser considerada constante, mientras que el tiempo de respuesta puede ser definido como  $R_i = C_i/v_i$  donde  $C_i = \alpha_i s_i$  es el tiempo de ejecución por trabajo (en el nivel de servicio  $s_i$ ) y  $v_i$  es la plataforma virtual,  $\alpha_i$  es una constante de proporcionalidad dependiente la aplicación  $i$ .

- En aplicaciones de control,  $R_i = C_i/v_i$ , donde  $C_i$  representa el tiempo nominal de ejecución, mientras que  $D_i$  se considera inversamente proporcional a la frecuencia de muestreo (o nivel de servicio)  $s_i$ , por ejemplo  $D_i = \gamma_i/s_i$ , donde  $\gamma_i$  es una constante de proporcionalidad dependiente la aplicación  $i$  [Chasparis et al. (2013)].

Estás relaciones conducen a la función nominal de emparejamiento (ecuación (3.5)) [Chasparis et al. (2016)].

$$\varphi_i(s_i, v_i) \doteq \beta_i \frac{v_i}{s_i} - 1 \quad (3.5)$$

$\beta_i$  es una constante positiva. La ecuación (3.5) satisface las siguientes propiedades:

Para alguna  $s_i, s'_i \in \mathcal{S}_i$  y  $v_i, v'_i \in \mathbf{v}$ .

1.  $s_i \neq 0 \Rightarrow \varphi_i(s_i, 0) < 0$ , el emparejamiento debe ser escaso si no se asignan recursos.
2.  $s_i \geq s'_i \Rightarrow \varphi_i \leq (s_i, v_i) \leq \varphi_i(s'_i, v_i)$ , esto es si la aplicación  $i$  baja su nivel de servicio, entonces el desempeño no debe disminuir.
3.  $v_i \geq v'_i \Rightarrow \varphi_i(s_i, v_i) \geq \varphi_i(s_i, v'_i)$ , esto es si el ancho de banda de la aplicación  $i$  decrece, entonces el desempeño no debería aumentar.

El RM puede también asignar pesos a las aplicaciones, el peso  $\lambda_i \in (0, 1]$  para representar la importancia que el RM asigna a la aplicación  $i$  cuando se ajusta sus plataforma virtual  $v_i$ .

La distribución equitativa de cada aplicación  $i$ , se define en la ecuación (3.6) [Maggio et al. (2013)]:

$$\Phi_i(\mathbf{s}, \mathbf{v}) \doteq -(1 - \bar{v}_i) \lambda_i [\varphi_i(s_i, v_i)]_- + \bar{v}_i \sum_{j \neq i} \lambda_j [\varphi_j(s_j, v_j)]_- \quad (3.6)$$

La notación  $[x]_-$  para cualquier  $x \in \mathbb{R}$

$$[x]_- = \begin{cases} x, & x \leq 0 \\ 0, & x > 0 \end{cases} \quad (3.7)$$

La función  $\Phi_i$  captura la deficiencia en recursos de la aplicación  $i$  comparada con el resto de las aplicaciones. Cuando la aplicación  $i$  no se está desempeñando correctamente, por ejemplo  $\varphi_i(\mathbf{s}, \mathbf{v}) < 0$ , y sus recursos disponibles  $v_i$  son pequeños,

mientras las demás aplicaciones se desempeñan bien, debemos esperar valores grandes de  $\Phi_i$ , debido a que:

- $-(1 - \bar{v}_i)\lambda_i[\varphi_i(s_i, v_i)]_- > 0$
- $\bar{v}_i \sum_{j \neq i} \lambda_j[\varphi_j(s_j, v_j)]_- = 0$

**Definición 3.2.1.** (*Distribución equitativa*)[Chasparis et al. (2016)] *Para algún perfil de nivel de servicio  $\mathbf{s} \in \mathcal{S}$ , un perfil de plataforma virtual  $\mathbf{v}^* \in \mathcal{V}$  es equitativo o balanceado si  $\Phi_i(\mathbf{s}, \mathbf{v}^*) \equiv 0 \ \forall \ i \in \ell$ .*

donde  $\mathcal{S} = \{\mathbf{s} = (s_1, \dots, s_\ell)\}$ .

Es decir una distribución de plataformas virtuales  $\mathbf{v}^*$  es equitativa para la aplicación  $i$  si y sólo si  $v_i^* \neq 0$ , pues en cero recursos  $\varphi_i(s_i, 0) < 0$  y  $\Phi_i(\mathbf{s}, \mathbf{v}) < 0$ . Por lo tanto una distribución  $\mathbf{v}^*$  es equitativa si se cumple cualquiera de las siguientes [Chasparis et al. (2016)]:

1.  $[\varphi_i(s_i, v_i^*)]_- \equiv 0 \ \forall \ i$
2.  $[\varphi_i(s_i, v_i^*)]_- < 0 \ \forall \ i$  y el radio de recursos  $\bar{v}_i^*/(1 - \bar{v}_i^*)$  coincide con el radio de las funciones de emparejamiento ponderadas.

Como  $[\varphi] \in [-1, 0]$  la segunda propiedad implica que los recursos están balanceados con desempeños negativos, es decir, la aplicación  $i$  no debería desempeñarse demasiado bien comparada con la demás. Informalmente, no puede haber una aplicación que monopolice los recursos en una distribución equitativa.

### 3.3 Dinámica de ajuste

El objetivo central de las distribuciones equitativas se descompone en un par de esquemas recursivos basados en desempeño, ejecutados independientemente por el RM y las aplicaciones, en consecuencia se evita la complejidad de procesamiento y comunicación de la optimización centralizada.

El RM actualiza el ancho de banda  $\bar{v}_i = v_i/\kappa$  normalizado con respecto al número de núcleos  $\kappa$ . El ancho de banda sin usar es  $v_r = \kappa - \sum_{i=1}^n v_i$ , y su versión normalizada  $\bar{v}_r = 1 - \sum_{i=1}^n \bar{v}_i$ .

A instantes de tiempo  $t_k, k = 0, 1, \dots$  el RM asigna recursos de la siguiente manera [Chasparis et al. (2013)]:

1. Mide la función de emparejamiento  $f_i = f_i(t_k)$  para cada  $i \in \ell$ , y procesa  $[f_i(t_k)]_-$
2. Se actualiza el vector de distribución del recurso normalizado  $\bar{v}$ :

$$\bar{v}_i(t_{k+1}) = \Pi_{\bar{v}_i} [\bar{v}_i(t_k) + \varepsilon F_i(t_k)] \quad (3.8)$$

donde  $\varepsilon$  es una pequeña cantidad de recursos, para cada  $i = 1, \dots, n$ , donde  $\bar{v}_i \doteq [0, 1/\kappa]$  y  $F_i(t_k)$  es la medición de la equidad observada, la cual se define como:

$$F_i(t_k) \doteq -(1 - \bar{v}_i(t_k))\lambda_i[f_i(t_k)]_- + \bar{v}_i(t_k) \sum_{j \neq i} \lambda_j[f_j(t_k)]_- \quad (3.9)$$

El ancho de banda no usado se actualiza de acuerdo con  $\bar{v}_r(t_{k+1}) = 1 - \sum_{i=1}^n \bar{v}_i(t_{k+1})$ .

3. Procesa los anchos de banda originales ajustando  $v_i(t_{k+1}) = k\bar{v}_i(t_{k+1})$ .
4. Actualiza el índice del tiempo  $k \leftarrow k + 1$  y se repite.

El RM provee información a cada aplicación  $i$  a través de un señal de observación  $Y_i(t_k)$ ,  $k = 0, 1, \dots$ , que capture su desempeño. Las aplicaciones están diseñadas para ajustar sus niveles de servicio basados en  $Y_i(t_k)$  de la siguiente forma [Chasparis et al. (2013)]:

$$s_i(t_{k+1}) = \Pi_{\mathcal{S}_i} [s_i(t_k) + \varepsilon Y_i(t_k)], \quad i \in \ell \quad (3.10)$$

- $Y_i(t_k)$ ,  $k = 0, 1, \dots$ , señal de observación, captura el desempeño.
- $Y_i(t_k) \equiv f_i(t_k)$ , si  $f_i(t_k) > 0 \Rightarrow s_i(t_k + 1) > s_i(t_k)$

### 3.4 Convergencia

El esquema distribuido propuesto converge a las distribuciones equitativas (Definición 3.2.1), una caracterización de las propiedades de convergencia es provisto en los siguientes casos:

1. Actualizaciones de las aplicaciones síncronas.



2. Actualizaciones de las aplicaciones asíncronas, constituyen una forma de perturbación del comportamiento síncrono nominal.

Propiedades [Chasparis et al. (2016)]:

- Factibilidad (*feasibility*), para un paso suficientemente pequeño  $\varepsilon = \varepsilon(n) > 0 \Rightarrow \bar{v}_i(t_k + 1) \rightarrow \{\mathbf{v}(t_k)\}$  la cual satisface  $\mathbf{v}(t_k) \in \mathcal{V} \forall k = 0, 1, \dots$  mientras  $\mathbf{v}(t_0) \in \mathcal{V}$ .
- Anulación de la inanición (*Starvation avoidance*), garantiza un monto positivo de recursos (al menos  $\varepsilon > 0$ ) para todas las aplicaciones con peso distinto de cero.
- *Balance*, en caso de un sobrecarga del CPU, ninguna aplicación es capaz de monopolizar los recursos.

Se asume que las perturbaciones debido al ruido de medición no están presentes, pues el RM tiene acceso directo al tiempo de respuesta de cada aplicación. Sin embargo, incertidumbres internas de una aplicación pueden resultar en pequeñas desviaciones de su función de emparejamiento nominal. Esto se debe a la pequeña densidad de probabilidad de tales eventos, no se discutirá la robustez con respecto a tales incertidumbres.

**Asignaciones factibles** Para un paso suficientemente pequeño  $\varepsilon = \varepsilon(n) > 0$ , la actualización de la recursividad de las plataformas virtuales proyectadas en (3.8) lleva a una secuencia de plataformas virtuales  $\{\mathbf{v}(t_k)\}$  la cual satisface  $\{\mathbf{v}(t_k)\} \in \mathcal{V} \forall k = 0, 1, \dots$  mientras  $\mathbf{v}(t_0) \in \mathcal{V}$  [Chasparis et al. (2016)].

Se hacen las siguientes definiciones [Chasparis et al. (2016)]:

- $L \doteq \sup_{i \in I, k \in \mathbb{N}} |F_i(t_k)| < \infty$ ,
- $\lambda \doteq \min_{i \in I} \lambda_i > 0$ .

**Anulación de la inanición (*Starvation avoidance*)**.  $\exists \varepsilon^* = \varepsilon^*(n) < 1/(L+1)_k$  con  $\varepsilon^* \rightarrow 0$  conforme  $n \rightarrow \infty$ , tal que para cualquier paso  $\varepsilon \leq \varepsilon^*$ ,  $\inf_{k \in \mathbb{N}} \bar{v}_i(t_k) > \varepsilon \forall i$ .

Es decir, existe un monto positivo de recursos acotado ( $\varepsilon^*$ ) que depende del número de aplicaciones, al ser mayor el número de aplicaciones, el monto de

recursos acotado tiende a ser menor, que garantiza que si el monto de recursos asignados en cada iteración  $\varepsilon$  es menor al monto de recursos acotado  $\varepsilon^*$ , los recursos asignados a las plataformas virtuales van a ser siempre mayores al monto de recursos asignados  $\varepsilon$  para toda aplicación.

**Balance.** Se elige  $0 < \zeta \leq 1/k$  tal que  $\max_{i \in \ell} \{\beta_i k \zeta / \underline{s}_i\} < 0$ . Para cualquier  $\varepsilon = \varepsilon(\zeta) < \zeta/L$ , existe un número de aplicaciones  $n^* = n^*(\zeta)$  tal que, para cualquier conjunto de aplicaciones  $\ell$  con  $|\ell| \geq n^*$  y para cualquier  $i \in \ell$ , se espera lo siguiente [Chasparis et al. (2016)]:

1. si  $\bar{v}_i(t_0) > \zeta$  entonces  $\bar{v}_i(t_k) \leq \zeta$  después de un número finito  $k$ .
2. si  $\bar{v}_i(t_0) \leq \zeta$  entonces  $\bar{v}_i(t_k) \leq \zeta \forall k = 1, 2, \dots$

También, conforme  $\zeta \rightarrow 0$ ,  $n^*(\zeta) \rightarrow \infty$  y  $\zeta n^*(\zeta) \rightarrow c$  para alguna constante positiva  $c$ .

La consecuencia 1 afirma que si se elige  $\bar{v}_i \leq \zeta$  los recursos de cualquier aplicación  $i$  terminarán siendo menores o iguales a  $\zeta$  a pesar de que los recursos asignados inicialmente eran mayores que  $\zeta$ , de manera similar en la consecuencia 2 se observa que si los recursos asignados inicialmente eran menores o iguales que  $\zeta$  estos mantiene la misma relación para todo tiempo finito.

### 3.4.1 Convergencia síncrona

Se consideran las siguientes hipótesis [Chasparis et al. (2016)]:

1.  $\beta_i / \underline{s}_i < 1$  para toda  $i$ , donde  $\underline{s}_i$  es el limite inferior de  $s_i$ .
2. El número de aplicaciones  $n$  debe ser suficientemente grande tal que, exista  $0 < \zeta \leq 1/k$  satisfaciendo las propiedades de la preposición de balance para  $n^*(\zeta) \leq n$ .

La primera hipótesis corresponde al caso donde las aplicaciones son muy demandantes, mientras que la segunda hipótesis corresponde al caso donde los recursos asignados son pequeños comparados al número de aplicaciones.

El comportamiento asintótico de las recursiones (3.8) y (3.10) pueden ser asociados con los puntos límites de las siguiente colección de ecuaciones diferenciales (no lineales) (ODE) [Chasparis et al. (2016)]:

$$\begin{pmatrix} \dot{s}_i(\tau) \\ \dot{v}_i(\tau) \end{pmatrix} = \begin{pmatrix} \varphi_i(s_i(\tau), k\bar{v}_i(\tau)) \\ \Phi_i(\mathbf{s}_i(\tau), k\bar{\mathbf{v}}_i(\tau)) \end{pmatrix} + \mathbf{z}_i(\tau), i \in l \quad (3.11)$$

$\mathbf{z}_i(\tau)$  es la fuerza mínima requerida para conducir a  $\mathbf{s}$  y  $\mathbf{v}$  de regreso a  $\mathcal{S}$  y a  $\mathbf{v}$  respectivamente.

