



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

“Estudio de estrategias de control PID basado en eventos
para la implementación de controles embebidos”

Tesis presentada a la

Facultad de Ciencias de la Electrónica

como requisito para la obtención del grado de

Licenciado en Ingeniería Mecatrónica

por

Moisés García Juárez

asesorado por

DR. JOSÉ FERMI GUERRERO CASTELLANOS

DR. GERMÁN ARDUL MUÑOZ HERNÁNDEZ

DR. CARLOS GRACIOS MARÍN

Puebla, Pue., México.
2014.

Título: “Estudio de estrategias de control PID basado en eventos para la implementación de controles embebidos”

Estudiante: MOISÉS GARCÍA JUÁREZ

COMITÉ

DR. GERARDO MINO
AGUILAR
Presidente

DRA. LILIANA CORTEZ
GEORGIEVNA
Secretario

M.C. RICARDO ÁLVAREZ
GONZÁLEZ
Vocal

ASESORES

DR. JOSÉ FERMI
GUERRERO CASTELLANOS

DR. GERMÁN ARDUL
MUÑOZ HERNÁNDEZ

Índice general

Agradecimientos	XI
Resumen	XIII
1. Introducción	1
1.1. Objetivos	2
1.2. Los sistemas de primer orden en el entorno científico y tecnológico	3
1.2.1. Servidor de e-mail	3
1.2.2. Propagación de onda en las redes neuronales	3
1.2.3. Motor de DC	4
1.2.4. Dinámica longitudinal de un automóvil	6
2. Contexto y Motivaciones	9
2.1. Formas simples de retroalimentación	9
2.1.1. Control automático	9
2.1.2. Retroalimentación	9
2.1.3. Control On- Off	11
2.1.4. Generalidades del controlador PID	11
2.1.5. Efectos de la acción proporcional, integral, derivativa, del controlador PID	12
2.1.6. Integrador Windup	13
2.1.7. Implementación en computadora	13
2.2. Aplicaciones del PID en la Industria	15
2.2.1. Control PID analógico SIM960	15
2.2.2. C2 Control de sistemas neumáticos	16
2.2.3. BTC-8300 Self-Tune Fuzzy / PID	16
2.3. Muestreo basado en tiempo vs. Muestreo basado en Eventos	17
2.4. Muestreo basado en eventos como una oportunidad para sistemas embebidos	18
2.4.1. La necesidad de un funcionamiento asíncrono en sistemas de control basados en redes de comunicaciones	18
2.4.2. Dificultades al conjugar el control asíncrono con paradigmas bien establecidos	19
2.5. Protocolo de comunicación CAN-BUS	19
2.5.1. Estructura de datos	22
2.5.1.1. Inicio de la estructura (SOF)	23
2.5.1.2. Campo de arbitraje	23
2.5.1.3. Campo de control	24
2.5.1.4. Campo de datos	25
2.5.1.5. Campo de CRC	25
2.5.1.6. Campo de ACK	25
2.6. TRUE TIME de MATLAB/Simulink	25
2.6.1. El núcleo de TRUE TIME	25

2.6.1.1.	Voltaje dinámico escalado	26
2.6.2.	La Red TRUE TIME	26
2.6.2.1.	CSMA/AMP(CAN)	28
2.6.3.	La batería de TRUE TIME	28
3.	Control PID basado en eventos	29
3.1.	Control PID basado en eventos propuesto por Árzén	29
3.2.	Nuevas estrategias de Control PID basado en eventos	31
3.2.1.	Algoritmo 1: Únicamente sin condición de límite de seguridad	32
3.2.2.	Algoritmo 2: Saturación de la ganancia integral	33
3.2.3.	Algoritmo 3: Factor exponencial olvidado del intervalo de muestreo	33
3.2.4.	Algoritmo 4: Estrategia Híbrida	34
3.2.5.	Integral absoluta como índice de desempeño del error	34
4.	Implementación y estudio del desempeño de las estrategias de control PI	37
4.1.	Identificación y control del sistema	38
4.1.1.	Identificación de la función de transferencia de un sistema de primer orden	38
4.1.2.	Sintonización del controlador PI	39
4.1.3.	Implementación en diagrama a bloques del sistema de control PI retroalimentado por red	40
4.1.4.	Implementación en código de la estrategia de control PI basado en tiempo	41
4.1.5.	Implementación en código de las estrategias de control PI basado en eventos	43
4.2.	Estudio de las estrategias de control PI	47
4.2.1.	Análisis caso nominal	48
4.2.1.1.	Estrategia basada en tiempo	48
4.2.1.2.	Estrategia basada en eventos de Árzén	49
4.2.1.3.	Estrategia sin condición de límite de seguridad	58
4.2.1.4.	Estrategia de saturación de la ganancia integral	59
4.2.1.5.	Estrategia del factor exponencial olvidado del intervalo de muestreo	60
4.2.1.6.	Estrategia Híbrida	61
4.2.1.7.	Comparación de la estrategia de control basada en tiempo vs. estrategias basadas en eventos	62
4.2.2.	Análisis de robustez respecto a interferencia en la red y perturbaciones externas	65
5.	Conclusiones	67
I	Apéndice	73
A.	Algunos comandos de TrueTime	75
A.1.	ttAnalogIn(TH)	75
A.2.	ttAnalogOut(TH)	75
A.3.	ttCallBlockSystem(TH)	75
A.4.	ttCreateInterruptHandler(I)	76
A.5.	ttCreateJob(ITH)	76
A.6.	ttCreateMailbox(I)	77
A.7.	ttCreatePeriodicTask(I)	77
A.8.	ttCreatePeriodicTimer(ITH)	77
A.9.	ttCreateTask(I)	78
A.10.	ttGetMsg(TH)	78
A.11.	ttInitKernel(I)	79
A.12.	ttInitNetwork(I)	79

<i>ÍNDICE GENERAL</i>	v
A.13.ttSendMsg(TH)	79
A.14.ttSetKernelParameter(ITH)	80
A.15.ttSetNetworkParameter(ITH)	81
A.16.ttSleepUntil(T)	81
B. Parámetros de batería	83
C. Abreviaturas utilizadas en las gráficas y tablas para el estudio de las estrategias del control PI.	85

Índice de figuras

1.1. Configuración básica del motor de DC.	5
1.2. Representación eléctrica y mecánica del motor DC.	5
1.3. Diagrama de bloques de un sistema de control de cruceo para un automóvil.	6
1.4. Curvas de <i>par</i> para el motor de un automóvil.	7
1.5. Un coche con control de cruceo, que se encuentra en un camino con pendiente.	8
1.6. Máquina de estados finitos para el sistema de control de cruceo.	8
2.1. Componentes de un sistema controlado por computadora.	10
2.2. Sistemas de lazo abierto y lazo cerrado	10
2.3. Características entrada/salida de un control on-off.	11
2.4. Respuesta a cambios en la señal escalón en el valor de referencia en un sistema con controlador(P,PI, PID).	12
2.5. Implementación del controlador PD.	13
2.6. Controlador PID con un filtro derivativo y anti-windup.	14
2.7. Control PID analógico SIM960.	16
2.8. C2 Control de sistemas neumáticos.	16
2.9. BTC-8300 Self-Tune Fuzzy / PID.	17
2.10. Muestreo basado en tiempo vs. Basado en eventos	17
2.11. Aplicación típica del bus CAN en un vehículo.	20
2.12. Estados lógicos del bus CAN.	21
2.13. Velocidad del bus CAN y longitud del bus.	22
2.14. Métodos de terminación del bus.	22
2.15. Estructura estándar de datos.	23
2.16. Estructura extendida de datos.	23
2.17. Ejemplo de arbitraje del bus CAN.	24
2.18. Cuadro de diálogo del bloque núcleo de TRUE TIME.	26
2.19. Cuadro de diálogo del bloque de red TRUE TIME.	27
2.20. Cuadro de diálogo del bloque batería de TRUE TIME.	28
3.1. Control PID basado en eventos de Årzén.	29
3.2. Muestreo en el controlador PID de Årzén.	30
3.3. Descomposición de un intervalo de estado estacionario.	31
3.4. Algoritmo de los controladores PID basados en eventos sin condición de límite de seguridad.	33
3.5. Dinámica de la ganancia integral de las estrategias de control basado en eventos sin condición de límite de seguridad	35
4.1. Vista en el osciloscopio de la repuesta de un motor de DC, al alimentarlo con 12v.	38
4.2. Diagrama a bloques del sistema de control de velocidad del motor de DC.	40

4.3. Diagrama a bloques del control de la velocidad de un motor de DC, sistema lazo cerrado por una red CAN-BUS.	41
4.4. Diagrama a bloques del sub sistema del bloque <i>integral del error</i>	42
4.5. Señales de respuesta de la estrategia de control PI basada en tiempo.	48
4.6. Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=0.04s.	49
4.7. Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=0.04s.	50
4.8. Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1 % del valor de la referencia, y hmax=0.04s.	51
4.9. Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1 % del valor de la referencia, y hmax=0.04s.	52
4.10. Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=1s.	53
4.11. Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=1s.	54
4.12. Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1 % del valor de la referencia, y hmax=1s.	55
4.13. Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1 % del valor de la referencia, y hmax=1s.	56
4.14. Señales de respuesta de la estrategia de control PI basada en eventos, sin condición de límite de seguridad.	58
4.15. Señales de respuesta de la estrategia de control PI basada en eventos, saturación de la ganancia integral.	59
4.16. Señales de respuesta de la estrategia de control PI basada en eventos, factor exponencial olvidado de la ganancia integral.	60
4.17. Señales de respuesta de la estrategia de control PI basada en eventos, estrategia híbrida.	61
4.18. Comparación de ts, Mp, IAE, IAEP, IAD, en las tres mejores estrategias.	63
4.19. Comparación de Menv y Dtb en las tres mejores estrategias.	64
4.20. Respuesta del sistema con las tres mejores estrategias de control PI(basada en tiempo, B. eventos de Árzén, B. eventos híbrida) con el 81 % del ancho de banda del nodo de interferencias, con perturbaciones externas en 4s y 14s.	65
4.21. Índices IAE, IAEP, IAD de las tres mejores estrategias de control PI(basada en tiempo, B. eventos de Árzén, B. eventos híbrida), con el 81 % del ancho de banda del nodo de interferencias, con perturbaciones externas en 4s y 14s.	66

Índice de tablas

2.1. Codificación del campo de control(Nivel dominante D, nivel recesivo R).	24
3.1. Código del controlador PID de Árzén.	30
4.1. Descripción de las variables para implementar la estrategia de control PI basado en tiempo, para controlar la velocidad del motor de DC.	43
4.2. Descripción de las variables exclusivas para implementar las estrategias de control PI basado en eventos , para controlar la velocidad del motor de DC.	47

Agradecimientos

Las primeras líneas quiero dedicarlas a Dios, que siempre a estado conmigo a lo largo de mi vida a través de amigos familiares, maestros. Agradezco la fortaleza y aliento que me dió para poder levantarme después de la etapa difícil de mi vida y así poder concluir con esta tesis. Agradezco su salvación y la oportunidad que me da de conocerle.

Reconozco y agradezco el incondicional apoyo de mi mamá Fracisca Juárez quien a pesar de las circunstancias difíciles que pasamos como familia nunca me negó su apoyo. Me asombro en ver el gozo que Dios ha puesto en su vida a pesar de las dificultades.

Gracias a mi única hermana Elsa Noemí por su tiempo dedicado para escucharme, aconsejarme, y reír conmigo. Y sin olvidar su gran apoyo en explicarme muchos conceptos a lo largo de mis estudios desde la primaria a la universidad.

Gracias a mi papá J.Máximo García por sus sabios consejos que me ha dado respecto a la vida y por las enseñanzas en agronomía.

Gracias a mi asesor Dr. José Fermi Guerrero Castellanos por su grande apoyo en terminar esta tesis. Gracias por la paciencia en explicarme los conceptos esenciales y guiarme en el proceso de esta tesis. Me admiro mucho que siendo un gran investigador de clase mundial no ha olvidado la humildad, y el valor de la amistad, yo he sido testigo de esto, creo que todos sus estudiantes y compañeros de trabajo. Deseo las mas grandes bendiciones de Dios para él, su esposa y su pequeña hija.

Gracias a mi coasesor Dr. Germán Ardul Muñoz Hernández por sus aportaciones importantes hechas a esta tesis. Gracias por los momentos que invirtió para darme clase en mis tiempos de estudiante. Gracias por su amigable carácter.

Gracias a mi asesor externo Dr. Carlos Gracios Marín por su gran apoyo en la etapa inicial de esta tesis y por las recomendaciones hechas en la etapa final.

Gracias por el tiempo invertido en la revisión de esta tesis, a los miembros del comité (Dr. Gerardo Mino Aguilar, Dra. Liliana Cortez Georgievna, M.C. Ricardo Álvarez González), gracias por las críticas constructivas hechas a esta tesis.

Gracia a mi primo Job Lazcano, por ser como un hermano. Sus sabios consejos, ánimo, exhortación han sido de gran valor a mi vida.

Gracias a mis amigos que siempre han estado conmigo dándome consejos, apoyándome, animándome, y también compartiendo su alegría conmigo, ellos han dejado una gran huella en mi vida. Ellos son: Mike, Anita, Dany Rosas, Samuel Rosas, Tania Morales, José Montes, Rebeca Arana, Raul Galicia, Eliut lozano, Iveth Montes, Manuel Rosas, Jocabeth Hernández, Guillermo Rosas, Dorian Dakota, Jared Aguilar, Elisa Martinez, Cesar Flores, Wemssie, Daniel Gámez, Juan Carlos García, Adriana Lazcano, Juan Carlos Luna.

Gracias a mi Abuelito Mariano Juárez por su grande apoyo y consejos. Gracias a mis tíos Pablo Juárez, Justino Juárez, Verónica Juárez, Eva Juárez, Alejandro Juárez, David Juárez por su mano amiga, ánimo y aprecio mostrado.

Gracias a mis Abuelitos Gloria y Rubén Viveros por su amor mostrado a mi familia y en especial a mí. Gracias por sus sabios consejos y su tiempo invertido en todas la etapas de mi vida. Son un reflejo vivo de Dios.

Gracias a mi Abuelita Conchita Romero por su amor y el ánimo que me dio para buscar a

Dios, hoy se encuentra en el cielo gozándose con Dios tal y como ella soñaba. Su ejemplo ha sido mas que mil palabras, sus obras permanecen y se incrementan al pasar el tiempo.

*A quien tengo yo en los cielos si no a ti?
Y fuera de ti nada deseo en la tierra.
Salmo 73:25*

Resumen

La presente tesis aborda el estudio de una clase particular de estrategias de control PID asíncrono, conocido también como control PID basado en eventos o disparado por eventos. En un control basado por eventos es el evento el que define cuando debe calcularse y aplicar el control y no el paso del tiempo como en los controles convencionales. El objetivo es mostrar que este tipo de controladores representan una oportunidad para la implementación de sistemas de control usando sistemas embebidos y cuando el lazo de control es cerrado mediante una red de comunicación. Se muestra que la parte fundamental en el control PID basado en eventos es la parte integral, por lo cual y sin pérdida de generalidad, la tesis se enfoca en el control PI basado en eventos aplicado a un sistema de primer orden. La elección del sistema de primer orden es por su simplicidad sin que eso demerite su importancia en el ámbito científico e ingenieril.

La tesis se conforma de cinco capítulos. En el capítulo uno se da la introducción a esta tesis, donde se describe la relevancia que tienen los sistemas de control basados en redes así como la importancia de los controles asíncronos; se definen los objetivos de esta tesis, también son descritos algunos de los sistemas de primer orden en el entorno científico y tecnológico. En el capítulo dos se describen los conceptos fundamentales del control automático y del control PID digital. Se muestran algunos de los módulos de control PID en la industria, se da una introducción a los sistemas de control PID basado en eventos y se describe el protocolo de comunicación CAN-BUS. Finalmente se describe la herramienta TRUE TIME de MATLAB/SIMULINK, que es de gran utilidad en la simulación de las nuevas estrategias de control usando redes de comunicación. En el capítulo tres se describen las estrategias de control PID basado en eventos, desde la estrategia propuesta por K.E.Árzén en 2008, hasta las cuatro estrategias propuestas por S. Durand en 2011. También son definidos los índices de desempeño (IAE, IAEP, IAD) que son utilizados para evaluar las estrategias de control PID basado en eventos. En el capítulo cuatro se muestra la implementación en TRUE TIME de MATLAB/SIMULINK del sistema de control retroalimentado por una red CAN-BUS. De la respuesta del sistema con cada una de las estrategias de control PID basado en eventos es obtenido: tiempo de asentamiento (t_s) y el sobre impulso máximo (M_p), los índices de desempeño (IAE, IAEP, IAD), la aproximación de la duración de batería (D_{tb}) del nodo control de la red, mensajes enviados en la red (M_{env}). Los resultados del sistema de control con cada una de las estrategias de control PID basado en eventos son comparados con el resultado de la estrategia de control PID basado en tiempo. Mediante tablas y gráficas se obtiene que la estrategia basada en eventos híbrida propuesta por S.Durand es mejor, reduce un 88% la cantidad de mensajes enviados a la red por el nodo control y así aumenta la duración de la batería que suministra el nodo control, se muestra que la estrategia híbrida es la más robusta ante perturbaciones externas y perturbaciones en la red. En el capítulo cinco se dan las conclusiones generales de esta tesis.

Capítulo 1

Introducción

En los últimos años el interés y la investigación de técnicas de control automático utilizando muestreo asíncrono han aumentado considerablemente. La razón es una consecuencia directa del impacto de las redes inalámbricas de sensores (Wireless Sensor Networks (WSN)) y sistemas de control basados en redes (Networked Control Systems (NCS)). Los sistemas de control basados en redes (NCSs) son sistemas de control en los cuales el lazo de control se cierra a través de una red en tiempo real. Las ventajas de este tipo de sistemas son principalmente la arquitectura flexible y la reducción de costo en instalación y mantenimiento. Los NCS son aplicados en un rango muy amplio de sistemas, tales como redes de sensores móviles, cirugía remota, sistemas de automatización de autopistas y en vehículos aéreos no tripulados [3], [4], [5].

Las principales desventajas de los NCSs son los efectos de la red, los cuales influyen en el desempeño y estabilidad del lazo de control. Tales efectos son los retardos y la pérdida de paquetes de información los cuales producen una pérdida de sincronización entre el control principal y los diferentes agentes o nodos. En esta situación, las técnicas de control basadas en eventos [6] representan una línea de investigación prometedora la cual permitirá desarrollar nuevas estrategias de control, donde el intercambio de la información conjuntamente con los demás agentes se realice sólo cuando se produzca un evento [7] y no por el paso del tiempo, como lo marcaría la teoría de control digital [1]. Otra de las principales razones del por qué es necesario utilizar técnicas de control basadas en eventos, son los recursos computacionales y de comunicaciones, esto es, la reducción de intercambio de datos entre sensores, control y actuadores. Las técnicas de control basadas en eventos, reducen la carga computacional de los sistemas embebidos, reducen el ancho de banda utilizado por el nodo de control y extienden el tiempo de vida de las baterías en los sistemas.

A pesar de todos los avances en las técnicas de control en los últimos 50 años, el control PID (Proporcional-Integral-Derivativo) sigue siendo una de las herramientas de control más usada en los sistemas mecatrónicos, ya sea industriales o en sistemas que tienen otros fines como médicos, de investigación, etc. Aun cuando sistemas de control más sofisticados son usados, en la práctica es muy común tener una estructura jerárquica con un control PID en el más bajo nivel. El control PID ha sobrevivido a muchos cambios en la tecnología, desde sistemas mecánicos y neumáticos, pasando por tubos de vacío, transistores y circuitos integrados. Hoy en día el 99% de los controladores PID son basados en microprocesadores [8]. El control PID en la industria ha sido la solución a muchos problemas de control para diferentes sistemas, su campo de aplicación es demasiado amplio. Generalmente lo ocupan en el control de temperatura de hornos y cámaras de conservación, control de flujo de algún fluido o mezcla, etc [9]. Sin embargo, muchos problemas surgen cuando un controlador PID digital diseñado y sintonizado para trabajar con periodo de muestreo constante es colocado en uno de los módulos de una red de comunicación, en lugar de estar trabajando de una forma síncrona comienza a desempeñarse de una manera asíncrona, debido a que la red de comunicación se satura, por consiguiente existe un retraso en la información e incluso pérdida parcial o total de datos.

La red de comunicación CAN (acrónimo del inglés Controller Area Network), fue desarrollada en 1980 por BOSCH (un fabricante de equipo eléctrico en Alemania). Después CAN fue estandarizado como ISO-11898 e ISO-11519, establecido como un protocolo estándar en la red de un automóvil en la industria automotriz. En la actualidad este protocolo es usado en muchas otras áreas de la ingeniería como es la aeroespacial e incluso la industria textil. CAN ofrece un protocolo de comunicación eficiente entre sensores, actuadores, controladores, y otros nodos en aplicaciones en tiempo real, y es conocido por su simplicidad, confiabilidad, y alto funcionamiento [10].

En este trabajo de tesis se propone la implementación en simulación de las cuatro estrategias de control PID (Proporcional-Integral-Derivativo) basado en eventos descritos en [11], [12] para controlar un sistema de primer orden teniendo como lazo de retroalimentación una red de comunicación CAN-BUS. El sistema de control basado en red implementado tiene cuatro nodos conectados a la red CAN-BUS. El primero es el nodo de interferencias, el segundo es el nodo del actuador, el tercero es el nodo de sensor, el cuarto es el nodo de control. El nodo de interferencias envía datos aleatorios a la red, que interfieren en la comunicación entre los demás nodos. El nodo actuador tiene su salida conectada al sistema a analizar, éste se encarga de recibir la señal de control que indica el nodo control y la aplica al sistema. El nodo sensor es el que se encarga de medir la señal a controlar del sistema y la envía al nodo control por la red CAN-BUS. El nodo control es donde se encuentra el control PID basado en eventos; lee la señal de referencia, recibe la señal medida por el nodo sensor, si es detectado un evento se aplica el control PID y entonces la señal de control es enviada al nodo actuador. La idea es mostrar que mediante un controlador PID basado en eventos es posible reducir: el tráfico de datos en la red de comunicaciones, el consumo de energía, costos; todo esto sin sacrificar el desempeño del sistema en lazo cerrado.