- Ya sea que se cumpla la hipótesis 1 o la hipótesis 2 se aplica la ODE (3.11), que exhibe puntos estacionario  $(\mathbf{s}^*, \bar{\mathbf{v}}^*)$ , los cuales satisfacen:

$$s_i^* = \underline{s}_i, \text{ and } \begin{cases} \Phi_i(\mathbf{s}^*, \kappa \bar{\mathbf{v}}^*) = 0, & \text{or} \\ \Phi_i(\mathbf{s}^*, \kappa \bar{\mathbf{v}}^*) > 0, & \bar{v}_i = 1/\kappa \end{cases} \forall i \quad (3.12)$$

La ecuación (3.12) implica que los puntos estacionarios de las recursiones están definidos por el valor mínimo del nivel de servicio y cuando las distribuciones sean equitativas, para toda aplicación.

- Si se cumple (i)  $\beta_i k / s_i \rightarrow 0 \forall i$  o (ii)  $n \rightarrow \infty$ , entonces:

(a) Cualquier punto estacionario de la ODE (3.11) satisface:

$$s_i^* = \underline{s}_i, \quad \bar{v}_i^* \rightarrow \min \left\{ \frac{1}{k}, \frac{\lambda_i}{\sum_j \lambda_j} \right\}, \quad \forall i \in \ell \quad (3.13)$$

La ecuación (3.13) representa que el punto estacionario del nivel de servicio se encuentra en su valor mínimo, mientras que el punto estacionario de la plataforma virtual también se puede encontrar en su valor mínimo, el cual está ligado al número de núcleos y al peso asignado a cada aplicación.

- (b)  $(s_i(t_k), \bar{v}_i(t_k)) \rightarrow (s_i^*, \bar{v}_i^*)$  conforme  $k \rightarrow \infty$  y  $\varepsilon \rightarrow 0$  esto denota que tanto la plataforma virtual como el nivel de servicio convergerán a sus puntos estacionarios conforme las iteraciones  $k$  vayan aumentando a la vez que los recursos mínimos asignados  $\varepsilon$  tienden a cero, debido a que al llegar a los puntos estacionarios no habrá necesidad de asignar más recursos.

Sin embargo, la plataforma virtual,  $v_i$ , indirectamente determina la velocidad de ejecución de la aplicación  $i$ . Por lo tanto, la actualización de la tasa de servicio  $s_i$  varía con respecto al tiempo. Es importante mencionar que bajo actualizaciones asíncronas, la asignación asintótica de plataformas virtuales no tiene porque ser

necesariamente equitativa para todas las aplicaciones.

### 3.4.2 Convergencia asíncrona

En la Figura 2.2 se muestra un esquema donde las actualizaciones del RM y una aplicación ( $app_i$ ) se llevan a cabo en diferentes momentos.

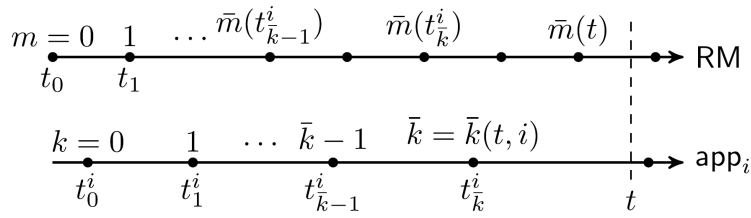


Figura 3.2: Visualización de aplicaciones asíncronas, donde RM es la línea del tiempo del manejador de recursos,  $app_i$  es la línea del tiempo de una aplicación  $i$ .

$t$  el tiempo de ejecución actual;

$t_k^i$  El instante de actualización de la aplicación  $i$ ;

$t_m$  el instante de actualización del RM;

- $\bar{k}(t, i) \doteq \{k \in \mathbb{N} : t_k^i \leq t < t_{k+1}^i\}$  es el número de veces que se ha actualizado la aplicación.
- $\bar{m}(t) \doteq \{k \in \mathbb{N} : t_k \leq t < t_{k+1}\}$  es el número de veces que se ha actualizado el RM.

La actualización de la recursividad de la aplicación  $i$  está dada por [Chasparis et al. (2016)]:

$$s_i(t_{k+1}^i) = \Pi_{\mathcal{S}_i}[s_i(t_k^i) + \varepsilon Y_i(t_{\bar{m}(t_k^i)})] \quad (3.14)$$

donde

$$Y_i(t_k^i) = N_i(k) \cdot Y_i'(t_k^i), \quad (3.15)$$

para cada  $i \in \ell$ , donde  $Y_i'(\cdot) \doteq f_i(\cdot)$ .

$N_i(k) \doteq \bar{m}(t_{k+1}^i) - \bar{m}(t_k^i)$  es el número de veces que el RM se ha actualizado dentro de  $[t_k^i, t_{k+1}^i]$ .

### **3.5 Resumen del capítulo**

Se citan diversos trabajos que tratan de resolver el problema de la distribución de recursos utilizando diferentes enfoques y abstracciones. Se profundiza en el RM descrito en [Maggio et al. (2013)], [Chasparis et al. (2013)] y [Chasparis et al. (2016)], así como en sus propiedades, debido a que es la base del RM que se utiliza en esta tesis.

# Capítulo 4

## Manejador de recursos difuso

### 4.1 Difusión (*Fuzzification*) del modelo síncrono

En el capítulo anterior se describe brevemente el modelo Takagi-Sugeno el cuál es aplicado al RM, se usan sistemas difusos tanto en el RM como en el sistema electromecánico debido a que su implementación es sencilla, son capaces de lidiar con las no linealidades al dividir el sistema en una serie de sistemas lineales más sencillos de controlar, aparte de ofrecer buenos resultados, esto se debe a que se disminuye la transición de estados fundamentales en el entorno físico.

Para realizar esta difusión se tomará el caso de 3 aplicaciones, las cuales serán sensor, controlador y actuador ya que son los segmentos principales de un sistema de control.

Las ecuaciones que describen el comportamiento del RM son las siguiente:

$$\bar{v}_i(t_{k+1}) = \Pi_{\bar{v}_i} [\bar{v}_i(t_k) + \epsilon F_i(t_k)] \quad (4.1)$$

$$s_i(t_{k+1}) = \Pi_{\mathcal{S}_i} [s_i(t_k) + \epsilon Y_i(t_{\bar{m}(t_k^i)})] \quad (4.2)$$

$$F_i(t_k) \doteq -(1 - \bar{v}_i(t_k)) \lambda_i [f_i(t_k)]_- + \bar{v}_i(t_k) \sum_{j \neq i} \lambda_j [f_j(t_k)]_- \quad (4.3)$$

$$Y_i(t_k) \equiv f_i(t_k) = \beta_i \frac{v_i(t_k)}{s_i(t_k)} - 1 \quad (4.4)$$

con  $t_k = \tau, i = 1, 2, 3 \dots$

Donde

$\tau$  es el tiempo.

$i$  es el número de aplicación.

$\bar{v}_i$  es el tamaño de la plataforma virtual asignada a la aplicación  $i$  que va de  $[0, 1]$ .

$s_i$  es el nivel de servicio de la aplicación  $i$ .

$F_i$  se le conoce como medida de equidad nominal (*nominal fairness measure*) que indica en qué medida la aplicación  $i$  está siendo afectada por las otras aplicaciones.

$f_i(t_k)$  Es la función de emparejamiento, la cual indicará si la aplicación está recibiendo suficientes o insuficientes recursos, el operador  $[\ ]_-$  limita su rango a  $[-1, 0]$ .

$\beta_i$  es una constante positiva que depende de la aplicación.

$\varepsilon$  es el paso (*step size*), representa una pequeña cantidad de recursos que se asigna o se quita en cada iteración.

$\lambda_i$  es el nivel de prioridad, está limitado de  $(0, 1]$ , si una aplicación tiene una prioridad mayor que las demás aplicaciones, entonces tendrá mayor prioridad en la asignación de recursos en comparación con las otras aplicaciones.

Además, para el modelo difuso se tomará  $\lambda_i$  como un estado adicional dependiente de  $\tau$ , y afectado por una entrada de control  $u_i$  provenientes del RM, esto con el fin tener propiedades de controlabilidad y observabilidad, las cuales se analizarán más adelante.

Todos los estados dependen de  $\tau$ , para simplificar la lectura de las ecuaciones esto se tomara de forma implícita, es decir  $\hat{v}_i(\tau) = \hat{v}_i$ ,  $\hat{s}_i(\tau) = \hat{s}_i$  y  $\lambda_i(\tau) = \lambda_i$ .

Definiendo la actualización de la plataforma virtual para la aplicación 1 ( $i = 1$ ), sustituyendo (4.3) en (4.1) se tiene.

$$\dot{\hat{v}}_1 = \bar{v}_1 + \varepsilon \{ -(1 - \bar{v}_1) \lambda_1 [f_1]_- + \bar{v}_1 (\lambda_2 [f_2]_- + \lambda_3 [f_3]_-) \} \quad (4.5)$$

De manera similar se obtiene la actualización de las plataformas 2 y 3.

La actualización del nivel de servicio  $i$ , con  $i = 1, 2, 3$ , se obtiene sustituyendo (4.4) en (4.2).

$$\dot{s}_i = s_i + \varepsilon \left\{ \beta_i \frac{\kappa \bar{v}_i}{s_i} - 1 \right\} \quad (4.6)$$

Es importante recordar que la actualización del nivel de servicio se lleva a cabo dentro de la aplicación, y el RM sólo recibe el valor de la función de emparejamiento, debido a esto la ecuación 4.6 no será sometida a una difusión.

Las ecuaciones a difundir son:

$$\dot{v}_1 = \bar{v}_1 + \varepsilon \{ -(1 - \bar{v}_1) \lambda_1 [f_1]_- + \bar{v}_1 (\lambda_2 [f_2]_- + \lambda_3 [f_3]_-) \} \quad (4.7)$$

$$\dot{v}_2 = \bar{v}_2 + \varepsilon \{ -(1 - \bar{v}_2) \lambda_2 [f_2]_- + \bar{v}_2 (\lambda_1 [f_1]_- + \lambda_3 [f_3]_-) \} \quad (4.8)$$

$$\dot{v}_3 = \bar{v}_3 + \varepsilon \{ -(1 - \bar{v}_3) \lambda_3 [f_3]_- + \bar{v}_3 (\lambda_1 [f_1]_- + \lambda_2 [f_2]_-) \} \quad (4.9)$$

$$\dot{\lambda}_1 = \lambda_1 + u_1 \quad (4.10)$$

$$\dot{\lambda}_2 = \lambda_2 + u_2 \quad (4.11)$$

$$\dot{\lambda}_3 = \lambda_3 + u_3 \quad (4.12)$$

Reescribiendo las ecuaciones (4.7)-(4.12).

$$\dot{v}_1 = \bar{v}_1 \{ 1 + \varepsilon (\lambda_1 [f_1]_- + \lambda_2 [f_2]_- + \lambda_3 [f_3]_-) \} - \varepsilon \lambda_1 [f_1]_- \quad (4.13)$$

$$\dot{v}_2 = \bar{v}_2 \{ 1 + \varepsilon (\lambda_1 [f_1]_- + \lambda_2 [f_2]_- + \lambda_3 [f_3]_-) \} - \varepsilon \lambda_2 [f_2]_- \quad (4.14)$$

$$\dot{v}_3 = \bar{v}_3 \{ 1 + \varepsilon (\lambda_1 [f_1]_- + \lambda_2 [f_2]_- + \lambda_3 [f_3]_-) \} - \varepsilon \lambda_3 [f_3]_- \quad (4.15)$$

$$\dot{\lambda}_1 = \lambda_1 + u_1 \quad (4.16)$$

$$\dot{\lambda}_2 = \lambda_2 + u_2 \quad (4.17)$$

$$\dot{\lambda}_3 = \lambda_3 + u_3 \quad (4.18)$$

De esta forma se definen 6 variables premisa.



$$z_1 = 1 + \varepsilon(\lambda_1[f_1]_- + \lambda_2[f_2]_- + \lambda_3[f_3]_-) \quad (4.19)$$

$$z_2 = -\varepsilon[f_1]_- \quad (4.20)$$

$$z_3 = 1 + \varepsilon(\lambda_1[f_1]_- + \lambda_2[f_2]_- + \lambda_3[f_3]_-) \quad (4.21)$$

$$z_4 = -\varepsilon[f_2]_- \quad (4.22)$$

$$z_5 = 1 + \varepsilon(\lambda_1[f_1]_- + \lambda_2[f_2]_- + \lambda_3[f_3]_-) \quad (4.23)$$

$$z_6 = -\varepsilon[f_3]_- \quad (4.24)$$

$$(4.25)$$

donde  $\lambda_1$ ,  $\lambda_2$  y  $\lambda_3$  son distintos de 0.

Se escribe el sistema en forma de espacio de estados.

$$\begin{bmatrix} \dot{\bar{v}}_1 \\ \dot{\bar{v}}_2 \\ \dot{\bar{v}}_3 \\ \dot{\lambda}_1 \\ \dot{\lambda}_2 \\ \dot{\lambda}_3 \end{bmatrix} = \begin{bmatrix} z_1 & 0 & 0 & z_2 & 0 & 0 \\ 0 & z_3 & 0 & 0 & z_4 & 0 \\ 0 & 0 & z_5 & 0 & 0 & z_6 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \bar{v}_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (4.26)$$

La matriz de la ecuación (4.26) donde se encuentran las variable premisa es la base para obtener las matrices de estados  $A_i$  del subsistema consecuencia  $i$ , en este caso el vector de entrada  $B$  es constante, por lo que en cada consecuencia su valor será el mismo.

Todos los estados son conocidos por el RM, así que la matriz  $C$  es una matriz identidad de dimensión  $6 \times 6$ .