El sistema de primer orden en cuestión junto con las estrategias de control propuestas será simulado utilizando la herramienta TrueTime de MATLAB/Simulink que permite simular tanto los sistemas dinámicos, como la propia dinámica de la red [13].

1.1. Objetivos

Objetivo general

- Analizar las cuatro estrategias de control PID basado en eventos utilizando la herramienta TrueTime de MATLAB/Simulink, teniendo como lazo de retroalimentación una red de comunicación CAN.

Objetivos específicos

- Analizar el modelo matemático de diferentes sistemas en la naturaleza que se representan mediante ecuaciones diferenciales o en diferencias de primer orden.
- Estabilizar un sistema de primer orden mediante un controlador PI, de tal forma que el sistema en lazo cerrado cumpla con especificaciones temporales específicas (tiempo de asentamiento, sobrepaso).
- Implementar en simulación, utilizando la herramienta TrueTime de MATLAB/Simulink las cuatro estrategias de control PID basado en eventos con el fin de controlar el sistema de primer orden propuesto, teniendo como lazo de retroalimentación una red de comunicación CAN-BUS.
- Evaluar las cuatro estrategias de control basado en eventos bajo condiciones iniciales idénticas utilizando la integral de la norma del error como índice de desempeño, así como el porcentaje de los mensajes transmitidos por la red y el ahorro de energía en el sistema.

1.2. Los sistemas de primer orden en el entorno científico y tecnológico

Con el fin de mostrar la importancia y la trascendencia del control de sistemas de primer orden, en esta sección se exponen cuatro sistemas que son modelados mediante una ecuación diferencial o en diferencias de primer orden. Se muestran que desde los sistemas informáticos a los sistemas biológicos pasando por los sistemas electromecánicos pueden ser modelados mediante este tipo de ecuaciones. Esto justifica el hecho de que en esta tesis el estudio se enfoque en el control de un sistemas de primer orden, sin embargo el estudio puede extenderse a sistemas de mayor orden.

1.2.1. Servidor de e-mail

El servidor Lotus de IBM es un sistema de software de colaboración que administra los usuarios del correo electrónico, documentos y notas. Las máquinas cliente interactúan con los usuarios finales para proporcionar acceso a los datos y aplicaciones. El servidor también se ocupa de otras tareas administrativas. En el desarrollo temprano del sistema se observó que el rendimiento era pobre cuando la unidad central de procesamiento (CPU) estaba sobrecargada debido a demasiadas solicitudes de servicio y mecanismos para controlar la carga.

La interacción entre el cliente y el servidor es en la forma de llamadas a procedimiento remoto (RPCs). El servidor mantiene un registro de las estadísticas de las solicitudes completadas. El número total de solicitudes que se sirven, llamado *RIS* (RPC en servidor), es también medido. La carga en el servidor es controlado por un parámetro llamado *MaxUsers*, que fija el número total de conexiones de cliente con el servidor. Este parámetro es controlado por el administrador del sistema. El servidor puede ser considerado como un sistema dinámico con *MaxUsers* como la entrada y *RIS* como la salida. La relación entre la entrada y la salida fue investigada por primera vez por la exploración del rendimiento del estado estacionario y se encontró que era lineal.

Un modelo dinámico en la forma de una ecuación diferencial de primer orden es utilizado para capturar el comportamiento dinámico de este sistema [1]. Usando técnicas de identificación, se construye un modelo de la forma:

$$y[k + 1] = ay[k] + bu[k],$$

donde $u = \text{MaxUsers} - \overline{\text{MaxUsers}}$ y $y = \text{RIS} - \overline{\text{RIS}}$. Los parámetros $a=0.43$ y $b=0.47$ describen la dinámica del sistema al rededor de un punto de operación, y $\overline{\text{MaxUsers}} = 165$ y $\overline{\text{RIS}} = 135$ representan el punto de operación nominal del sistema. El número de solicitudes fue el promedio sobre un periodo de muestreo de 60s.

1.2.2. Propagación de onda en las redes neuronales

Las dinámicas del potencial de una membrana en una célula son un mecanismo fundamental en la comprensión de señalización en las células, particularmente en las neuronas y las células musculares. Las ecuaciones de Hodgkin-Huxley dan un modelo simple para el estudio de las ondas de propagación en las redes de neuronas. El modelo de una sola neurona tiene la forma:

$$C \frac{dV}{dt} = -I_{Na} - I_K - I_{leak} + I_{input},$$

donde V es el potencial de la membrana, C es la capacitancia, I_{Na} y I_K son la corriente causada por el transporte de sodio y potasio a través de la membrana de una célula, I_{leak} es la corriente de fuga y I_{input} es la estimulación externa de la célula. Cada corriente obedece la ley de Ohm, es decir:

$$I = g(V - E),$$

Donde g es la conductancia y E es el equilibrio de voltaje. El equilibrio de voltaje es dado por la ley de Nernst,

$$E = \frac{RT}{nF} \Rightarrow \frac{C_e}{C_i},$$

donde R es la constante de Boltzmann, T es la temperatura absoluta, F es la constante de Faraday, n es la carga (o valencia) del ion, C_i y C_e son la concentración de iones dentro de la célula y en el fluido externo. A $20^\circ C$ se tiene $RT/F = 20mV$.

El modelo de Hodgkin-Huxley se desarrolló originalmente como un medio para predecir el comportamiento cuantitativo del calamar gigante. Hodgkin y Huxley compartieron el premio Nobel de Fisiología 1963(junto con J.C.Eccles) para el análisis de los eventos eléctricos y químicos en las descargas de las células nerviosas [1].

1.2.3. Motor de DC

El motor de corriente continua (DC) es una de las primeras máquinas ideadas para convertir la energía eléctrica en energía mecánica. En los últimos años, el uso de máquinas DC está asociado exclusivamente con aplicaciones donde las únicas características del motor DC justifican su costo o donde el equipo portátil debe ejecutarse desde una fuente de alimentación DC [2]. El motor DC se presta fácilmente para controlar su velocidad. Su compatibilidad con los nuevos amplificadores (tiristores y transistores), además de su mayor rendimiento debido a la disponibilidad de nuevos materiales mejorados en los imanes, las escobillas y epóxidos también han revivido el interés en las máquinas de corriente continua.

La reciente evolución de los microprocesadores, materiales magnéticos, tecnología de semiconductores y mecatrónica proporcionan una amplia gama de aplicaciones para el alto desempeño de los motores eléctricos en diversos procesos industriales. Para un alto desempeño en aplicaciones de manejo como la robótica, molinos rodantes, herramientas máquina, etc, la velocidad exacta y el control de posición son de gran importancia. Los motores DC son ampliamente utilizados en estas aplicaciones debido a su fiabilidad y facilidad de control debido a la desacoplada naturaleza del campo y las fuerzas con motivo del magneto de armadura. Los dos tipos de motores de DC comúnmente utilizados (imán de excitación independiente y permanente (PM)), el motor DC de imán permanente tiene la ventaja de que no requiere ningún suministro DC adicional para el campo, como el propio imán permanente actúa como la fuente del flujo. El motor de imán permanente es así compacto en tamaño, robusto y muy eficiente.

Un sistema manejador (*Drive*) consiste de un motor, un convertidor y un controlador, integrados para desarrollar una maniobra mecánica precisa. Los manejadores de motor de DC son usados en muchos procesos industriales, robótica, molinos(papel, pulpa, acero), cintas transportadoras y otras aplicaciones de control de velocidad/posición precisas [2]. Han sido empleados varios tipos de métodos de control bien establecidos.

Aplicaciones complejas tal como robots y manipuladores no requieren únicamente el control de posición, sino también control de seguimiento ó trayectoria [2]. Una trayectoria es una historia en el tiempo de la posición y/o velocidad que el efector final de un manipulador debe seguir a todos los puntos en el tiempo. Por eso el motor en el manipulador tiene que seguir a una velocidad predeterminada o seguimiento de posición durante cambio de marcha y velocidad.

Descripción física

Un imán permanente en el motor de DC es un mecanismo que convierte la energía eléctrica en energía mecánica por acoplamiento magnético. La energía eléctrica es provista por una fuente de voltaje, mientras que la energía mecánica es provista por un rotor de giro.

Un motor de DC básicamente está construido de dos componentes principales: El rotor ó armadura y el estator. La armadura gira dentro del armazón del estator fijo. Una ilustración simple del motor DC se muestra en la figura 1.1.

El estator consiste de imanes permanentes que crean un campo magnético. La armadura consiste de un electroimán creado por una bobina enrollada al rededor de un núcleo de hierro. La armadura gira debido al fenómeno de atracción y oposición de fuerzas de los dos campos magnéticos. Un campo magnético es generado mediante la armadura por el envío de corriente eléctrica por la

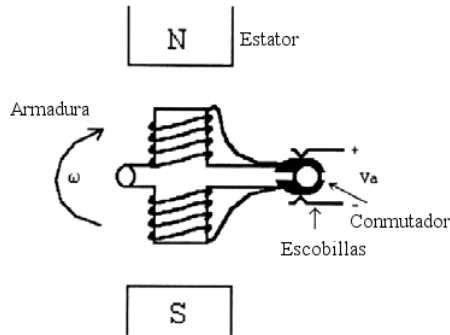


Figura 1.1: Configuración básica del motor de DC.

bobina y la polaridad es constantemente cambiada por alternar la corriente a través de la bobina (también llamada conmutación) causando el giro de la bobina[2].

El conmutador es hecho de dos segmentos semicirculares de cobre montados en el eje al final del rotor. Cada terminal de la bobina del rotor es conectada a un segmento de cobre. Las escobillas fijas son montadas en los segmentos de cobre con que la bobina del rotor es conectada a una fuente de voltaje DC fija por un contacto cercano sin fricción.

El circuito eléctrico equivalente del motor DC se ilustra en figura 1.2. Se puede representar por una fuente de voltaje (V_a) a través de la bobina de la armadura. La equivalencia eléctrica de la bobina de la bobina de la armadura se puede describir por una inductancia (L_a) en serie con una resistencia (R) en serie con el voltaje inducido (V_c) que se opone a la fuente de voltaje. El voltaje inducido es generado por la rotación de la bobina eléctrica a través de las líneas de flujo fijas del imán permanente. Muchas veces este voltaje es referido como fuerza electromotriz atrasada.