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \bar{v}_3 \\ \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \quad (4.27)$$

Conociendo las cotas de cada estado, se calculan los máximos y mínimos de las variables premisas

$$z_{1min} = 1 - 3\epsilon \qquad z_{1Max} = 1 \qquad (4.28)$$

$$z_{2min} = 0.09 \qquad z_{2Max} = \epsilon \qquad (4.29)$$

$$z_{3min} = 1 - 3\epsilon \qquad z_{3Max} = 1 \qquad (4.30)$$

$$z_{4min} = 0.09 \qquad z_{4Max} = \epsilon \qquad (4.31)$$

$$z_{5min} = 1 - 3\epsilon \qquad z_{5Max} = 1 \qquad (4.32)$$

$$z_{6min} = 0.09 \qquad z_{6Max} = \epsilon \qquad (4.33)$$

$$(4.34)$$

Las variables subrayadas indican el límite inferior variable, es el caso de  $\underline{\lambda}_1$ ,  $\underline{\lambda}_2$  y  $\underline{\lambda}_3$  no pueden ser cero, pero sí cercano.

Sabiendo el valor máximo y mínimo de las variables premisas, podemos definir funciones de membresía tipo triangular que representen el valor de membresía de cada variable premisa, las funciones de membresía deben cumplir las siguientes ecuaciones:

$$z_j = M_{j1}z_{imax} + M_{j2}z_{jmin} \qquad (4.35)$$

$$1 = M_{j1} + M_{j2} \qquad (4.36)$$

donde  $j = 1, 2, 3 \dots 6$ .

De manera general las seis funciones de membresía estarán definidas de la siguiente manera, la figura 4.1 describe la forma que tendrán las funciones:

$$M_{j1}(z_j) = 1 - \frac{z_j - z_{jmax}}{z_{jmin} - z_{jmax}} \qquad (4.37)$$

$$M_{j2}(z_j) = \frac{z_j - z_{jmax}}{z_{jmin} - z_{jmax}} \qquad (4.38)$$

Los enunciados asociados a las funciones de membresía representa los siguientes en sus valores máximos:

- $M_{11}$  Recursos desbalanceados.
- $M_{12}$  Recursos balanceados.
- $M_{21}$  Recursos suficientes APP1.
- $M_{22}$  Recursos insuficientes APP1
- $M_{31}$  Recursos desbalanceados.
- $M_{32}$  Recursos balanceados.
- $M_{41}$  Recursos suficientes APP2.
- $M_{42}$  Recursos insuficientes APP2
- $M_{51}$  Recursos desbalanceados.
- $M_{52}$  Recursos balanceados.
- $M_{61}$  Recursos suficientes APP3.
- $M_{62}$  Recursos insuficientes APP3.

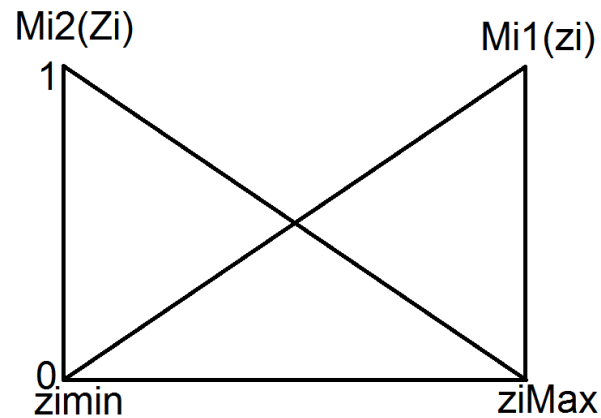


Figura 4.1: Función de membresía triangular.

A cada función de pertenencia se le asigna un enunciado, al ser doce variables premisa las posibles combinaciones serán  $2^6$ , en la Tabla 4.1.

$w_i(z)$	IF	THEN $A_i + Bu$
$w_1(z)$	$M_{11}(z_1) \cdot M_{21}(z_2) \cdot M_{31}(z_3) \cdot M_{41}(z_4) \cdot M_{51}(z_5) \cdot M_{61}(z_6)$	$A_1x + Bu$
$w_2(z)$	$M_{11}(z_1) \cdot M_{21}(z_2) \cdot M_{31}(z_3) \cdot M_{41}(z_4) \cdot M_{51}(z_5) \cdot M_{62}(z_6)$	$A_2x + Bu$
$w_3(z)$	$M_{11}(z_1) \cdot M_{21}(z_2) \cdot M_{31}(z_3) \cdot M_{41}(z_4) \cdot M_{52}(z_5) \cdot M_{61}(z_6)$	$A_3x + Bu$
$\vdots$	$\vdots$	$\vdots$
$w_{63}(z)$	$M_{12}(z_1) \cdot M_{22}(z_2) \cdot M_{32}(z_3) \cdot M_{42}(z_4) \cdot M_{52}(z_5) \cdot M_{61}(z_6)$	$A_{63}x + Bu$
$w_{64}(z)$	$M_{12}(z_1) \cdot M_{22}(z_2) \cdot M_{32}(z_3) \cdot M_{42}(z_4) \cdot M_{52}(z_5) \cdot M_{62}(z_6)$	$A_{64}x + Bu$

Tabla 4.1: Combinaciones entre las funciones de membresía  $M_{j1}(z_j), M_{j2}(z_j); j = 1, 2, 3$ , ( $\cdot$ ) denota la función lógica *AND* y  $w_i, i = 1, 2, 3 \dots 64$  es la cuantificación de cada combinación.

Donde las matrices consecuencia son:

$$A_1 = \begin{bmatrix} 0.7 & 0 & 0 & 0.09 & 0 & 0 \\ 0 & 0.7 & 0 & 0 & 0.09 & 0 \\ 0 & 0 & 0.7 & 0 & 0 & 0.09 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0.7 & 0 & 0 & 0.09 & 0 & 0 \\ 0 & 0.7 & 0 & 0 & 0.09 & 0 \\ 0 & 0 & 0.7 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

⋮

$$A_{63} = \begin{bmatrix} 1 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.09 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{64} = \begin{bmatrix} 1 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Conociendo la cuantificación de cada combinación, se puede obtener el centroide con (2.6), el cual es el peso que tiene cada consecuencia en el resultado general.

$$\begin{aligned}
h_1(z(t)) &= \frac{w_1(z(t))}{\sum_{j=1}^{64} w_j(z(t))} \\
h_2(z(t)) &= \frac{w_2(z(t))}{\sum_{j=1}^{64} w_j(z(t))} \\
h_3(z(t)) &= \frac{w_3(z(t))}{\sum_{j=1}^{64} w_j(z(t))} \\
&\vdots \\
h_{63}(z(t)) &= \frac{w_{63}(z(t))}{\sum_{j=1}^{64} w_j(z(t))} \\
h_{64}(z(t)) &= \frac{w_{64}(z(t))}{\sum_{j=1}^{64} w_j(z(t))}
\end{aligned}$$

La defuzzificación se logra con la ecuación (4.39).

$$x(t+1) = \sum_{i=1}^{64} h_i(z(t)) \{A_i x(t) + Bu(t)\}, \quad (4.39)$$

$$y(t) = \sum_{i=1}^{64} h_i(z(t)) Cx(t), \quad (4.40)$$

Cada consecuencia  $A_i x + Bu$  es controlable, debido a que la matriz  $\mathfrak{C}$  (4.41) es de rango completo para  $i = 1, 2, 3, \dots, 8$ .

$$\mathfrak{C}_i = [B \quad A_i B \quad A_i^2 B \dots A_i^{n-1} B] \quad (4.41)$$

De manera similar, el sistema difuso es observable pues la matriz  $\mathfrak{D}$  (4.42) es de rango completo para  $i = 1, 2, 3, \dots, 8$ .

$$\mathfrak{D}_i = \begin{bmatrix} C \\ CA_i \\ CA_i^2 \\ \vdots \\ CA_i^{n-1} \end{bmatrix} \quad (4.42)$$

El sistema no pierde ninguna de sus propiedades al volverlo difuso, por lo que las propiedades citadas en el capítulo 2 siguen siendo válidas.

El ciclo del sistema es el siguiente:

1. Por medio de los valores anteriores de  $v_1, v_2, v_3, \lambda_1, \lambda_2, \lambda_3$  y el valor de las funciones de emparejamiento  $[f_1]_-, [f_2]_-, [f_3]_-$  se calcula la siguiente plataforma virtual.
2. Con el valor de la nueva plataforma virtual se calcula el tiempo de ejecución.
3. La primera aplicación se ejecuta en el tiempo de ejecución calculado y esta calcula el nuevo nivel de servicio por medio de la ecuación 4.6, después se ejecuta la segunda y la tercera aplicación en los tiempos de ejecución calculados, y cada una de ellas calcula su propio nivel de servicio y entregan el nivel de emparejamiento.
4. El ciclo se repite con los nuevos valores de los estados.

Un esquema del RM difuso se presenta en la Figura 4.2.

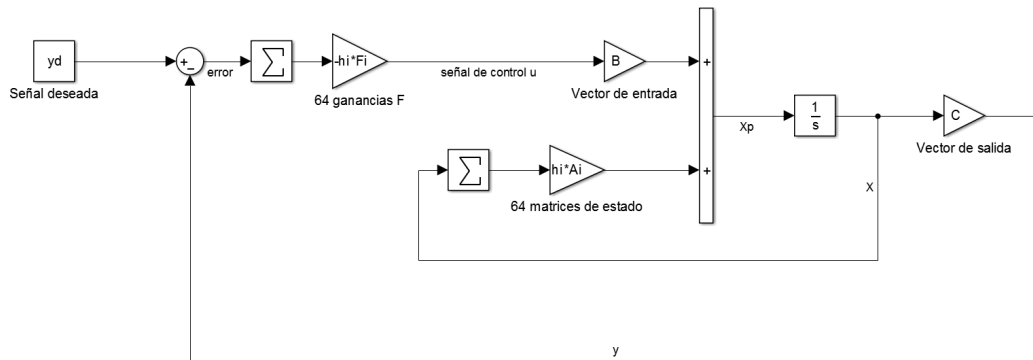


Figura 4.2: Esquema del sistema difuso del manejador de recursos

## 4.2 Estabilidad del sistema

Al ser un sistema difuso discreto con  $B_1 = B_2 = \dots = B_r$  sólo se debe encontrar una matriz  $P$  común para todos los subsistemas y las ganancias  $F_i$  que cumplan

con la desigualdad (2.20) para garantizar estabilidad asintótica como se menciona en el apartado 2.9.2, para encontrar las ganancias  $F_i$  y la matriz  $P$  se recurre a programas matemáticos como *MATLAB* [Duan and Yu (2013)].

$$P = \begin{bmatrix} 1539.4 & 0 & 0 & 189.7 & 0 & 0 \\ 0 & 1539.4 & 0 & 0 & 189.7 & 0 \\ 0 & 0 & 1539.4 & 0 & 0 & 189.7 \\ 189.7 & 0 & 0 & 286 & 0 & 0 \\ 0 & 189.7 & 0 & 0 & 286 & 0 \\ 0 & 0 & 189.7 & 0 & 0 & 286 \end{bmatrix} \quad (4.43)$$

Se realizó una simulación utilizando del controlador diseñado, usando los siguientes parámetros.

$$\beta_1 = 0.004$$

$$\beta_2 = 0.003$$

$$\beta_3 = 0.005$$

$$\varepsilon = 0.1$$

y condiciones iniciales

$$\bar{v}_1 = 0.4$$

$$\bar{v}_2 = 0.3$$

$$\bar{v}_3 = 0.5$$

$$\lambda_1 = 0.3$$

$$\lambda_2 = 0.2$$

$$\lambda_3 = 0.1$$

$$\bar{s}_1 = 102$$

$$\bar{s}_2 = 103$$

$$\bar{s}_3 = 105$$

Las plataformas virtuales deseadas son  $\bar{v}_1 = 0.2$ ,  $\bar{v}_2 = 0.5$ ,  $\bar{v}_3 = 0.3$ . Los resultados arrojados se muestran en las figuras 4.3, 4.4 y 4.5.



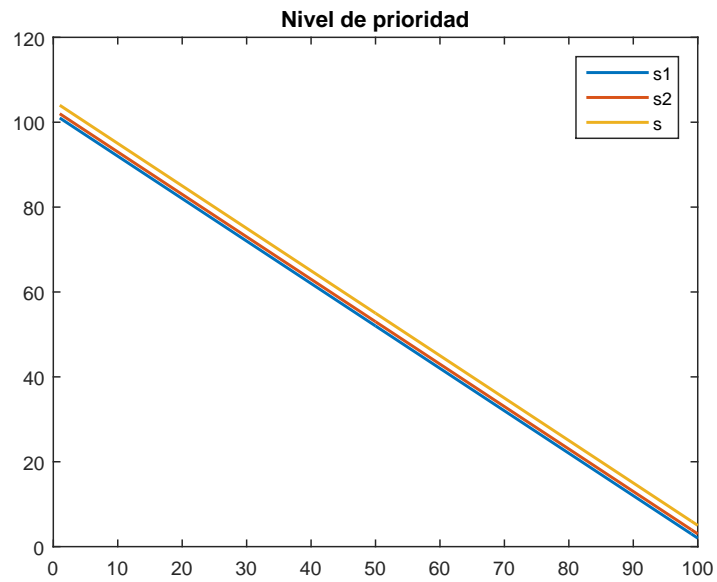


Figura 4.3: Nivel de servicio de la prueba.

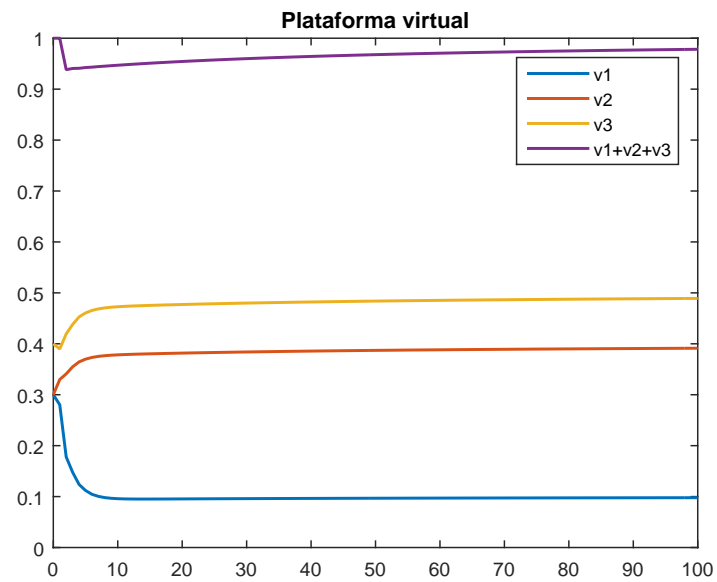


Figura 4.4: Plataforma virtual de la prueba.

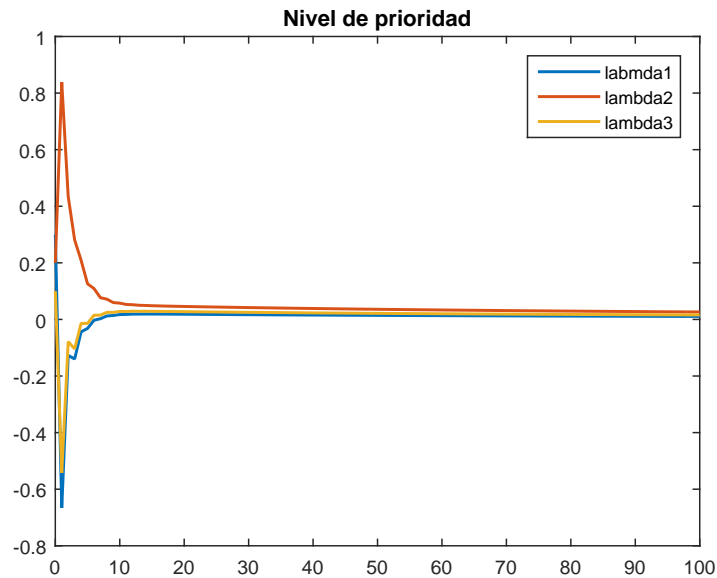


Figura 4.5: Nivel de prioridad de la prueba.

### 4.3 Resumen del capítulo

Con base a la teoría citada en el capítulo 1 y 2 se desarrolla el sistema difuso para el RM, gracias a esto se ha logrado lidiar con las no linealidades reconstruyendo el sistema no lineal con ayuda de sólo 8 subsistemas lineales en la parte de la distribución de plataformas virtuales  $v_i$  sin perder sus propiedades de estabilidad. Se tomó la prioridad como señal de control, de esta forma se puede controlar a las plataformas virtuales e indirectamente al nivel de servicio.

El cálculo del nivel de servicio se lleva a cabo dentro de cada aplicación, la cual sólo entregará el nivel de equidad al RM, el RM no tiene conocimiento del nivel de servicio, pues este es dependiente de parámetros propios de cada aplicación.

# Capítulo 5

## Modelo y control del Giroscopio

Para probar experimentalmente que el sistema de control de un sistema electromecánico con el RM da como resultado un ahorro de tiempo. Se eligió como sistema electromecánico un giroscopio. Se asume que una aplicación es capaz de ejecutarse a diferentes niveles de servicio, donde un alto nivel de servicio implica una alta calidad de servicio, como consecuencia se tiene un alto monto de consumo de recursos. Ejemplos de esto es la posibilidad de ejecutar un controlador a diferentes tasas de muestreo. En este caso en particular se toma cada parte del sistema del giroscopio como una aplicación, las cuales serán la siguientes:

- Sensores.
- Controlador.
- Actuador.

A continuación se describen las propiedades del giroscopio, así como su modelo matemático.

### 5.1 Giroscopio de 3 grados de libertad

Los giroscopios son usados para detección del movimiento angular, son el corazón de los pilotos automáticos en aeroplanos, sistemas de cohetes guiados, vehículos espaciales, sistemas de control de altitud, girocompás de barcos y autonavegadores internos en submarinos [Robert H. Cannon (2003)].

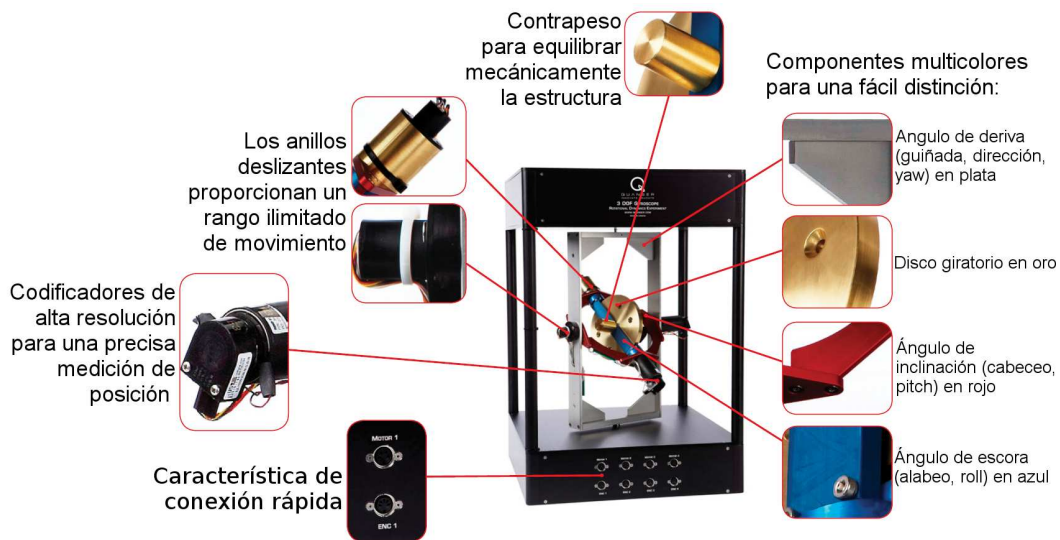


Figura 5.1: Partes del giroscopio de 3 grados de libertad [Quanser (2012a)].

	Voltaje	C. pico	C. nominal	Frecuencia limite	Encoder
Motor 1	$\pm 10V$	3A	1A	50HZ	1024 lineas/rev
Motor 2, 3, 4	$\pm 10V$	5A	3A	50HZ	1000 lineas/rev

Tabla 5.1: Características de los motores del giroscopio [Quanser (2012b)]

El giroscopio de tres grados de libertad utiliza los principios de momento angular para medir y mantener la orientación [Quanser (2012a)]. Este sistema robusto consiste de un disco montado dentro de una suspensión cardán la cual a su vez está montada en una suspensión cardán externa color roja. La estructura completa está soportada por un marco plateado rectangular que es libre de girar sobre sus eje verticales de simetría usando un diseño de anillo rotatorio. Las suspensiones cardanes también están equipadas con anillos rotatorios, permitiéndoles rotar libremente y dándole al disco tres grados de libertad. La planta está equipada con cuatro motores DC y cuatro encoders. Motores separados accionan el giro del disco, las suspensiones cardanes roja y azul. El cuarto motor puede ser usado para crear una perturbación controlada. Todas las partes se ilustran en la Figura 5.1. Las características de los motores se muestran en la Tabla 5.1.

Siguiendo el marco de coordenadas de la Figura 5.2, despreciando uno de los grados de libertad (marco gris) el cual se suele utilizar para ingresar perturbaciones al

sistema, se calculan las ecuaciones de movimiento, se asume que el rotor está perfectamente balanceado, y que los soportes son rígidos y perfectamente centrados, se asume también que el par  $M_b$  es ejercido sobre eje  $y$ , entre las suspensiones azul y roja y que un par  $M_r$  es ejercido sobre el eje  $z$  en la suspensión externa (roja) por la base (marco gris), en este análisis se toma el marco gris como una referencia interna.

La Figura 5.2, muestra el sistema en una configuración arbitraria (con respecto a una referencia XYZ), y coordenadas  $\phi$  y  $\psi$  son definidas. El sistema tiene dos grados de libertad (la velocidad  $\dot{\theta}$  es constante).

Definiendo dos marcos de referencia, un marco de referencia  $\alpha$  fijado en la suspensión interna (azul), y el marco de referencia  $\beta$  fijado en la suspensión externa (roja).

La velocidad angular absoluta del marco  $\beta$  es

$$\Omega_\beta = \dot{\phi} \hat{k} \quad (5.1)$$

$\hat{k}$  es un vector unitario sobre el eje  $z$ .

La velocidad y aceleración angular de un cuerpo rígido son cantidades vectoriales. Obedecen la suma de vectores. Se define  $\Omega_{\alpha/\beta}$  como la velocidad angular del marco  $\alpha$  relativa al marco  $\beta$ , siendo  $\Omega_\beta$  la velocidad angular absoluta del marco  $\beta$ , entonces la velocidad angular absoluta del marco  $\alpha$  es

$$\Omega_\alpha = \Omega_{\alpha/\beta} + \Omega_\beta = \dot{\phi} \hat{j} + \dot{\psi} \hat{K} \quad (5.2)$$

Reescribiendo  $\dot{\psi} \hat{K}$  en términos de sus componentes

$$\dot{\psi} \hat{K} = -\dot{\psi} \sin(\phi) \hat{i} + \dot{\psi} \cos(\phi) \hat{j} \quad (5.3)$$

sustituyendo (5.3) en (5.2)

$$\Omega_\alpha = -\dot{\psi} \sin(\phi) \hat{i} + \dot{\phi} \hat{j} + \dot{\psi} \cos(\phi) \hat{k} \quad (5.4)$$

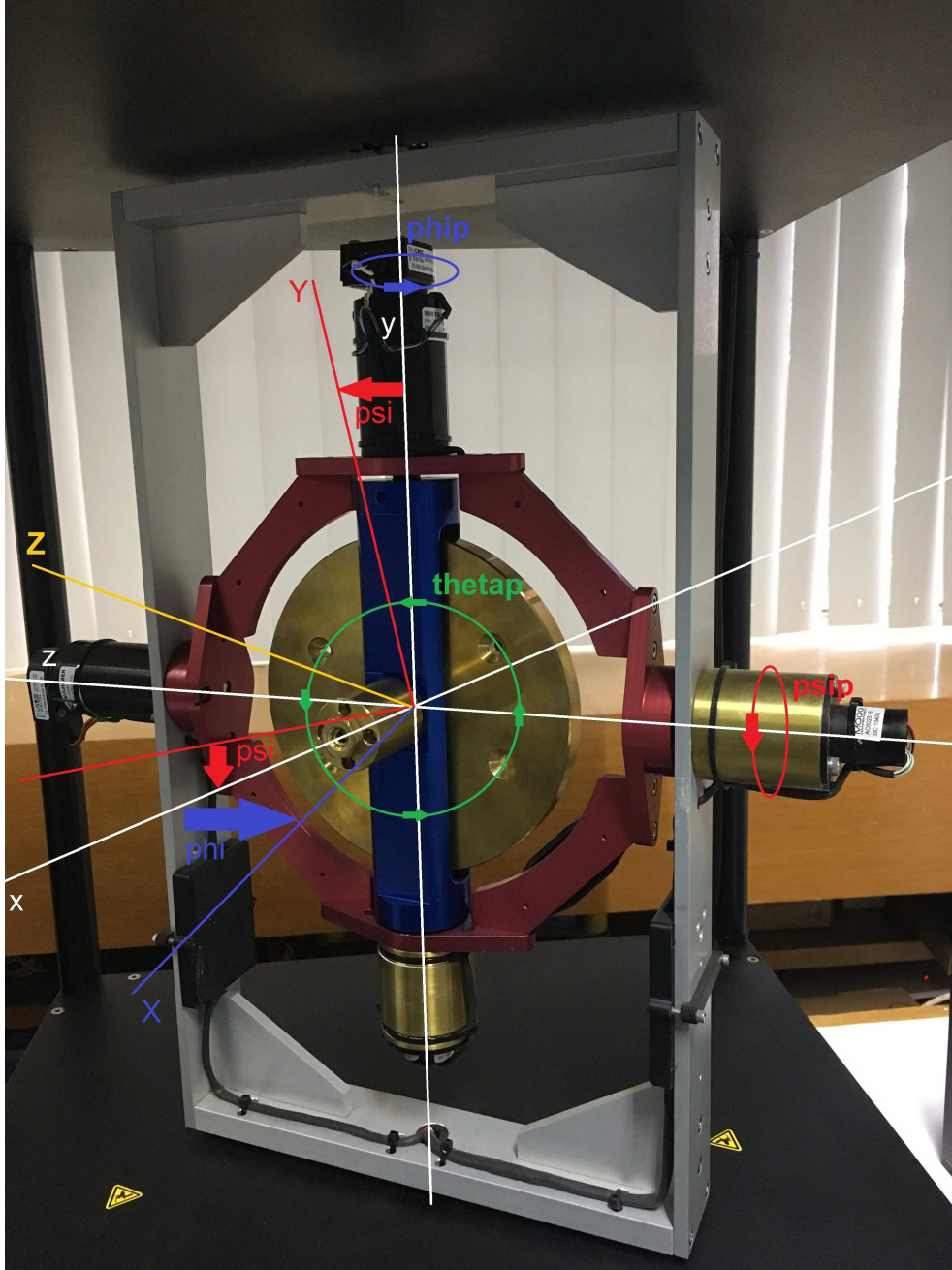


Figura 5.2: Marco de coordenadas del giroscopio.

Por último la velocidad angular del rotor

$$\Omega_d = \Omega_{d/\alpha} + \Omega_d = \dot{\theta}\hat{i} + \dot{\phi}\hat{j} + \psi\hat{K} \quad (5.5)$$

de manera similar que en 5.2 se reescribe  $\psi\hat{K}$  en sus componentes, entonces  $\Omega_d$  queda de la siguiente forma

$$\Omega_d = (\dot{\theta} - \psi \sin(\phi))\hat{i} + \dot{\phi}\hat{j} + \psi \cos(\phi)\hat{k} \quad (5.6)$$

Utilizando el principio de D' Alembert [Lanczos (1949)]

$$\sum M^*_{\text{sobre cualquier eje}} = 0 \quad (5.7)$$

donde  $M^*$  son todas las inercias y fuerzas que actúan sobre un eje, se pueden encontrar las relaciones del sistema.

Gracias a la condición que asume que el sistema se encuentra en perfecto balance, las fuerzas de rigidez tiene momento cero sobre el eje y, entonces la ecuación de equilibrio sobre el eje y, tomando sólo en cuenta el rotor y la suspensión interna (azul) es.

$$\sum M^*_y = (M_i^d + M_i^b)_y + M_b \quad (5.8)$$

$M_i^r$  es la inercia en el rotor,  $M_i^b$  es la inercia en la suspensión, el par  $M_b$  es el ejercido por el motor que actúa sobre eje y.

De manera similar se la ecuación de equilibrio que relaciona ambas suspensiones y el rotor es

$$\sum M^*_z = (M_i^d + M_i^b + M_i^r)_z + M_r = 0 \quad (5.9)$$

Donde  $M_i^r$  es la inercia de la suspensión externa (roja) y  $M_r$  es el par del motor que actúa sobre el eje z.