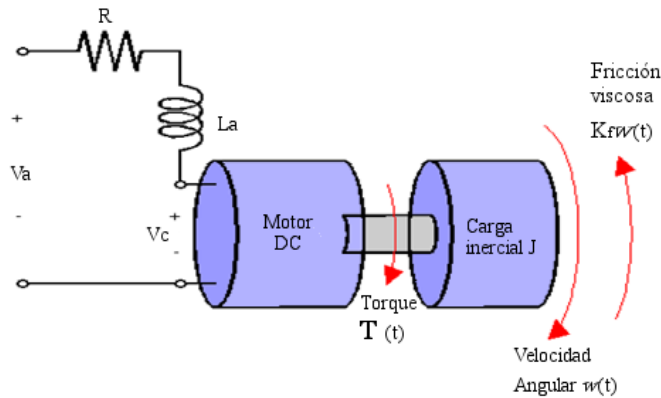


Figura 1.2: Representación eléctrica y mecánica del motor DC.

Del diagrama de la figura 1.2, se describe la dinámica del motor DC representada en la ecuación 1.1 de primer orden, se considera que $L_a \approx 0$, también se excluye la fricción viscosa [2].

$$\dot{\omega} = -\frac{K_m^2}{J_m R} \omega + \frac{K_m}{J_m R} u \quad (1.1)$$

Donde ω es la velocidad del motor, u es la señal de control, K_m es la constante de par motor, R es la resistencia de la armadura, J_m es la inercia del rotor.

1.2.4. Dinámica longitudinal de un automóvil

La dinámica longitudinal de un automóvil es conocida como control de crucero, es un sistema retroalimentado que comúnmente es encontrado en la vida cotidiana. El sistema intenta mantener una velocidad constante de un automóvil en presencia de trastornos causados principalmente por los cambios en la pendiente de una carretera. El controlador compensa estas incógnitas mediante la medición de la velocidad del coche y ajustando el acelerador adecuadamente.

Para modelar el sistema se comienza con el diagrama de bloques de la Figura 1.3. Sea v la velocidad del coche y la v_r la velocidad deseada (referencia). El controlador, que normalmente es del tipo proporcional integral (PI) recibe las señales v y v_r , y genera una señal de control u que se envía a un actuador que controla la posición del acelerador (figura 1.5b). El acelerador a su vez controla el par T entregado por el motor, que se transmite a través de los engranajes y las ruedas, genera la fuerza F para mover el coche. Hay fuerzas de perturbación F_d debido a las variaciones en la pendiente de la carretera, la resistencia a rodar y las fuerzas aerodinámicas. El controlador de crucero también cuenta con una interfaz hombre-máquina que permite al conductor establecer y modificar la velocidad deseada. También hay funciones que desconectan el control de crucero cuando se toca el freno.

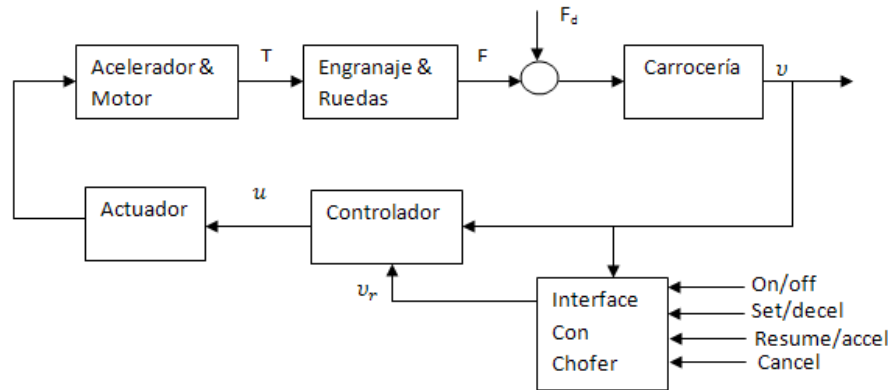


Figura 1.3: Diagrama de bloques de un sistema de control de crucero para un automóvil.

El sistema tiene muchos componentes individuales - actuador, motor, transmisión, ruedas y cuerpo del coche - un modelo detallado puede ser muy complicado. A pesar de esto, el modelo requerido para diseñar el controlador de crucero puede ser muy simple.

Para desarrollar un modelo matemático se empieza con un balance de fuerzas para la carrocería del vehículo. Sea v la velocidad del coche, m la masa total (incluidos los pasajeros), F la fuerza generada por el contacto de las ruedas con el camino, y F_d la fuerza de perturbación debida a la gravedad, la fricción y la resistencia aerodinámica. La ecuación de movimiento del automóvil es simplemente:

$$m \frac{dv}{dt} = F - F_d \quad (1.2)$$

La fuerza F es generada por el motor, cuyo par es proporcional a la velocidad de inyección de combustible, que es en sí misma proporcional a una señal de control $0 \leq u \leq 1$ que controla la posición del acelerador. El par también depende de la velocidad del motor (ω). Una simple representación del par motor a máxima aceleración está dada por la curva de par:

$$T(\omega) = T_m \left(1 - \beta \left(\frac{\omega}{\omega_m} - 1 \right)^2 \right), \quad (1.3)$$

CAPÍTULO 1. INTRODUCCIÓN

1.2. LOS SISTEMAS DE PRIMER ORDEN EN EL ENTORNO CIENTÍFICO Y TECNOLÓGICO

Donde se obtiene el máximo torque T_m de la velocidad del motor ω_m . Los parámetros típicos son $T_m = 190Nm$, $\omega_m = 420rad/s$ (aproximadamente 4000 RPM) y $\beta=0.4$. Sea n la relación de transmisión y r el radio de la rueda. La velocidad del motor está relacionada con la velocidad a través de la expresión:

$$\omega = \frac{n}{r}v =: \alpha_n v, \quad (1.4)$$

La fuerza de impulso puede ser descrita como:

$$F = \frac{nu}{r}T(\omega) = \alpha_n u T(\alpha_n v) \quad (1.5)$$

Los valores típicos de α_n para engranes del 1 al 5 son $\alpha_1 = 40$, $\alpha_2 = 25$, $\alpha_3 = 16$, $\alpha_4 = 12$ y $\alpha_5 = 10$. La inversa de α_n tiene una interpretación física como el radio efectivo de la rueda. En la figura 1.4, la gráfica de la izquierda muestra el par generado por el motor como una función de la velocidad angular del motor, la figura de la derecha muestra que el efecto del engrane es para *aplanar* la curva del par de manera que un torque casi completo se puede obtener casi en todo el rango de velocidad.

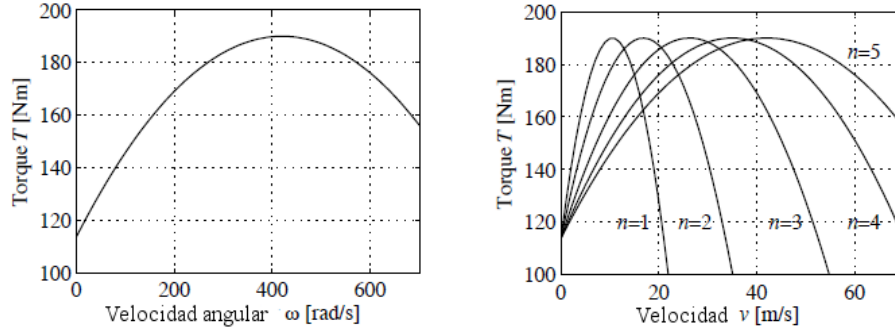


Figura 1.4: Curvas de *par* para el motor de un automóvil.

La fuerza de perturbación F_d tiene tres componentes principales: F_g , las fuerzas debidas a la gravedad; F_r , las fuerzas debido a la fricción al rodar; y F_a , la resistencia aerodinámica. La pendiente del camino es θ , la fuerza de gravedad $F_g = mg \sin \theta$, como se ilustra en la figura 1.5a, donde $g=9.8m/s^2$ es la constante de gravitación. Un modelo simple de fricción de rodadura es:

$$F_r = mgC_r sgn(v),$$

donde C_r es el coeficiente de fricción de rodadura y $sgn(v)$ es el signo de $v(\pm 1)$ o cero si $v = 0$. Un valor típico para el coeficiente de fricción de rodadura es $C_r=0.01$. Por último, la resistencia aerodinámica es proporcional al cuadrado de la velocidad:

$$F_a = \frac{1}{2}\rho C_d A v^2,$$

donde ρ es la densidad del aire, C_d es el coeficiente de arrastre aerodinámico de forma-dependiente y A es el área frontal de los parámetros del automóvil. Los parámetros típicos son $\rho=1.3kg/m^3$, $C_d=0.32$ y $A=2.4m^2$.

Resumiendo, se encuentra que el coche puede ser modelado por

$$m \frac{dv}{dt} = \alpha_n u T(\alpha_n v) - mgC_r sgn(v) - \frac{1}{2}\rho C_d A v^2 - mg \sin \theta, \quad (1.6)$$

donde la función T viene dada por la ecuación 1.3. El modelo 1.6 es un sistema dinámico de primer orden. El estado es la velocidad v de un automóvil, que es también la salida. La entrada es la señal u que controla la posición del acelerador, y la perturbación es la fuerza F_d , que depende

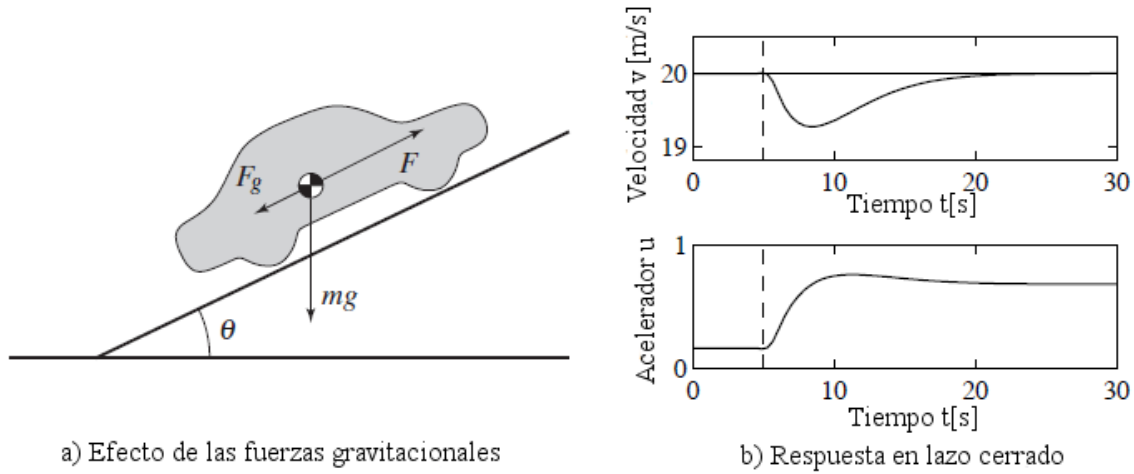


Figura 1.5: Un coche con control de crucero, que se encuentra en un camino con pendiente.

de la pendiente de la carretera. El sistema es no lineal por que de la curva de par, el término de gravedad y el carácter no lineal de la fricción de rodadura y la resistencia aerodinámica. También puede haber variaciones en los parámetros, por ejemplo, la masa del coche depende del número de pasajeros y la carga que se transporta en el coche.

El sistema de control de crucero también cuenta con una interfaz hombre-máquina que permite al conductor para comunicarse con el sistema. Hay muchas formas para implementar este sistema, el sistema ilustrado en la figura 1.6a tiene cuatro botones :on-off, ajuste/desacelerar, reanudar/acelerar y cancelar. El funcionamiento del sistema se rige por una máquina de estados finitos mostrado en la figura 1.6b, controla los modos del controlador PI y el generador de referencia; la transición entre los modos es controlado presionando uno de los cinco botones de la interfaz de control de crucero: encendido, apagado, ajuste, reanudar o cancelar.

El uso del control en sistemas automotrices va mucho más allá de la simple sistema de control de crucero descrito anteriormente. Las aplicaciones incluyen el control de las emisiones, control de tracción, control de potencia (especialmente en los vehículos híbridos) y el control de crucero adaptativo [1].

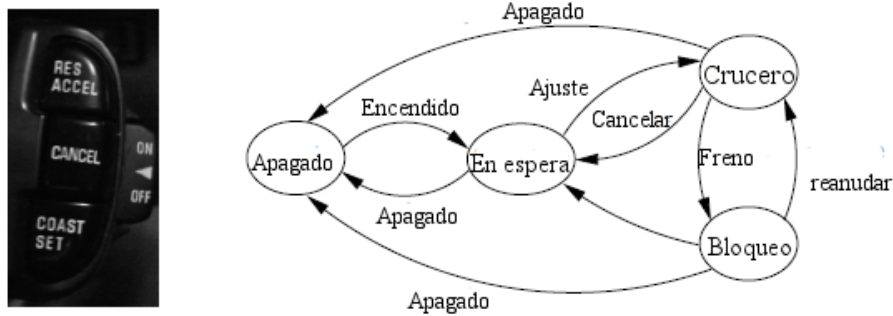


Figura 1.6: Máquina de estados finitos para el sistema de control de crucero.

Capítulo 2

Contexto y Motivaciones

En este capítulo se muestra el contexto y las motivaciones para el desarrollo de esta tesis. Al inicio se define el concepto de control automático, la retroalimentación y se describe el control On-off. Luego se dan las generalidades del controlador PID, y así se describen los efectos de la acción proporcional, integral y derivativa. También se describe el integrador Windup y la implementación en computadora del controlador PID. Se muestran unos de los controladores PID industriales.

Se describen las generalidades del muestreo basado en tiempo vs. muestreo basado en eventos, muestreo basado en eventos como una oportunidad de sistemas embebidos. Se detallan las dificultades al conjugar el control asíncrono con paradigmas bien establecidos. Se describe el protocolo CAN-BUS y su estructura de datos. Finalmente se abordan los conceptos más importantes para el desarrollo de esta tesis del simulador TRUE TIME de MATLAB/Simulink.

2.1. Formas simples de retroalimentación

2.1.1. Control automático

El término control tiene varios significados entre las comunidades de ingeniería. Un control moderno inspecciona la operación de un sistema, compara éste con el comportamiento deseado, calcula las acciones correctivas basadas en un modelo de la respuesta del sistema a entradas externas y actúa el sistema hasta que tenga el efecto del comportamiento deseado. Este lazo básico de retroalimentación, detección, cálculo y actuación es el concepto central del control. La clave formulada en el diseño lógico de control es asegurar que la dinámica del sistema en lazo cerrado sea estable (al limitar las perturbaciones el error también es limitado) y que tenga un comportamiento adicional deseado (buena atenuación de perturbaciones, rápida sensibilidad a cambios en puntos de operación, etc). Estas propiedades son establecidas usando una variación del modelado y técnicas de análisis que capturan la dinámica esencial del sistema y permiten la exploración de posibles comportamientos en la presencia de incertidumbre, ruido y el fallo de un componente. Un ejemplo típico de un sistema de control se muestra en la figura 2.1. Los elementos básicos de inteligencia, cálculo y actuación se ven claramente. En los sistemas de control moderno, el cálculo se implementa típicamente en computadoras digitales, requiriendo el uso de convertidores analógico digital(A/D) y digital analógico (D/A). El algoritmo que calcula la acción de control como una función de los valores del sensor comúnmente se llama ley de control. El sistema puede ser influenciado externamente por un operador que introduce órdenes al sistema [1].

2.1.2. Retroalimentación

Un sistema dinámico es el que cambia su comportamiento durante el tiempo, el término retroalimentación se refiere a una situación en que dos o mas sistemas dinámicos son conectados juntos,

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES
2.1. FORMAS SIMPLES DE RETROALIMENTACIÓN

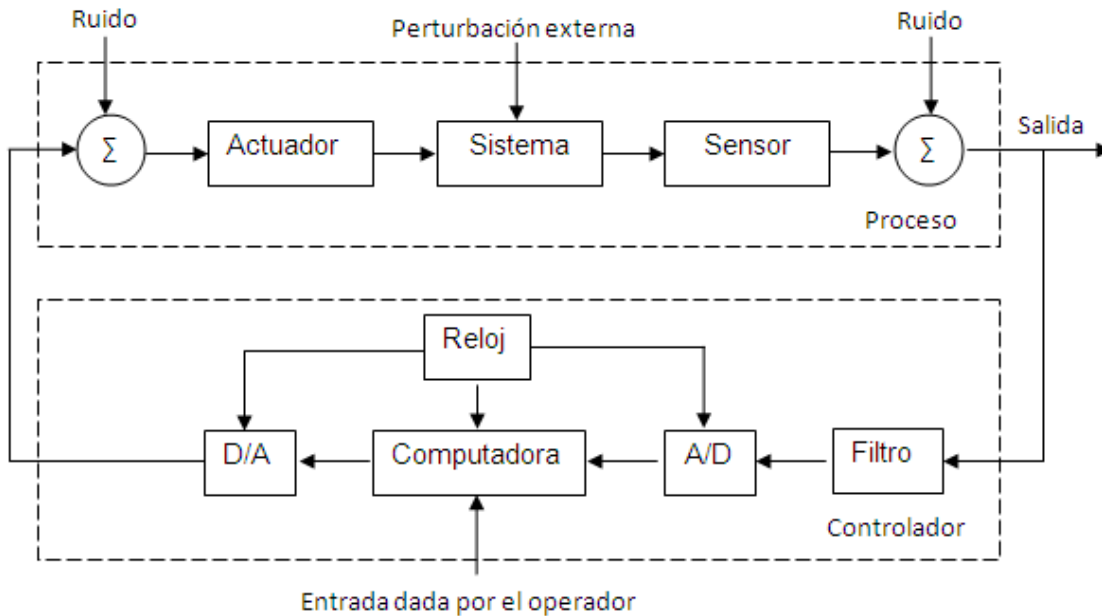


Figura 2.1: Componentes de un sistema controlado por computadora.

tal que cada sistema influencia al otro y sus dinámicas son de esta manera acopladas fuertemente. Se dice que un sistema es de lazo cerrado si el sistema está interconectado en un ciclo, como se muestra en la figura 2.2a. Si se rota la interconexión, ahora se trata de un sistema de lazo abierto, como se muestra en la figura 2.2b [1].

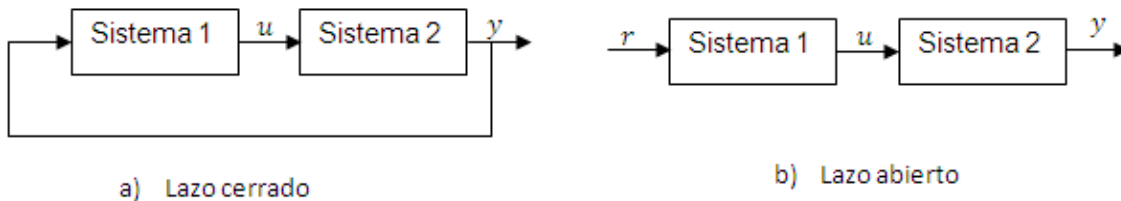


Figura 2.2: Sistemas de lazo abierto a) y lazo cerrado b).

Los sistemas de retroalimentación están en sistemas naturales y de ingeniería.

Los sistemas de control mantienen el medio ambiente, iluminación y luz eléctrica en construcciones y fábricas, regulan la operación de los automóviles, consumidores electrónicos y procesos de manufactura, activan la transportación y comunicación de sistemas, son elementos de alta importancia en sistemas militares y espaciales [1].

La retroalimentación es una característica central de la vida. El proceso de la retroalimentación gobierna como nos desarrollamos, responde a tensión y discusión; y regula factores tal como la temperatura del cuerpo, presión de la sangre y nivel de colesterol. El mecanismo opera en cada nivel, la interacción de proteínas en las células y la interacción de organismos en ecología compleja [14].

La retroalimentación tiene una gran desventaja. Puede crear dinámicas inestables en un sistema, causando oscilaciones y comportamientos desconocidos. Otro inconveniente, especialmente en sistemas de ingeniería, es que la retroalimentación puede introducir ruidos no deseados al sen-

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES

2.1. FORMAS SIMPLES DE RETROALIMENTACIÓN

sor , requiriendo un cuidadoso filtrado de señales. Es por esta razón que una porción substancial del estudio de sistemas de retroalimentación es dedicado a desarrollar un conocimiento pleno de dinámicas y dominio en técnicas de sistemas dinámicos [1].

2.1.3. Control On- Off

La idea de retroalimentación para hacer acciones correctivas basadas en la diferencia entre los valores deseados y el actual de una cantidad puede ser implementado de muchas formas. El beneficio de retroalimentar puede ser obtenido por leyes de retroalimentación muy simples, tal como control on-off, control proporcional, control proporcional-integral-derivativo. Un mecanismo simple de retroalimentación se puede describir de la siguiente manera:

$$u = \begin{cases} u_{max}, & \text{Si } e > 0 \\ u_{min}, & \text{Si } e < 0 \end{cases} \quad (2.1)$$

Donde el error de control $e = r - y$ es la diferencia entre la señal de referencia (o señal ordenada) r y la salida del sistema y , y u es la señal de control. La figura 2.3 muestra la relación entre error y control. Esta ley de control implica la máxima acción correctiva se usa siempre [1].

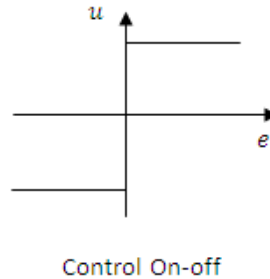


Figura 2.3: Características entrada/salida de un control on-off.

La retroalimentación en la ecuación 2.1 es llamada control on-off. Una de las principales ventajas es que es simple y no tiene parámetros que elegir. El control on-off muchas veces tiene éxito en guardar las variables en proceso cerradas a la referencia, tal como el uso de un simple termostato para mantener la temperatura de un cuarto. Típicamente resulta en un sistema donde las variables de control oscilan, que es a menudo aceptable si la oscilación es suficientemente pequeña. Se observa que en la ecuación 2.1 la variable de control no está definida donde el error es cero [1].