$(M_i^d + M_i^b)_z$  puede ser escrita en términos de sus componentes

$$(M_i^d + M_i^b)_Z = (M_i^d + M_i^b)_z \cos(\phi) + (M_i^d + M_i^b)_x \sin(\phi) \quad (5.10)$$

sustituyendo (5.10) en (5.9)

$$(M_i^d + M_i^b)_z \cos(\phi) - (M_i^d + M_i^b)_x \sin(\phi) + (M_i^r)_Z + M_r = 0 \quad (5.11)$$

Se pueden escribir expresiones para los momentos de inercia usando las siguientes relaciones físicas [Robert H. Cannon (2003)].

En movimientos rotacionales de dos dimensiones se simplifican las ecuaciones de movimientos, porque están enfocadas sólo en los en de rotación y vectores de momento angular en una simple dirección, es decir perpendicular al plano.

$$-M_i = \dot{H}_c = (J_z \dot{\Omega}_z) \hat{K} \quad (5.12)$$

$H_c$  es la tasa de cambio del momento angular, el cual es ahora perpendicular al plano de movimiento.

Por otro lado la ecuación que describe la tasa de cambio  $\dot{H}_c$  con respecto a un marco de referencia no rotativo es

$$\dot{H}_c = H_c^\alpha + \Omega^\alpha \times H_c \quad (5.13)$$

con  $H_c^\alpha$  es la tasa de cambio de  $H_c$  desde el punto de vista de un marco  $\alpha$ .

Aplicando las relaciones (5.12) y (5.13) al caso del giroscopio se obtienen las siguientes ecuaciones:

$$-M_i^d = \dot{H}^d = \overset{\alpha}{H}^d + \Omega^\alpha \times H^d \quad (5.14)$$

$$-M_i^b = \dot{H}^b = \overset{\alpha}{H}^b + \Omega^\alpha \times H^b \quad (5.15)$$

$$-M_i^r = \dot{H}^r = \overset{\beta}{H}^r + \Omega^\beta \times H^r \quad (5.16)$$



La diferenciación en los marcos  $\alpha$  y  $\beta$ , fueron seleccionados porque  $H^d$  y  $H^b$  son constantes expresadas en el marco  $\alpha$ , mientras que  $H^r$  es constante desde el punto de vista del marco  $\beta$ . Por simetría los ejes principales son  $x$ ,  $y$ ,  $z$ .

El valor de un momento angular  $H_c$  para un cuerpo rígido teniendo movimiento general se expresa de una forma más sencilla de términos de los momentos de inercia del cuerpo sobre sus tres principales ejes a través de su centro de masa ( $x_p$ ,  $y_p$ ,  $z_p$ ). Entonces:

$$H_c = (J_{xp}\Omega_{xp})\hat{i}_p + (J_{yp}\Omega_{yp})\hat{j} + (J_{zp}\Omega_{zp})\hat{k} \quad (5.17)$$

Aplicando la relación (5.17) al problema del giroscopio, la ecuaciones que relacionan el momento angular en términos de los momento de inercia son

$$H^d = [J_x^d\Omega_{dx}] + [J_y^d\Omega_{dy}] + [J_z^d\Omega_{dz}] \quad (5.18)$$

$$H^b = [J_x^b\Omega_{bx}] + [J_y^b\Omega_{by}] + [J_z^b\Omega_{bz}] \quad (5.19)$$

$$H^r = [J_x^r\Omega_{rx}] + [J_y^r\Omega_{ry}] + [J_z^r\Omega_{rz}] \quad (5.20)$$

donde de manera general el término  $J_i^j$  es el momento de inercia de  $i$  sobre el eje  $j$ , por ejemplo,  $J_x^d$  es el momento de inercia del disco sobre el eje  $x$ , mientras el término  $\Omega_{ij}$  se refiere a la velocidad angular de  $i$  sobre el eje  $j$ , por ejemplo,  $\Omega_{by}$  hace referencia a la velocidad angular de la suspensión interna (azul) sobre el eje  $y$ .

La suspensión externa (roja) tiene como eje principal a  $Z$ , sustituyendo (5.1) y (5.3) en las ecuaciones (5.18)-(5.20) el sistema queda como:

$$H^d = [J_x^d(\dot{\theta} - \psi \sin(\phi))]\hat{i} + [J_y^d\dot{\phi}]\hat{j} + [J_z^d\dot{\phi} \cos(\phi)]\hat{k} \quad (5.21)$$

$$H^b = [J_x^b(-\psi \sin(\phi))]\hat{i} + [J_y^b\dot{\phi}]\hat{j} + [J_z^b\psi \cos(\phi)]\hat{k} \quad (5.22)$$

$$H^r = [J_z^r\dot{\phi}]\hat{K} \quad (5.23)$$

Se definen la constante "momento de giro"

$$h \triangleq J_z^d \dot{\omega} \quad (5.24)$$

Además se definen constantes de inercias generales sobre los eje principales (xyz) de la suspensión interna (azul) y el rotor (disco).

$$J_x \triangleq J_x^d + J_x^b \quad (5.25)$$

$$J_y \triangleq J_y^d + J_z^d \quad (5.26)$$

$$J_z \triangleq J_y^d + J_z^d \quad (5.27)$$

Sustituyendo (5.21)-(5.23) en (5.14)-(5.16) para obtener los términos que hacen falta en las ecuaciones de equilibrio (5.8) y (5.10).

$$-(M_i^d + M_i^b)_y = J_y \ddot{\phi} + h\dot{\psi} \cos(\phi) + (J_z + J_x) \dot{\psi}^2 \sin(\phi) \cos(\phi) \quad (5.28)$$

$$\begin{aligned} -(M_i^d + M_i^b)_z \cos(\phi) &= J_z (\ddot{\phi} \cos^2(\phi) - \dot{\psi} \dot{\psi} \sin(\phi) \cos(\phi)) - h\dot{\phi} \cos(\phi) \\ &+ (J_z - J_y) \dot{\psi} \dot{\phi} \sin(\phi) \cos(\phi) \end{aligned} \quad (5.29)$$

$$\begin{aligned} -(M_i^d + M_i^b)_z \sin(\phi) &= -J_x (\ddot{\phi} \sin^2(\phi) + \dot{\psi} \dot{\phi} \cos(\phi) \sin(\phi)) \\ &+ (J_z - J_y) \dot{\psi} \dot{\phi} \cos(\phi) \sin(\phi) \end{aligned} \quad (5.30)$$

Sustituyendo (5.28), (5.29) y(5.30) en las ecuaciones de equilibrio (5.8) y (5.10) se obtienen las ecuaciones de movimiento para un giroscopio ideal de dos grados de libertad [Emedi and Karimi (2015)]:

$$J_y \ddot{\phi} - J_x^d \dot{\theta} \dot{\psi} \cos(\phi) + (J_z - J_x) \dot{\psi}^2 \sin(\phi) \cos(\phi) = M_b \quad (5.31)$$

$$(J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)) \ddot{\phi} + J_x^d \dot{\theta} \dot{\phi} \cos(\phi) + 2(J_x - J_z) \dot{\psi} \dot{\phi} \sin(\phi) \cos \phi = M_r \quad (5.32)$$

Resumiendo los términos que componen las ecuaciones de movimiento (5.31) y (5.32) son:

$\dot{\theta}$  velocidad angular del disco alrededor de su propio eje de rotación  $x$ .

$\phi$  posición angular de la suspensión cardán azul sobre  $y$ .

$\psi$  posición angular de la suspensión cardán roja sobre  $Z$ .

$J_z^r$  momento de inercia de la suspensión cardán roja sobre el eje  $Z$ .

$J_z^d$  momento de inercia del disco sobre el eje  $x$ .

$J_x$  momento de inercia del disco y la suspensión cardán azul alrededor del eje  $x$ .

$J_y$  momento de inercia del disco y la suspensión cardán azul alrededor del eje  $y$ .

$J_z$  momento de inercia del disco y la suspensión cardán azul alrededor del eje  $z$ .

$M_b$  par externo total alrededor del eje de rotación de la suspensión cardán azul.

$M_r$  par externo total alrededor del eje de rotación de la suspensión cardán roja.

El modelo se puede simplificar si se toman en cuenta los valores de los parámetros obtenidos del manual de usuario [Apkarian and Javid (2013)].

- $\theta = 150rad/s$
- $J_z^r = 0.0342kgm^2$
- $J_z^d = 0.0056kgm^2$
- $J_x = 0.0074kgm^2$
- $J_y = 0.0026kgm^2$
- $J_z = 0.0056kgm^2$

Debido a estos valores, en la ecuación (5.31) se observa que  $|J_x^d \dot{\theta} \dot{\psi} \cos(\phi)| \gg |J_x^d \dot{\theta} \dot{\psi} \cos(\phi)|$ , de manera similar en la ecuación (5.32) se aprecia que  $|J_x^d \dot{\theta} \dot{\phi} \cos(\phi)| \gg |2(J_x - J_z) \dot{\psi} \dot{\phi} \sin(\phi) \cos \phi|$ . Por consecuencia el sistema se puede aproximar de la siguiente manera:

$$J_y \ddot{\phi} - J_x^d \dot{\theta} \dot{\psi} \cos(\phi) = M_b \quad (5.33)$$

$$(J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)) \ddot{\phi} + J_x^d \dot{\theta} \dot{\phi} \cos(\phi) = M_r \quad (5.34)$$

### 5.1.1 Modelo no lineal en espacio de estados

Se define el vector de estados:

$$X_g^T = [\dot{\phi} \quad \ddot{\phi} \quad \dot{\psi} \quad \ddot{\psi}] \quad (5.35)$$

despejando  $\ddot{\phi}$  de la ecuación 5.33:

$$\ddot{\phi} = \frac{J_x^d \dot{\theta} \dot{\psi} \cos(\phi)}{J_y} \dot{\phi} + \frac{M_b}{J_y} \quad (5.36)$$

De manera similar se despeja  $\ddot{\psi}$  de la ecuación 5.34:

$$\ddot{\psi} = -\frac{J_x^d \dot{\theta} \dot{\phi} \cos(\phi)}{J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)} \dot{\phi} + \frac{M_r}{J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)} \quad (5.37)$$

Además  $M_b = K_b u_b$  y  $M_r = K_r u_r$ , ignorando las dinámicas del amplificador de potencia y motores.  $u_b$  y  $u_r$  es las entradas de control de los motores que actúan sobre las suspensiones roja y azul respectivamente, y  $K_b = K_r = 0.03985$  es la constante eléctrica de los motores.

De esta manera se puede definir el sistema en espacio de estados:

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{J_x^d \dot{\theta} \cos(\phi)}{J_y} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{J_x^d \dot{\theta} \cos(\phi)}{J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)} & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \psi \\ \dot{\psi} \end{bmatrix} \quad (5.38)$$

$$+ \begin{bmatrix} 0 & 0 \\ \frac{k_p}{J_y} & 0 \\ 0 & 0 \\ 0 & \frac{k_r}{J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

## 5.2 Variables premisa

Se definen las variables premisa:

$$z_1 = \frac{J_x^d \dot{\theta} \cos(\phi)}{J_y} \quad (5.39)$$

$$z_2 = -\frac{J_x^d \dot{\theta} \cos(\phi)}{J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)} \quad (5.40)$$

$$z_3 = \frac{k_r}{J_z^r + J_z \cos^2(\phi) + J_x \sin^2(\phi)} \quad (5.41)$$

Se restringen los ángulos  $\phi$  y  $\psi$  ambos en  $[-45^\circ, 45^\circ]$ , debido a que no se trabaja con toda la circunferencia.

Los máximos y mínimos de las variables premisa son:

$$\begin{aligned} z_{1MAX} &= 326.1^\circ & z_{1min} &= 228.4^\circ \\ z_{2MAX} &= -16.92^\circ & z_{2min} &= -24.56^\circ \\ z_{3MAX} &= 1.165^\circ & z_{3min} &= 1.135^\circ \end{aligned} \quad (5.42)$$

Usando las ecuaciones 4.37 y 4.38, se calculan las funciones de pertenencia.

$$\begin{aligned}
M_{11}(z_1) &= 1 + \frac{z_1 - 326.1}{97.7} & M_{12}(z_1) &= -\frac{z_1 - 326.1}{97.7} \\
M_{21}(z_2) &= 1 + \frac{z_2 + 16.92}{7.64} & M_{22}(z_2) &= -\frac{z_2 + 16.92}{7.64} \\
M_{31}(z_3) &= 1 + \frac{z_3 - 1.165}{0.03} & M_{32}(z_3) &= \frac{z_3 - 1.165}{0.03}
\end{aligned} \tag{5.43}$$

Las funciones de pertenencia (5.43) cumplen las propiedades descritas por las ecuaciones (2.1), las consecuencias se enlistan en la Tabla 5.2.

$w_i(z)$	IF	THEN $A_i + B_i u$
$w_1(z)$	$M_{11}(z_1) \cdot M_{21}(z_2) \cdot M_{31}(z_3)$	$Ag_1x + Bg_1u$
$w_2(z)$	$M_{11}(z_1) \cdot M_{21}(z_2) \cdot M_{32}(z_3)$	$Ag_1x + Bg_2u$
$w_3(z)$	$M_{11}(z_1) \cdot M_{22}(z_2) \cdot M_{31}(z_3)$	$Ag_2x + Bg_1u$
$w_4(z)$	$M_{11}(z_1) \cdot M_{22}(z_2) \cdot M_{32}(z_3)$	$Ag_2x + Bg_2u$
$w_5(z)$	$M_{12}(z_1) \cdot M_{21}(z_2) \cdot M_{31}(z_3)$	$Ag_3x + Bg_1u$
$w_6(z)$	$M_{12}(z_1) \cdot M_{21}(z_2) \cdot M_{32}(z_3)$	$Ag_3x + Bg_2u$
$w_7(z)$	$M_{12}(z_1) \cdot M_{22}(z_2) \cdot M_{31}(z_3)$	$Ag_4x + Bg_1u$
$w_8(z)$	$M_{12}(z_1) \cdot M_{22}(z_2) \cdot M_{32}(z_3)$	$Ag_4x + Bg_2u$

Tabla 5.2: Combinaciones entre las funciones de membresía para el giroscopio

Los valores de las matrices de los subsistemas consecuentes son los siguientes:

$$Ag_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 228.4 \\ 0 & 0 & 0 & 1 \\ 0 & -24.56 & 0 & 0 \end{bmatrix} \tag{5.44}$$

$$Ag_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 228.4 \\ 0 & 0 & 0 & 1 \\ 0 & -16.92 & 0 & 0 \end{bmatrix} \tag{5.45}$$

$$A_{g3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 326.1 \\ 0 & 0 & 0 & 1. \\ 0 & -24.56 & 0 & 0 \end{bmatrix} \quad (5.46)$$

$$A_{g4} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 326.1 \\ 0 & 0 & 0 & 1 \\ 0 & -16.92 & 0 & 0 \end{bmatrix} \quad (5.47)$$

$$B_{g1} = \begin{bmatrix} 0 & 0 \\ 15.3269 & 0 \\ 0 & 0 \\ 0 & 1.1350 \end{bmatrix} \quad (5.48)$$

$$B_{g2} = \begin{bmatrix} 0 & 0 \\ 15.3269 & 0 \\ 0 & 0 \\ 0 & 1.1650 \end{bmatrix} \quad (5.49)$$

### 5.3 Observabilidad y controlabilidad

El sistema es controlable debido a que cada matriz de controlabilidad conformada por cada par  $A_{g_i}$  y  $B_{g_i}$  es de rango completo.