2.1.4. Generalidades del controlador PID

El algoritmo del controlador PID que generalmente se describe en los libros es:

$$u(t) = k_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.2)$$

Donde y es la variable del proceso medido, r ó y_{sp} es la referencia de la variable, u es la señal de control y e es el error de control ($e = r - y$). La variable de referencia es a menudo llamada señal de referencia (set point). La señal de control es así una suma de los tres términos: el término P (es proporcional al error), el término I (es proporcional a la integral del error), y el término D (es proporcional a la derivada del error). Los parámetros del controlador son proporcionales a la ganancia k_p , tiempo integral T_i , y tiempo derivativo T_d . La parte integral pueden ser interpretada como acción de control basada en el pasado. La parte derivativa puede ser interpretada como predicción [15].

2.1.5. Efectos de la acción proporcional, integral, derivativa, del controlador PID

Se desarrolla una intuición de lo que es la respuesta a cambios en la señal escalón en el valor de referencia en un sistema con un controlador proporcional P (a), controlador PI (b) y controlador PID (c). El proceso tiene una función de transferencia $P(S) = 1/(s + 1)^3$, los resultados se muestran en la figura 2.4. Con ganancias $K_p = 1, 2y5$, el error en estado estacionario es 0.5, 0.33 y 0.17. El error decrece con el aumento de la ganancia, pero el sistema también será más oscilatorio. Se observa en la figura que el valor inicial de la señal del controlador igualará a la ganancia del controlador [1].

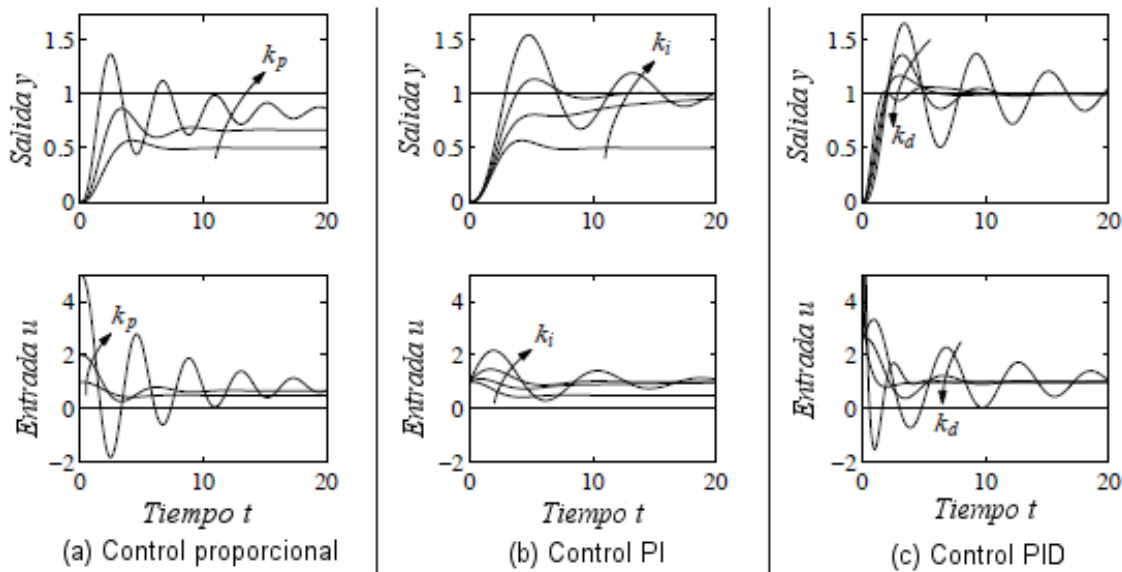


Figura 2.4: Respuesta a cambios en la señal escalón en el valor de referencia en un sistema con controlador(P,PI, PID).

La acción integral no requiere un conocimiento preciso de los parámetros del proceso, ésta alcanza un cero en el error de estado estacionario. La propiedades de la acción integral están ilustradas en la figura 2.4b para una entrada escalón unitario. La ganancia proporcional es constante, $k_p = 1$, y las ganancia integrales son $k_i = 0, 0.2, 0.5$ y 1. El caso $k_i = 0$ corresponde a un control puramente proporcional, con error de estado estacionario de 50%. El error de estado estacionario es eliminado cuando la acción integral es usada. La respuesta se arrastra lentamente hacia la referencia para valores pequeños de k_i y llega más rápido con valores grandes de ganancia integral, pero el sistema llega a ser más oscilatorio debido a su pobre robustez [1].

El efecto del término derivativo k_d como retroalimentación en un controlador PID es proveer la acción predictiva o anticipatoria. La combinación de los términos proporcional y derivativa puede ser escrita como en la ecuación 2.3.

$$u = k_p e + k_d \frac{de}{dt} = k_p \left(e + T_d \frac{de}{dt} \right) = k_p e_p \quad (2.3)$$

Donde e_p puede ser interpretada como una predicción del error al tiempo $t + T_d$ por extrapolación lineal. La predicción de tiempo $T_d = k_d/k_p$ es la constante de tiempo derivativo del controlador.

La acción derivativa puede ser implementada de tomar la diferencia entre la señal y esto es un filtro pasa bajas como se muestra en la figura 2.5.

La función de transferencia para el sistema es:

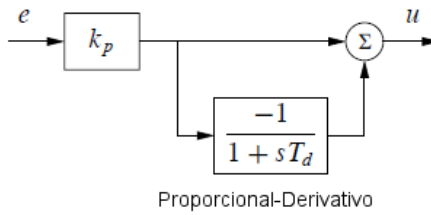


Figura 2.5: Implementación del controlador PD.

$$G_{ue}(s) = k_p \left(1 - \frac{1}{1 + sT_d} \right) = k_p \frac{sT_d}{1 + sT_d} \quad (2.4)$$

La función de transferencia del sistema queda $G(s) = \frac{sT_d}{1+sT_d}$, que aproxima un derivativo a frecuencias bajas ($|s| < T_d$).

La figura 2.4c ilustra el efecto de la acción derivativa: el sistema es oscilatorio cuando la acción derivativa no es usada, y llega mas amortiguado cuando la ganancia derivativa es incrementada. Provoca deterioros con una acción derivativa muy alta [1].

2.1.6. Integrador Windup

Existen limitaciones en los actuadores de un sistema lo que hace que en vez que éstos sean lineales, tienen fenómenos no lineales, unos ejemplos de limitaciones en actuadores es: un motor tiene una velocidad limitada, una válvula no puede ser más que completamente abierta o completamente cerrada. Para un sistema que opera sobre un ancho condicionado de rango, podría pasar que la variable de control alcance los límites del actuador. Cuando esto pasa, el lazo de retroalimentación se rompe y el sistema corre en lazo abierto porque el actuador permanece a este límite independientemente de la salida del proceso tan largo como el actuador permanezca saturado. El término integral se construirá desde que el error es típicamente no nulo. El término integral y la salida del controlador podrían entonces llegar a ser muy largos. La señal de control entonces permanecerá saturada cuando el error cambie, y esto podría tomar un tiempo largo antes que el integrador y la salida del controlador vengán dentro del rango de saturación. La consecuencia es un transitorio largo [1].

Uno de los métodos para eliminar el windup está en la figura 2.6: El sistema tiene una trayectoria de retroalimentación extra que es generada de la medición actual de la salida del actuador, o la salida de un modelo matemático de la saturación del actuador, y forma una señal de error $e_{sat} = u - v$ como la diferencia entre la salida del controlador v y la salida del actuador u . La señal e_{sat} es alimentada a la entrada de la ganancia directa del integrador k_i . La señal e_{sat} es cero cuando allí no hay saturación y el lazo extra de retroalimentación no tiene efecto en el sistema. Cuando el actuador se satura la señal e_{sat} es retroalimentada al integrador en tal forma que e_{sat} va a cero. Esto implica que la salida del controlador sea guardada cerrada al límite de saturación. La salida del controlador cambiará tan pronto como la señal de error cambie y la integral windup es eliminada [1].

2.1.7. Implementación en computadora

La computadora opera periódicamente con la señal que muestrea el sensor, que la convierte en forma digital por medio del convertidor A/D, calcula la señal de control y entonces la salida es convertida a forma analógica para los actuadores. La secuencia de operación es la siguiente:

1. Espera por una interrupción de reloj.
2. Lee la entrada del sensor.

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES
2.1. FORMAS SIMPLES DE RETROALIMENTACIÓN

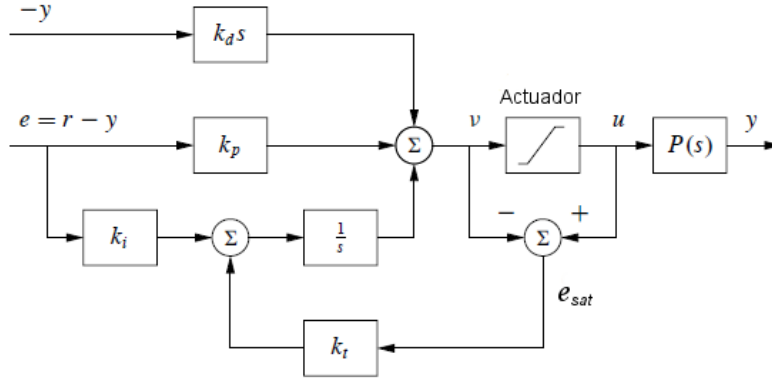


Figura 2.6: Controlador PID con un filtro derivativo y anti-windup.

3. Calcula la señal de control.
4. Envía la salida al actuador.
5. Actualiza las variables del controlador.
6. Repite el proceso.

Se observa que la salida es enviada al actuador tan pronto como esté disponible. El tiempo de retardo es minimizado por hacer los cálculos en el paso 3 tan cortos como sea posible y realizar todas las actualizaciones después que la salida es comandada. Ésta es una forma simple de reducir la latencia, desafortunadamente rara vez es usada en sistemas comerciales [1].

La ilustración del controlador PID está en la figura 2.6, que tiene un filtro derivativo, referencia de ponderación y protección contra integración windup. El controlador es un sistema dinámico en tiempo continuo. Para que sea implementado en una computadora se tiene que pasar el sistema continuo a un sistema en tiempo discreto [1].