De manera similar cada par de matriz de observabilidad conformado por  $A_{g_i}$  y  $C_g$  es de rango completo, por lo que el sistema es observable.

### 5.4 Controlador

Al ser un sistema difuso continuo debe cumplir con la siguiente LMI (5.50)-(5.52).

$$-XA_{gi}^T - A_{gi}X + M_{gi}^T B_{gi}^T + B_{gi}M_{gi} > 0 \quad (5.50)$$

$$-XA_{gi}^T - A_{gi}X - XA_{gj}^T - A_{gj}X + M_{gj}^T B_{gi}^T + B_{gi}M_{gj} + M_{gi}^T B_{gj}^T + B_{gj}M_{gi} \geq 0 \quad (5.51)$$

$$i < j : h_i \cap h_j \neq 0 \quad (5.52)$$

Donde  $G_{ij} = A_{gi} - B_{gi}F_{gj}$ , además  $X_{gi} = P_g^{-1}$ ,  $M_{gi} = F_{gi}X_g$ .

Las condiciones (5.50)-(5.52) son LMIs con respecto a  $X_g$  y  $M_{gi}$ , se puede calcular una matriz positiva definida  $X_g$  y  $M_{gi}$  satisfaciendo las LMIs o determinar que tal  $X_g$  y  $M_{gi}$  no existen. Las ganancias  $F_{gi}$  y  $P_g$  común pueden ser obtenidas como:

$$P_g = X_g^{-1}, \quad F_{gi} = M_{gi}X_g^{-1} \quad (5.53)$$

Un esquema del controlador difuso para el giroscopio se puede observar en la Figura 5.3.

En las Figura 5.4 se observa el seguimiento de los ángulos  $\phi = -5$  y  $\psi = 3$  y en la Figura 5.5 se muestra el error de dicho seguimiento.

## 5.5 Resumen del capítulo

Se llevo a cabo la fuzzificación del modelo del giroscopio de tres grados de libertad, usando la teoría del modelo Takagi-Sugeno mostrada en el capítulo 2, una vez obtenido el modelo difuso se comprobó sus propiedades de controlabilidad y observabilidad, se llevaron a cabo los calculos para un controlador de este sistema, las ganancias del controlador se obtuvieron con ayuda de una *Linear matrix inequality* (LMI), el sistema de control se simuló en el programa *SIMULINK* para comprobar que el error convergiera a cero en dos de los tres grados de libertad.

Esta aproximación está conformada por 8 subsistemas lineales que representan en su totalidad al sistema no lineal en el rango de  $\phi = [-45^\circ, 45^\circ]$  y  $\psi = [-45^\circ, 45^\circ]$ .



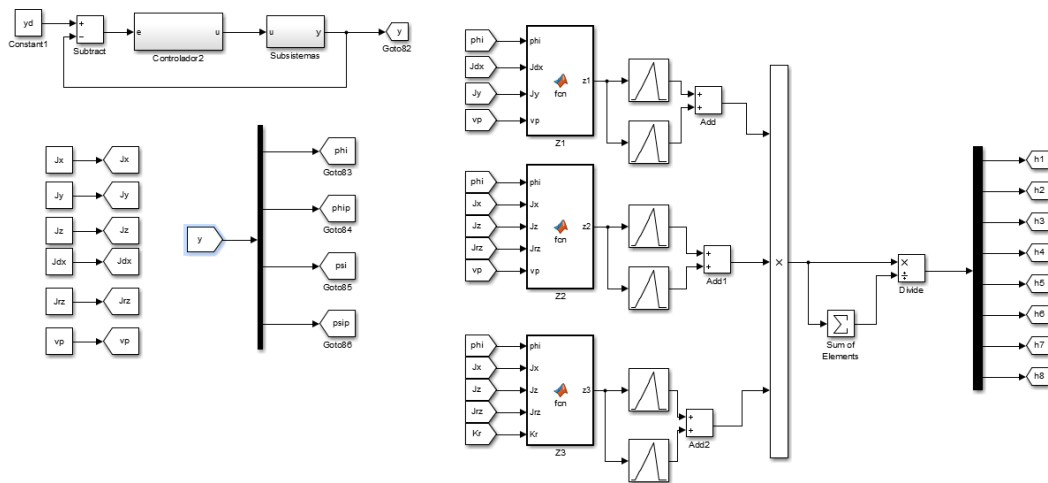


Figura 5.3: Esquema del controlador difuso del Giroscopio, los bloque de controlador dependen de los pesos  $h_i$  calculados al extremo izquierdo del diagrama, el bloque subsistemas contiene todas la matrices  $A_i$  y  $B_i$  de la parte consecuente.

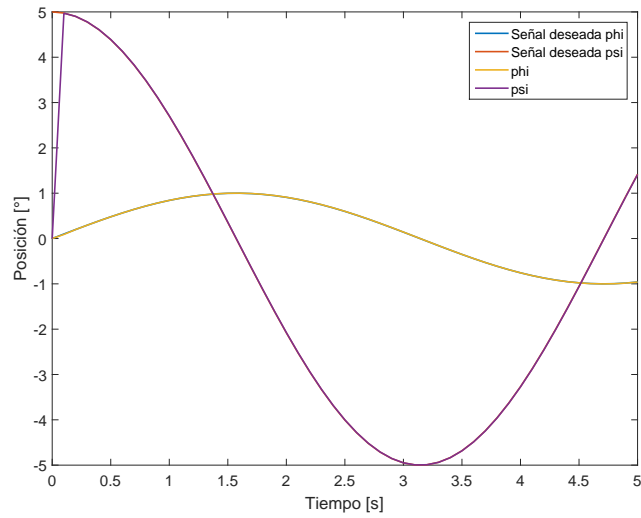


Figura 5.4: Posición deseada y posición de giroscopio

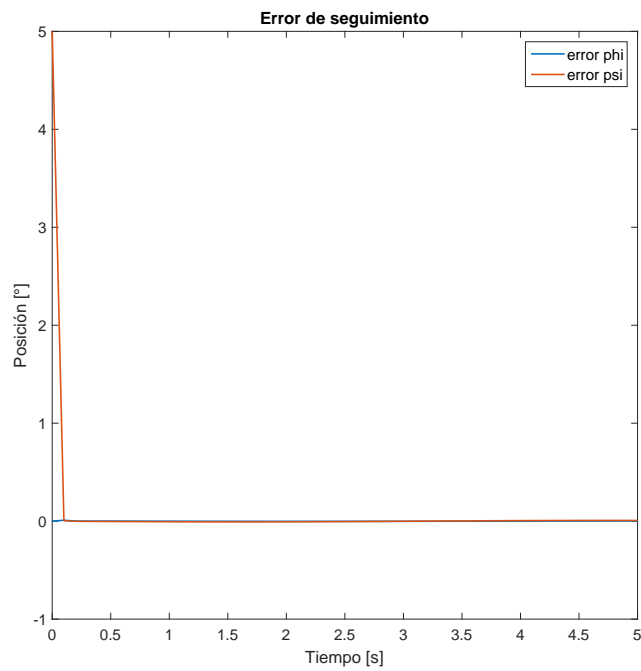


Figura 5.5: Señales de error del giroscopio.

# Capítulo 6

## Resultados

Se aplicaron los valores de plataforma virtual en forma de tiempo de tarea y nivel de servicio en forma de tiempo de muestreo, ambos calculado por el RM. El RM busca alcanzar la plataforma virtual deseada modificando el nivel de prioridad y la cantidad de plataforma virtual en las aplicaciones. Hay que recordar que el RM no tiene control sobre el nivel de servicio, debido a que es un proceso que se lleva a cabo dentro de cada aplicación. La señal deseada del giroscopio con un par de señales senoides de diferente fase y amplitud, dichas señales pueden ser observadas en la Figura 6.1.

Los parámetros utilizados son:

$$\varepsilon = 0.1$$

$$\beta_1 = 0.1$$

$$\beta_2 = 0.3$$

$$\beta_3 = 0.5$$

$$\underline{s}_1 = 2$$

$$\underline{s}_2 = 1.5$$

$$\underline{s}_3 = 1$$

## 6.1 Experimento 1

En este experimento se demuestra la importancia del nivel de servicio, en un sistema de control en tiempo real el nivel de servicio es el inverso al periodo de muestreo  $T_{si} = \frac{1}{s_i}$  [Chasparis et al. (2013)], por lo que un nivel de servicio alto puede resultar en un periodo de muestreo menor y traer como consecuencia una falla de las tres partes acopladas del sistema, por otra parte la plataforma virtual se le pidió al RM un nivel alto cómo se observa en la Figura 6.2 para que sólo se muestre el efecto del nivel de servicio, en la Figura 6.3 se puede apreciar como el RM alcanza los niveles deseados, mientras que en la Figura 6.4 se puede ver como el RM modifica las prioridades para alcanzar el objetivo que se le ha pedido. En la Figura 6.5 se muestra el nivel de servicio, junto con un límite inferior, el cual es el nivel de servicio que necesita el controlador para funcionar correctamente, el cual está calculado en 5, pues este controlador con los parámetros establecidos necesita un periodo de muestreo mínimo de  $0.2s$  para funcionar correctamente, se puede observar esto en las Figuras 6.5 y 6.6, esta última muestra la señal de error de seguimiento de los ángulos  $\phi$  (azul) y  $\psi$  (rojo) del giroscopio.

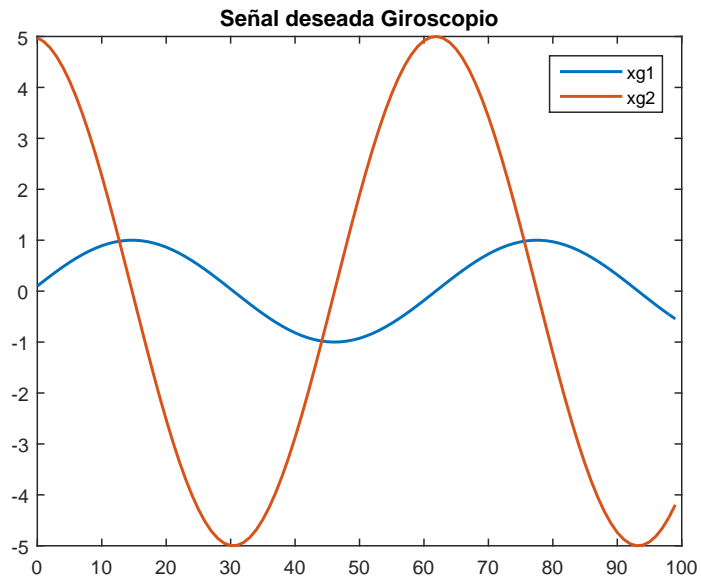


Figura 6.1: Señal deseada del giroscopio en todos los experimentos.

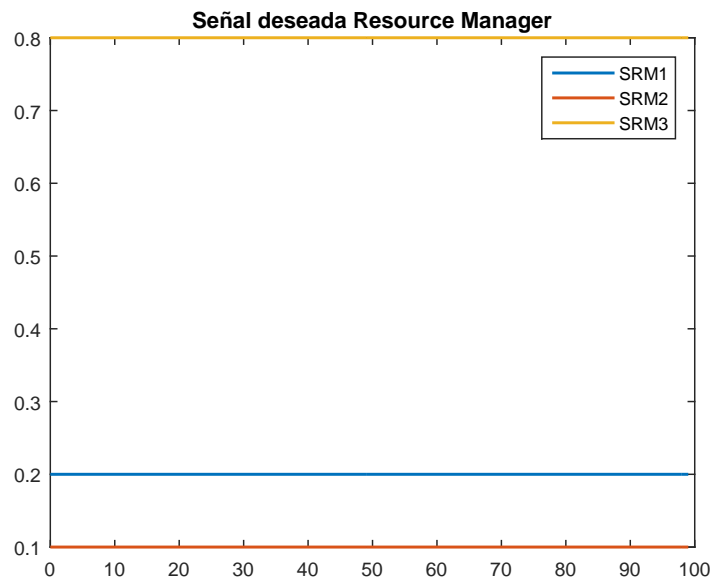


Figura 6.2: Señal deseada del Resource Manager experimento 1

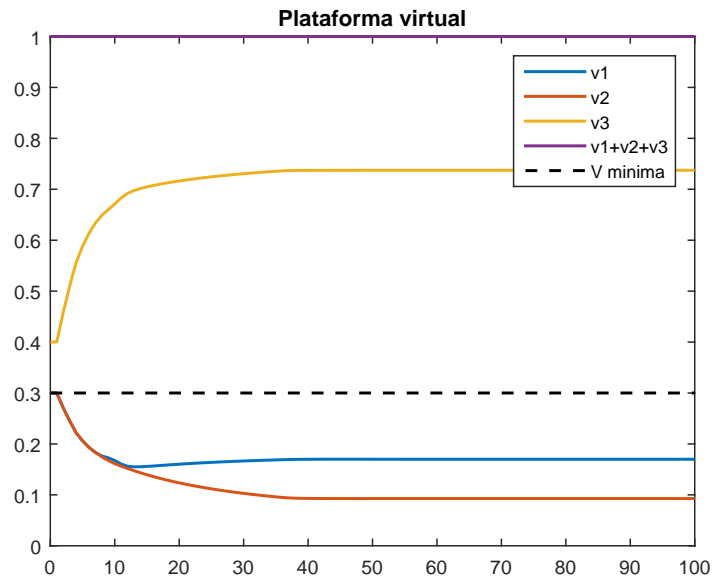


Figura 6.3: Plataformas virtuales del experimento 1

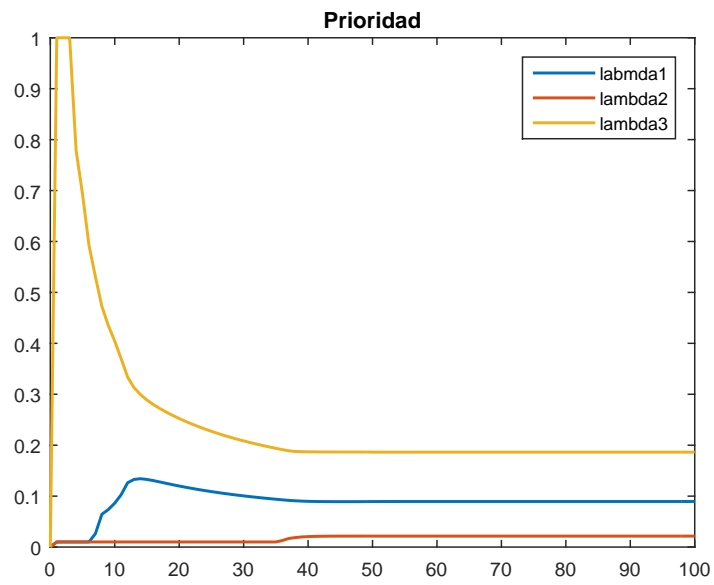


Figura 6.4: Prioridades del experimento 1

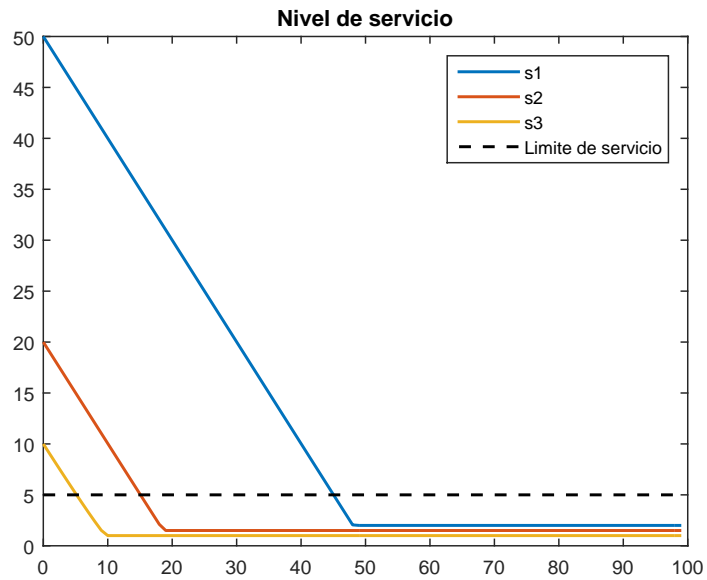


Figura 6.5: Niveles de servicio del experimento 1

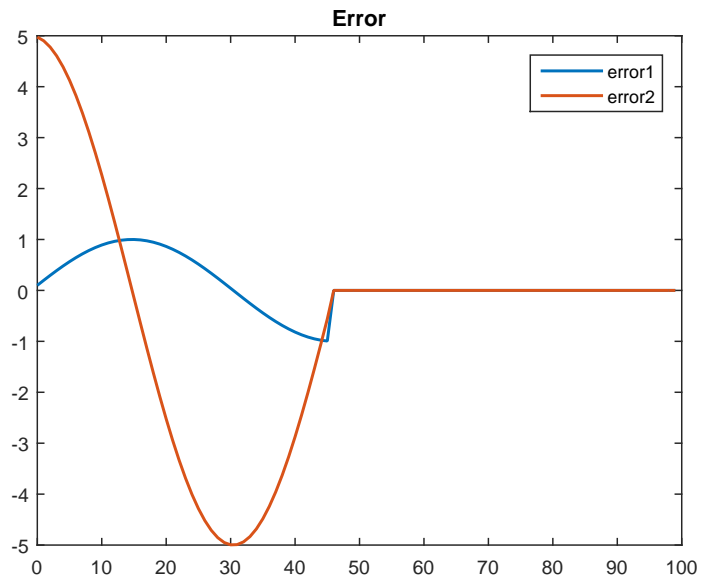


Figura 6.6: Señal de error del giroscopio experimento 1

## 6.2 Experimento 2

En los siguientes experimentos el nivel de servicio se puso en un nivel estable (por abajo de 5) para enfocar la atención en la plataforma virtual, además el tiempo de ejecución está sólo en relación con la aplicación 3. La señal deseada del RM son tres senoidales de diferente fase y frecuencia (Figura 6.7), el RM modifica los niveles de prioridad en función de la señal deseada, se debe notar que a pesar de los cambios en la plataforma virtual está sigue conservando su relación  $v_1 + v_2 + v_3 = 1$  como se muestra en la Figura 6.8. El resultado de dicha modificación se muestra en la Figura 6.9, junto con un nivel mínimo de plataforma virtual requerida por el sistema para el correcto funcionamiento del controlador. El error en los ángulos del giroscopio se observa en la Figura 6.10, donde el error en  $\psi$  comienza en su nivel máximo debido a que se trata de una señal seno con un desfase de  $\pi/2$  y que el controlador entra en acción hasta la muestra 1.

La plataforma virtual está relacionada con el tiempo de ejecución, dicha relación es  $C_i = R_i V_i$ , donde  $R_i$  es el tiempo de respuesta al cual se le tiene un valor de 2, y  $C_i$  es el tiempo de ejecución. Fijando la atención en la aplicación 3 se tiene que  $R_3 = 2$ , entonces  $C_3 = 2V_3$ , por lo tanto el tiempo de ejecución mínimo que requiere el sistema en la aplicación 3 es de 0.6s, por lo tanto la plataforma virtual mínima debe ser de  $v_3 = 0.3$ , lo cual se aprecia en la Figura 6.10, donde el error crece hasta el nivel de la señal deseada puesto que la salida de la planta se queda en cero justo cuando la plataforma virtual es menor al valor mínimo calculado.



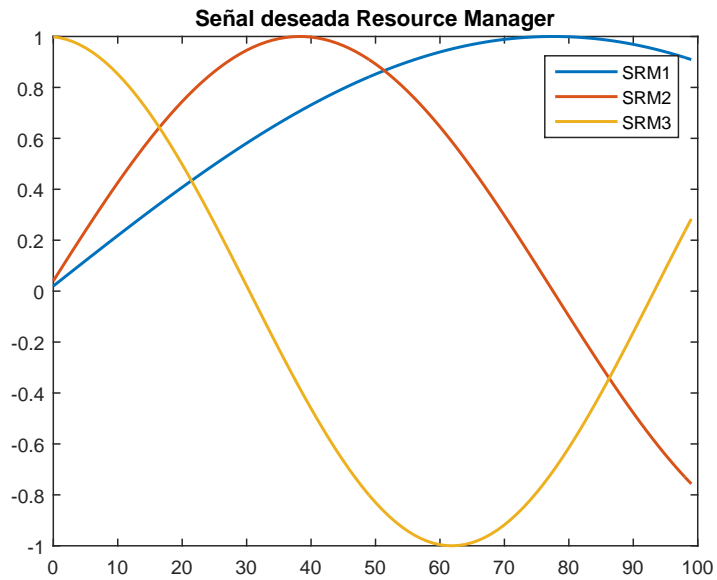


Figura 6.7: Señal deseada del Resource Manager experimento 2

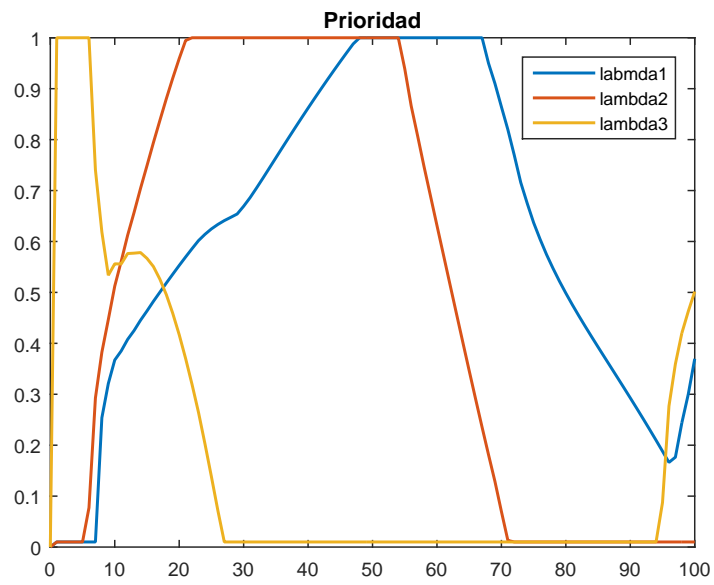


Figura 6.8: Prioridades del experimento 2

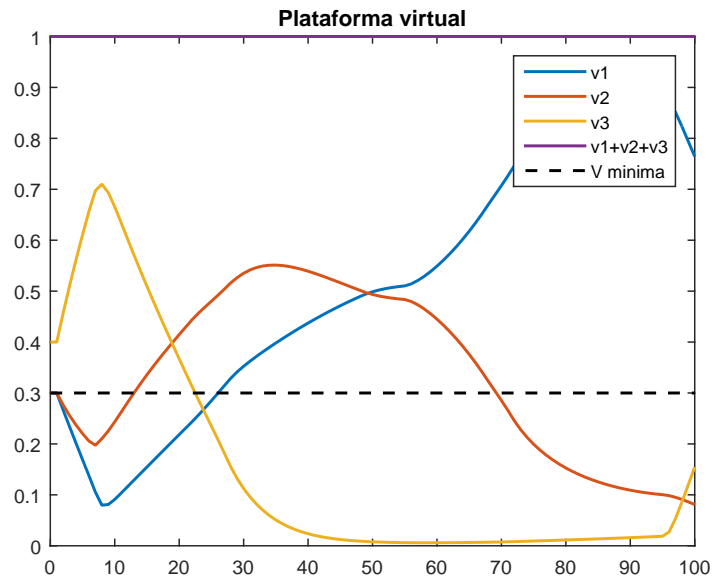


Figura 6.9: Plataformas virtuales del experimento 2

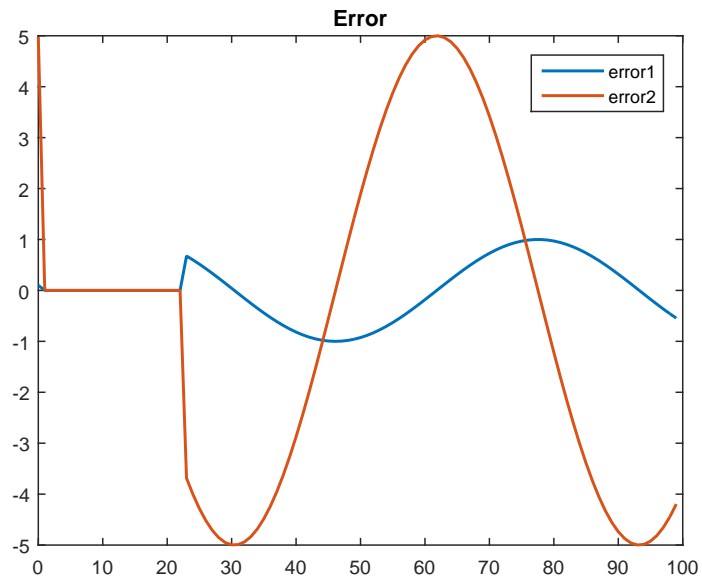


Figura 6.10: Señal de error del giroscopio experimento 2

## 6.3 Experimento 3

Para clarificar las ideas del experimento anterior se llevó a cabo el experimento 3, en el cual la plataforma virtual deseada (Figura 6.11) en la aplicación 3 pasa de un valor máximo a un mínimo, regresando en la muestra 75 a un valor apenas mayor al nivel mínimo de plataforma virtual requerido por el sistema para un correcto funcionamiento.

Se muestra en la Figura 6.13 como cambian los valores de plataforma virtual respecto a los valores deseados, estos convergen a los valores deseados de una forma suave debido a que el RM balancea las cargas de plataformas virtuales de tal manera que no sobrepasa los recursos disponibles, por otra parte está convergencia depende del parámetro  $\varepsilon$  el cual representa una cantidad mínima de recursos asignados o retirados a la aplicación.

Cómo se observa en la Figura 6.14 el error converge a cero cuando la aplicación 3 recibe la plataforma virtual suficiente que es 0.3 y cuando esta es menor a este valor el error vuelve crecer hasta los niveles de la señal deseada, sin embargo si se le vuelve a pedir más recursos para la aplicación 3, el RM vuelve a cambiar las prioridades (Figura 6.12) para alcanzar su objetivo, el sistema apenas alcanza la plataforma virtual requerida, este vuelve a llevar el error a cero.

El comportamiento tanto como del experimento 2 como en el 3 se debe a que la falla en una parte del sistema se transfiere a la siguiente, trayendo consigo un colapso del sistema, en este caso se ha podido restablecer el control y hacer converger el error a cero debido a que se trata de una simulación, en un sistema electromecánico real esto puede no ser posible, debido a que en el tiempo en que el control se restablece, el sistema puede ir a zonas para las que el controlador no fue diseñado, en este caso específico el controlador no está diseñado para ángulos  $|\phi| > 45^\circ$   $|\psi| > 45^\circ$ .