En base a la figura 2.6, la señal v es la suma de los términos proporcional, integral y derivativo, la salida del controlador es $u = sat(v)$, donde sat es la función de saturación que modela el actuador. La implementación del término proporcional $k_p(\beta(r - y))$ es simple solo se reemplazan las variables continuas con su versión muestreada [1]. Por lo tanto:

$$P(t_k) = k_p(\beta(r(t_k) - y(t_k))) \quad (2.5)$$

Donde t_k denota el instante de muestreo que es el tiempo donde la computadora lee una entrada. El operador da h_{nom} que representa el periodo de muestreo, así que $t_{k+1} = t_k + h_{nom}$. El término integral es obtenido aproximando la integral con la suma.

$$I(t_{k+1}) = I(t_k) + k_i h_{nom} e(t_k) + \frac{h_{nom}}{T_t} (sat(v) - v) \quad (2.6)$$

Donde $T_t = h_{nom}/k_t$ representa el término anti windup. El término filtro derivativo D es dado por la ecuación diferencial

$$T_f \frac{dD}{dt} + D = -k_d y$$

Aproximando la parte derivativa con diferenciación de atraso se obtiene:

$$T_f \frac{D(t_k) - D(t_{k-1})}{h_{nom}} + D(t_k) = -k_d \frac{y(t_k) - y(t_{k-1})}{h_{nom}}$$

Que finalmente es escrita la parte derivativa como:

$$D(t_k) = \frac{T_f}{T_f + h_{nom}} D(t_k - 1) - \frac{k_d}{T_f + h_{nom}} (y(t_k) - y(t_k - 1)) \quad (2.7)$$

La ventaja de usar diferenciación de atraso es que el parámetro $T_f/(T_f + h_{nom})$ no es negativo y es menor que 1 para todo $h_{nom} > 0$, que garantiza que la ecuación diferencial es estable. Reor-

ganizando las ecuaciones 2.5-2.7, el controlador PID puede ser descrito con el siguiente pseudocódigo.

```
**Pre cálculo de coeficientes del controlador**  
 $bi = ki * h_{nom}$   
 $ad = Tf / (Tf + h_{nom})$   
 $bd = kd / (Tf + h_{nom})$   
 $br = h_{nom} / Tt$   
**Algoritmo de control del lazo principal**  
while(running)  
 $r = adin(ch1)$  **Lee la señal de referencia del canal ch1**  
 $y = adin(ch2)$  ** Lee la variable del proceso en el canal ch2**  
 $P = kp * (b * r - y)$  ** Calcula la parte proporcional.**  
 $D = ad * D - bd * (y - yold)$  ** Actualiza la parte derivativa**  
 $v = P + I + D$  ** Calcula la salida temporal**  
 $u = sat(v, ulow, uhigh)$  **Simula la saturación del actuador**  
 $daout(ch1)$  **Ajusta la salida analógica ch1**  
 $I = I + bi * (r - y) + br * (u - v)$  ** Actualiza la parte integral**  
 $yold = y$  **Actualiza la salida del proceso anterior**  
sleep( $h_{nom}$ ) **Espera hasta el siguiente intervalo de actualización**
```

Pre calculo de los coeficientes bi , ad , bd y br guarda el tiempo de computadora en el lazo principal. Estos cálculos tienen que ser efectuados únicamente cuando los parámetros del controlador han cambiado. El lazo principal es ejecutado una vez en todo el periodo de muestreo. El programa tiene 3 estados: $yold$, I , y D . Una variable de estado puede ser eliminada al costo de un código menos legible. Se observa que el código calcula el filtro derivativo del proceso de salida, que tiene referencia de ponderación y protección anti windup [1].

2.2. Aplicaciones del PID en la Industria

El control PID en la industria ha sido la solución a muchos problemas de control para diferentes sistemas, su campo de aplicación es demasiado amplio. Generalmente lo ocupan en el control de temperatura de hornos y cámaras de conservación, control de flujo de fluidos ó mezclas, etc. El control PID es muy eficiente para cada tarea de control que se realiza en la industria, otra de las razones por la cual el control PID es usado frecuentemente en la industria es por que en el mercado industrial existen muchos tipos, con diferentes características técnicas, con la robustez necesaria para controlar cualquier proceso industrial, y con la facilidad necesaria para que el usuario pueda manipular las ganancias del control PID. Se encuentran controles PID que son exclusivamente para controlar temperatura, sistemas neumáticos [16][17]. Para un mejor desempeño en las aplicaciones industriales se desarrolló el control de Lógica Fuzzy mas el controlador PID basado en microprocesador[18].

2.2.1. Control PID analógico SIM960

Es un instrumento proyectado para la mayoría de las aplicaciones de control, se observa en la Figura 2.7. Este combina la señal analógica manipulada con un control digital, dando lo mejor de ambos. Tiene un ancho de banda alto (100kHz), su lazo de control podría ser implementado sin tiempo discreto. Tiene un mejor desempeño en aplicaciones de sensibilidad al ruido, incluye láser de poder y estabilización de longitud de onda, examina microscópicamente. La ganancia ajustable por el usuario es arriba de 1000x significa gran flexibilidad, reduciendo la necesidad de pre amplificación de la entrada [16].

El panel frontal de control habilita fácilmente la modificación de parámetros del sistema y monitorea la entrada y salida de señales. La energía y comunicación serial son por un conector D-sub

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES

2.2. APLICACIONES DEL PID EN LA INDUSTRIA

de 15-pin. Todos los parámetros de los instrumentos pueden ser cargados y consultados por la interface serial. Amplifica la señal de error y la salida puede ser monitoreada con un despliegue en una barra de leds, la resolución numérica se despliega. Las señales de entrada y la señal de referencia pueden ser monitoreadas en la pantalla numérica [16].



Figura 2.7: Control PID analógico SIM960.

2.2.2. C2 Control de sistemas neumáticos

El C2 es un control analógico avanzado, para válvulas proporcionales y motores de tipo bobina, se muestra en la figura 2.8. El C2 tiene una fundación analógica para proveer tiempo real, alto control de fidelidad. Para el despliegue digital usa una interface con LCD y botón-empuje que provee un fácil control al usuario. El sistema también tiene provisiones para configuraciones especiales de fábrica para comunicarse con redes industriales y digitales [17].



Figura 2.8: C2 Control de sistemas neumáticos.

2.2.3. BTC-8300 Self-Tune Fuzzy / PID

BTC-8300 Lógica difusa mas controlador PID basado en microprocesador que se muestra en la figura 2.9, incorpora algo brillante, fácil para leer 4 dígitos en la pantalla de leds, indicando el valor del proceso. La tecnología lógica difusa activa un proceso para alcanzar un punto de referencia pre determinado en el tiempo más corto, con el mínimo de sobre tiro durante alta de energía o carga externa de perturbaciones. La unidades características son tres teclas de pulsación para

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES

2.3. MUESTREO BASADO EN TIEMPO VS. MUESTREO BASADO EN EVENTOS

seleccionar varios controles y entrada de parámetros. Como una opción adicional están disponibles la comunicación digital RS-485 y RS-232. Estas opciones permiten al BTC-8300 ser integrado con un sistema de control supervisor y software [18].



Figura 2.9: BTC-8300 Self-Tune Fuzzy / PID.

2.3. Muestreo basado en tiempo vs. Muestreo basado en Eventos

La estructura clásica en tiempo discreto de los sistemas de control, consiste en muestrear el sistema uniformemente en el tiempo, con un periodo constante de muestreo h_{nom} , calcular y actualizar la ley de control cada instante de tiempo $t_k = k \cdot h_{nom}$, donde $k \in \mathbb{N}$. Este principio, denota el caso de accionamiento por tiempo (o el caso sincrónico, todas las mediciones de señal son sincrónicas), se describe en la figura 2.10 (a). El campo ha sido ampliamente investigado [19], en el caso de fluctuación de muestreo o pérdida de medición con perturbaciones temporales puede ser visto como asíncrono. Sin embargo existen algunos trabajos enfocados más recientemente en el muestreo basado en eventos, donde los intervalos de muestreo son manipulados eventualmente. La señal medida es muestreada en el instante donde la salida cambia y cruza el nivel dado $q_j = j \cdot q_{nom}$, donde $j \in \mathbb{Z}$. Este principio, denota el caso de accionamiento por eventos (o el caso asíncrono en comparación con el primer caso), esto se muestra en la figura 2.10 (b).

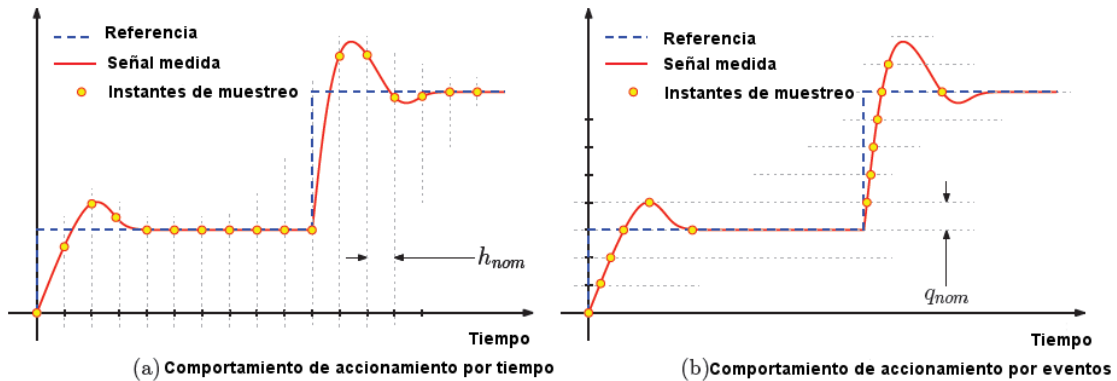


Figura 2.10: Muestreo basado en tiempo a) vs. Basado en eventos b) (Cortesía Tesis Doctoral S.Durand 2011 [20]).

Extendiendo la analogía al cálculo de la integral de Riemann y Lebesgues. Labesgues fue in-

roducida para denotar este esquema de muestreo: Las mediciones de las elevaciones son tomadas únicamente cuando las variables cruzan algunos niveles específicos por oposición al muestreo de Riemann donde las mediciones de las elevaciones son tomadas en instantes específicos. La idea es minimizar la carga computacional del controlador por la reducción del número de muestreos con el comportamiento basado en eventos. No es necesario calcular la señal de control durante el intervalo de estado constante desde que el sistema alcanza el estado estable. En el esquema de muestreo accionado por eventos, el término intervalo de muestreo denota un intervalo entre dos cruces de nivel consecutivos de la medida, que son dos instantes de muestreo sucesivos. El intervalo de muestreo no es equidistante, como se puede analizar en los intervalos de tiempo entre dos instantes sucesivos en la figura 2.10(b)[20].

2.4. Muestreo basado en eventos como una oportunidad para sistemas embebidos

Aunque la periodicidad simplifica el diseño y el análisis, esto resulta un uso conservador de los recursos desde que la ley de control es calculada y actualizada con el mismo periodo independientemente si es necesario o no. Como resultado, un recurso al tanto de la implementación de la ley de control, usando muestreo basado en eventos, podría ser una solución prometedor para chips embebidos. Por esta razón, las diferentes comunidades científicas están enfocadas a desarrollar nuevas estrategias de control basado en eventos. Sin embargo, extendiéndose a algunos resultados de sistemas muestreados con Riemann a sistemas muestreados con Lebesgues parece no ser una buena estrategia, desde el tiempo será inevitable aparecer una nueva relación. La conversión tiene que efectuarse cuidadosamente [20].