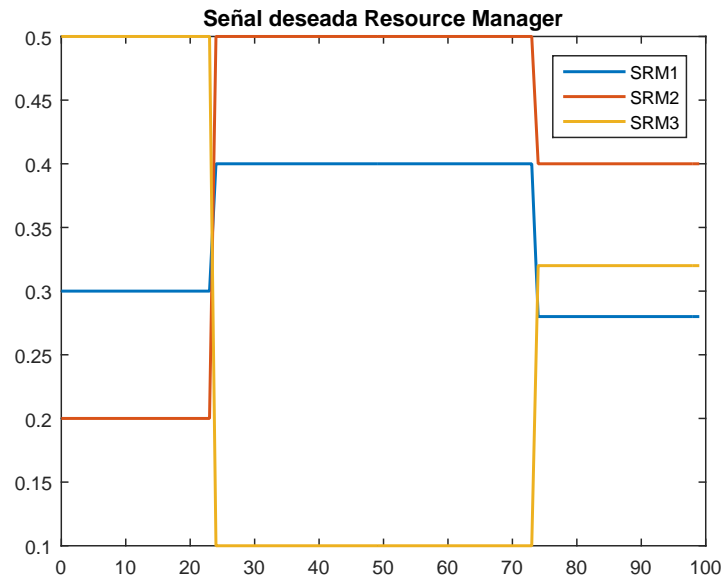


Figura 6.11: Señal deseada del Resource Manager experimento 3

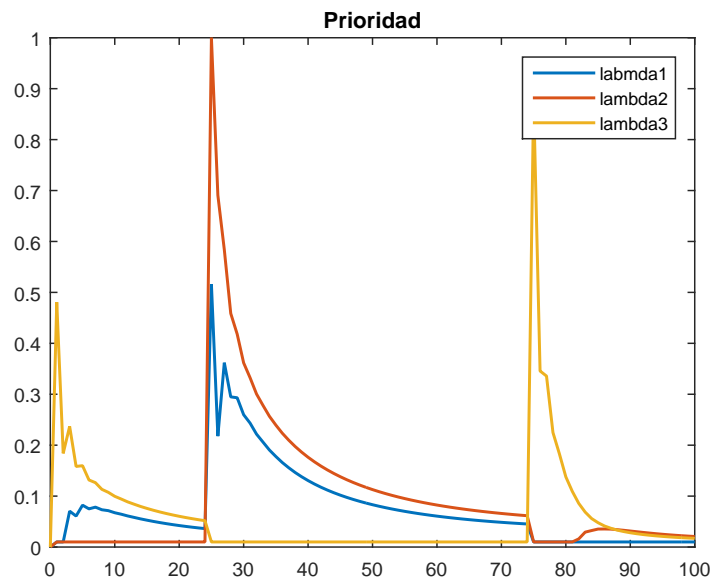


Figura 6.12: Prioridades del experimento 3

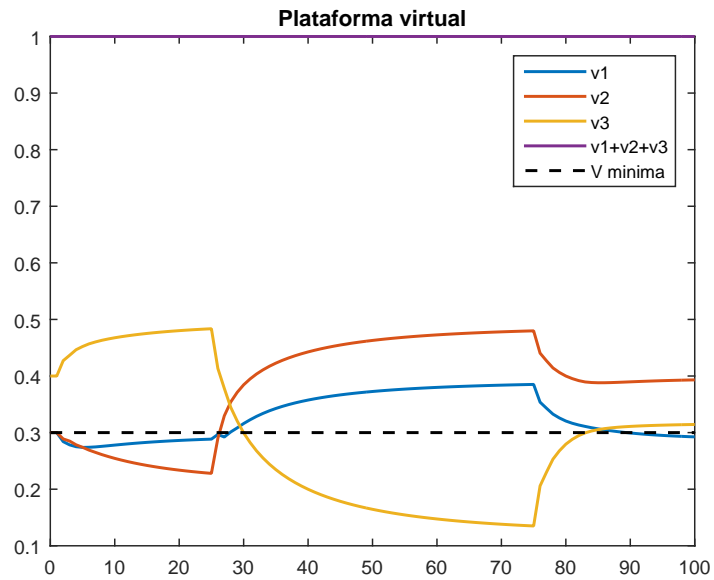


Figura 6.13: Plataformas virtuales del experimento 3

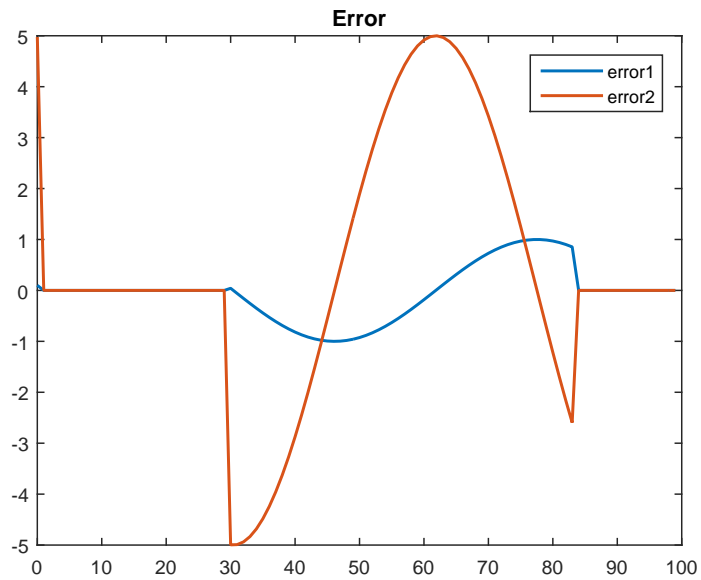


Figura 6.14: Señal de error del giroscopio experimento 3

# Capítulo 7

## Conclusiones

Se construyó un sistema de compensación de comunicación entre diversos nodos, en los cuales se lleva a cabo la adaptación de recursos y del nivel de servicio. Estos dos procesos se realizan de forma casi independiente, y sólo se enlazan gracias a la función de emparejamiento. Esto es muy útil debido a que los parámetros de las aplicaciones son en la mayoría de los casos desconocidos por el usuario.

Debido a que los sistemas de control son sensibles al tiempo, se simula la adaptación de recursos a un sistema de control en un sistema electromecánico, con el objetivo de demostrar que la adaptación de recursos da como resultado un ahorro de tiempo, utilizando como controlador un controlador difuso continuo debido a las no linealidades propias del sistema electromecánico.

El sistema presenta problemas de integración al acoplarse ambos sistemas, debido a que el sistema es muy lento con respecto al control, una posible solución al problema es usar algún algoritmo de integración numérica [Forsythe et al. (1977)], aun así se puede garantizar la estabilidad del sistema electromecánico con el RM acoplado debido a que ambos sistemas son asintóticamente estables.

### 7.1 Trabajo futuro

Se espera implementar el sistema de distribución de recursos en un sistema electromecánico real, una vez logrado, se llevará a cabo la implementación del sistema de distribución de recursos en diversos sistemas de control conectados a una misma red al "mismo" tiempo, esto puede llevar a una generalización o a una

predicción que podría ayudar a una mejor distribución de recursos en los sistemas de control.

# Bibliografía

- Abeni, L. and Buttazzo, G. (1998). Integrating multimedia applications in hard real-time systems. In *Proceedings 19th IEEE Real-Time Systems Symposium (Cat. No.98CB36279)*, pages 4–13.
- Al-Ali, R. J., Amin, K., von Laszewski, G., Rana, O. F., Walker, D. W., Hategan, M., and Zaluzeć, N. (2004). Analysis and provision of qos for distributed grid applications. *Journal of Grid Computing*, 2(2):163–182.
- Apkarian, J. and Javid, A. (2013). *Laboratory Guide 3DOF Gyroscope Experiment for MATLAB-Simulink*. Quanser inc.
- Bini, E., Buttazzo, G., Eker, J., Schorr, S., Guerra, R., Fohler, G., Arzen, K. E., Romero, V., and Scordino, C. (2011). Resource management on multicore systems: The actors approach. *IEEE Micro*, 31(3):72–81.
- Boutalis, Y., Theodoridis, D., Kottas, T., and Christodoulou, M. A. (2014). *System Identification and Adaptive Control: Theory and Applications of the Neurofuzzy and Fuzzy Cognitive Network Models*. Springer.
- Bowersox, D. J., Closs, D. J., and Cooper, M. B. (2002). *Supply chain logistics management*. Mc Graw Hill.
- Boyd, S., Feron, A., Ghaoui, L. E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics Philadelphia.
- Chasparis, G., Maggio, M., Årzén, K. E., and Bini, E. (2013). Distributed management of cpu resources for time-sensitive applications. In *2013 American Control Conference*, pages 5305–5312.



- Chasparis, G. C., Maggio, M., Bini, E., and Årzén, K.-E. (2016). Design and implementation of distributed resource management for time-sensitive applications. *Automatica*, 64:44 – 53.
- Cucinotta, T., Palopoli, L., Abeni, L., Faggioli, D., and Lipari, G. (2010). On the integration of application level and resource level qos control for real-time applications. *IEEE Transactions on Industrial Informatics*, 6(4):479–491.
- Duan, G.-R. and Yu, H.-H. (2013). *LMIs in Control Systems: Analysis, Design and Applications*. CRC Press.
- E.800, R. I.-T. (1994). *Terms and definitions related to quality of service and network performance including dependability*. International Telecommunication Union.
- Emedi, Z. and Karimi, A. (2015). Fixed-order linear parameter varying controller design for a 2dof gyroscope. *IFAC-PapersOnLine*, 48(26):230 – 235.
- Fang, C.-H., Liu, Y.-S., Kau, S.-W., Hong, L., and Lee, C.-H. (2006). A new lmi-based approach to relaxed quadratic stabilization of t-s fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, 14(3):386–397.
- Forsythe, G. E., Malcolm, M. A., and Moler, C. B. (1977). *Computer Methods for Mathematical Computations*. Prentice-Hall,.
- Ghaoui, L. E. and Niculescu, S.-I. (2000). *Advances in Linear Matrix Inequality Methods in Control*. Society for Industrial and Applied Mathematics Philadelphia.
- Golnaraghi, F. and Kuo, B. C. (2010). *Automatic Control Systems*. Joh Wiley & Sons, inc., 9th ed. edition.
- Goodwin, G. C., Graebe, S. F., and Salgado, M. E. (2000). *Control System Desing*. Prentice-Hall.
- Grosu, D. and Chronopoulos, A. T. (2005). Noncooperative load balancing in distributed systems. *Journal of Parallel and Distributed Computing*, 65(9):1022 – 1034.
- Jensen, E., Locke, C., and Tokuda, H. (1985). *TIME-DRIVEN SCHEDULING MODEL FOR REAL-TIME OPERATING SYSTEMS.*, pages 112–122. IEEE.

- Jones, M., Leach, P. J., Draves, R., and Barrera, J. S. (1995). Modular real-time resource management in the rialto operating system. Institute of Electrical and Electronics Engineers, Inc.
- Jones, M. and Rosu, M.-C. (1997). Cpu reservations and time constraints: Efficient, predictable scheduling of independent activities. Technical report.
- Kachroo, P. (1995). Modeling, analysis and control of fuzzy systems. In *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, volume 3, pages 2046–2051 vol.3.
- Kalman, R. (1959). On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3):110–110.
- Kircher, M. and Jain, P. (2004). *Pattern-Oriented Software Architecture Patterns for Resource Management*, volume 3. Joh Wiley & Sons, LTD.
- Lanczos, C. (1949). *The Variational Principles of Mechanics*. University of Toronto Press.
- Lee, C., Lehoczky, J., Siewiorek, D., Rajkumar, R., and Hansen, J. (1999). A scalable solution to the multi-resource qos problem. In *Proceedings 20th IEEE Real-Time Systems Symposium (Cat. No.99CB37054)*, pages 315–326.
- Li, B. and Nahrstedt, K. (1999). A control-based middleware framework for quality-of-service adaptations. *IEEE Journal on Selected Areas in Communications*, 17(9):1632–1650.
- Maggio, M., Bini, E., Chasparis, G., and Årzén, K. E. (2013). A game-theoretic resource manager for rt applications. In *2013 25th Euromicro Conference on Real-Time Systems*, pages 57–66.
- Mahajan, R., Rodrig, M., Wetherall, D., and Zahorjan, J. (2004). Experiences applying game theory to system design. In *Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems, PINS '04*, pages 183–190, New York, NY, USA. ACM.
- Mercer, C. W., Savage, S., and Tokuda, H. (1994). Processor capacity reserves: operating system support for multimedia applications. In *1994 Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 90–99.

- Nesbit, K. J., Moreto, M., Cazorla, F. J., Ramirez, A., Valero, M., and Smith, J. E. (2008). Multicore resource management. *IEEE Micro*, 28(3):6–16.
- Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition.
- Quanser (2012a). *3 DOF Gyroscope Brouchure*. Quanser inc.
- Quanser (2012b). *USER MANUAL 3 DOF Gyroscope Experiment Set Up and Configuration*. Quanser inc.
- Robert H. Cannon, J. (2003). *Dynamics Of Physical Systems*. Dover Publications, INC.
- Roughgarden, T. (2004). Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2):332–350.
- Årzén, K.-E., Romero Segovia, V., Schorr, S., and Fohler, G. (2011). Adaptive resource management made real.
- Saffady, W. (1986). *Informatica Documental para Bibliotecas*. Ediciones Diaz de Santos.
- Seidi, M., Hajiaghamemar, M., and Segee, B. (2012). Fuzzy control systems: Lmi-based design. In Iqbal, S., Boumella, N., and Garcia, J. C. F., editors, *Fuzzy Controllers- Recent Advances in Theory and Applications*, chapter 18. InTech, Rijeka.
- Sommerville, I. (2011). *Software engineering*. Addison-Wesley, 9th ed. edition.
- Steere, D. C., Goel, A., Gruenberg, J., McNamee, D., Pu, C., and Walpole, J. (1999). A feedback-driven proportion allocator for real-rate scheduling. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, pages 145–158, Berkeley, CA, USA. USENIX Association.
- Subrata, R., Zomaya, A. Y., and Landfeldt, B. (2008). A cooperative game framework for qos guided job allocation schemes in grids. *IEEE Transactions on Computers*, 57(10):1413–1422.

- Tanaka, K., Ikeda, T., and Wang, H. O. (1998). Fuzzy regulators and fuzzy observers: relaxed stability conditions and lmi-based designs. *IEEE Transactions on Fuzzy Systems*, 6(2):250–265.
- Tanaka, K. and Wang, H. O. (2001). *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*. Wiley-Interscience.
- Wang, W.-J. and Sun, C.-S. (2002). Relaxed stability condition for t-s fuzzy discrete system. In *Fuzzy Systems, 2002. FUZZ-IEEE'02. Proceedings of the 2002 IEEE International Conference on*, volume 1, pages 244–249.
- Wei, G., Vasilakos, A. V., Zheng, Y., and Xiong, N. (2010). A game theoretic method of fair resource allocation for cloud computing services. *The Journal of Supercomputing*, 54(2):252–269.
- Wüst, C. C., Steffens, L., Verhaegh, W. F. J., Bril, R. J., and Hentschel, C. (2005). Qos control strategies for high-quality video processing. *Real-Time Systems*, 30(1):7–29.