2.4.1. La necesidad de un funcionamiento asíncrono en sistemas de control basados en redes de comunicaciones

Muchas razones están motivando los sistemas accionados por eventos, en particular porque se encuentran mas y mas sistemas asíncronos o sistemas con necesidades asíncronas. Actualmente, la demanda de componentes electrónicos de bajo consumo de voltaje en todas las aplicaciones embebidas y miniaturizadas anima a las compañías a desarrollar versiones asíncronas de los componentes existentes accionados por tiempo, donde una significativa reducción en consumo de energía puede ser lograda por disminuir el muestreo y consecuentemente la utilización del CPU: En los sistemas micro-electrónicos como es el caso del microcontrolador 80C51 de Philips Semiconductor, el cual ha mostrado reducir en un orden de 4 el consumo de energía que su contraparte síncrona [21].

Los sensores y los actuadores basados en eventos por cruce de nivel también existen, dando un completo control de lazo asíncrono[20]. Como un resultado, el concepto basado en eventos está tomando mas y mas importancia en la comunidad de control [6],[22],[23], [24].

Aunque el control accionado por eventos está bien motivado y permite relajar la periodicidad para el cálculo de la ley de control, únicamente pocos trabajos reportan resultados teóricos acerca de la estabilidad, convergencia y desempeño de sistemas de control accionado por eventos. Por ejemplo en [25],[26], es probado que tal acercamiento reduce el número de instantes de muestreo para el mismo desempeño final.

Esto es también mostrado en [27], que controlar un sistema muestreado por Lebesgues o un sistema en tiempo continuo con medidas calculadas y una ley de control constante en el periodo de muestreo son problemas equivalentes. Trabajos recientes distribuidos con el problema de programar la tarea de control para sistemas lineares de tiempo continuo [28],[29],[30] y sistemas lineares de tiempo discreto [31] donde la estabilidad y algunas propiedades robustas son aprovechadas. Además, en [31] un modelo de esquema de control predictivo es usado donde las políticas de accionamiento por eventos son usadas para relajar la demanda computacional de algoritmos. En [28], un análisis formal hace el intercambio entre la reducción en la utilización de recursos y el desempeño del control

para un esquema particular basado en eventos. Algunas importantes contribuciones también vienen de la comunidad de control de tiempo real. De hecho, las tareas de control síncrono en tiempo real son a menudo consideradas como tareas difíciles en término de tiempo de sincronización, requiere unas fuertes limitaciones de tiempo real. La principal consecuencia es el sobre tamaño de los cálculos que acarrea un costo adicional no muy compatible con una larga producción como para sistemas embebidos. La aproximación adoptada en este campo es a menudo cualquiera para cambiar dinámicamente el periodo de muestreo relacionado a la carga [32],[33] o a usar un control basado en eventos donde los eventos son generados con una mezcla de cruce de nivel y un máximo periodo de muestreo [6],[24]. Un paradigma accionado por eventos es activado siempre que sean realmente necesario.

2.4.2. Dificultades al conjugar el control asíncrono con paradigmas bien establecidos

Muchas de las aproximaciones asíncronas son implícitamente síncronas en el sentido que el tiempo es usado para determinar si el control debe ser calculado. Es difícil de desligar eso ya que el paradigma síncrono está bien establecido. Por ejemplo, el máximo periodo de muestreo en [6] fue agregado por razones de estabilidad, en orden para cumplir la condición de muestreo de Nyquist-Shannon. Así, una nueva señal de control es ejecutada cuando la suma del transcurso del tiempo desde el último muestreo excede un cierto límite. Sin embargo, esta condición de límite de seguridad no es muy consistente gracias a la detección de nivel y, intuitivamente, ésta puede ser removida para reducir otra vez el costo de CPU. Sin embargo, este límite de seguridad es inútilmente aplicado en varios "papers", tal como en [23],[34]. No obstante, un primer acercamiento podría ser para analizar algunas estrategias de control basado en eventos sin algún límite de seguridad. Entonces, esto podría también ser interesante para encontrar algunas herramientas teóricas para probar que los controladores basados en eventos permiten asegurar un mejor desempeño, incluso si el sistema no es muestreado por un largo tiempo. En el futuro, una estructura de control totalmente asíncrona es esperada eso subsecuentemente sería adecuado para muchos sistemas.

2.5. Protocolo de comunicación CAN-BUS

Sus siglas vienen del acrónimo del inglés "Controller Area Network", es un protocolo de comunicación bus serial desarrollado en 1980(marca el comienzo de la etapa del *control en red*) por BOSCH (manufacturador de equipo eléctrico en Alemania), fue estandarizado como ISO-11898 e ISO-11519 y establecido como un protocolo estándar de la red de un automóvil. Actualmente el protocolo CAN es usado en muchos campos que llaman a la red control embebido; estos campos son, la industria de automatización, aplicaciones médicas, automatización de la construcción, maquinas textiles, y producción de maquinaria. CAN ofrece un protocolo de comunicación eficiente, entre sensores, actuadores, controladores, y otros nodos en aplicaciones de tiempo real, es conocido por su sencillez, confiabilidad y su alto desempeño [10].

El protocolo CAN está basado en la topología bus y únicamente son necesarios dos alambres para la comunicación a través del bus CAN. El bus tiene una estructura multi maestro donde cada dispositivo en el bus puede enviar o recibir datos. Únicamente un dispositivo puede enviar datos a la vez mientras todos los demás escuchan. Si dos o más dispositivos intentan enviar datos al mismo tiempo el que tenga la prioridad más alta enviara datos mientras los demás pasan a modo de recepción.

Como se muestra en la figura 2.11, en una aplicación típica vehicular hay usualmente mas de un bus CAN y estos operan a velocidades diferentes. Dispositivos lentos, como el control de puerta, control de clima y módulo de información de manejo, pueden ser conectados a un bus de velocidad lenta. Dispositivos que requieren respuesta rápida, sistema de frenos antibloqueo ABS, el módulo de control de transmisión, y el módulo de aceleración electrónica, son conectados a un bus CAN

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES

2.5. PROTOCOLO DE COMUNICACIÓN CAN-BUS

rápido.

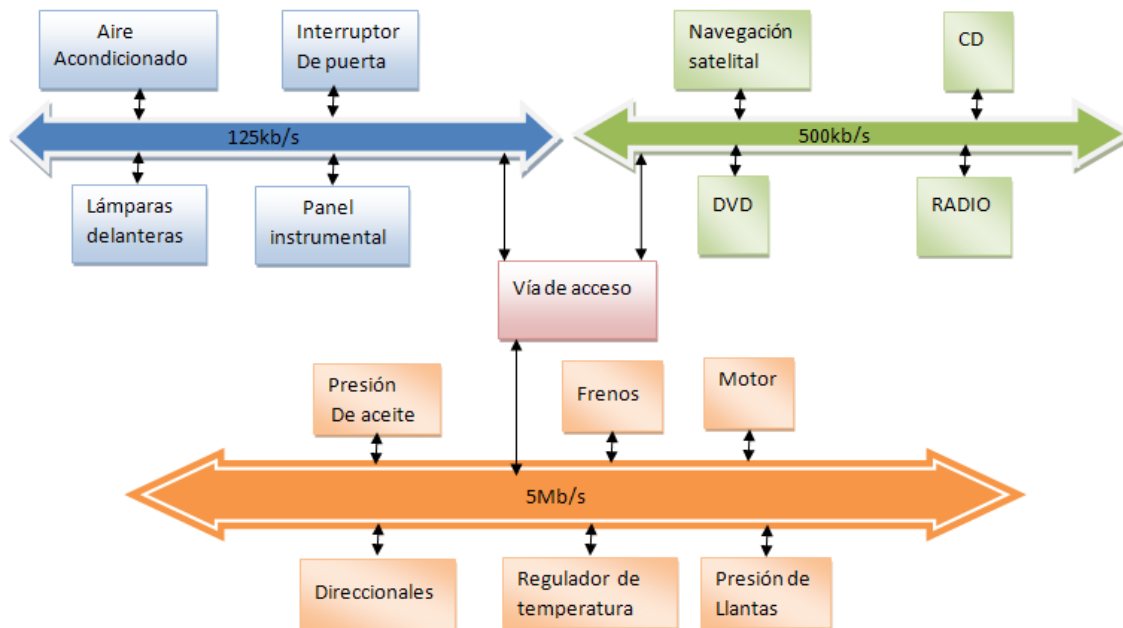


Figura 2.11: Aplicación típica del bus CAN en un vehículo.

La industria automotriz es la principal causa de la producción de los controladores CAN. Actualmente se estima que son 400 millones de módulos CAN que son vendidos todo el año, y son integrados controladores CAN en muchos microcontroladores, incluyendo microcontroladores PIC que están disponibles a un muy bajo costo.

El protocolo CAN está basado en el protocolo CSMA/CD+AMP (acceso múltiple de sentido-transportable/ detección de colisión con arbitraje de prioridad de mensaje), que es similar al protocolo usado en Ethernet LAN. Cuando el protocolo Ethernet detecta una colisión, el nodo que está enviando simplemente detiene la transmisión y espera una cantidad aleatoria de tiempo antes que intente enviar otra vez. El protocolo CAN, sin embargo, resuelve los problemas de colisión usando el principio de arbitraje, donde únicamente el nodo con prioridad más alta tiene permiso de enviar datos [10].

Hay básicamente dos versiones de protocolos CAN: 2.0A y 2.0B. CAN 2.0A es la primera versión estándar con 11 bits de identificación, mientras que CAN 2.0B es la nueva versión estándar extendida con 29 bits de identificación. Los controladores de la 2.0B son compatibles con versiones anteriores con controladores 2.0A y pueden transmitir y recibir mensajes en ambos formatos.

Algunas de las características del protocolo CAN son:

- El bus CAN es multimaestro. Cuando el bus está libre, cualquier dispositivo conectado al bus puede iniciar a enviar mensajes.
- El protocolo bus CAN es flexible. Los dispositivos conectados a él no necesitan tener dirección. Significa que los mensajes no son transmitidos de un nodo a otro basados en una dirección. Todos los nodos en el sistema reciben todos los mensajes transmitidos en el bus y cada nodo decide si el mensaje recibido será guardado o descartado. Un mensaje puede ser destinado para un nodo particular o para varios nodos, dependiendo de cómo está diseñado el sistema. Otra ventaja de no tener dirección es que cuando un dispositivo es agregado o quitado del bus, no necesita que sea cambiada la configuración de los datos.

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES

2.5. PROTOCOLO DE COMUNICACIÓN CAN-BUS

- El bus ofrece petición de transmisión a distancia (RTR), que significa que un nodo en el bus es capaz de pedir información de otro nodo. En lugar de esperar que un nodo envíe información continuamente, la petición de información puede ser enviada al nodo. Por ejemplo, en un vehículo, donde el control de temperatura es un parámetro importante, el sistema puede ser diseñado para que la temperatura sea enviada periódicamente por el bus. Sin embargo, una solución mas elegante es pedir la temperatura cuando sea necesario, esto minimiza el trafico en el bus mientras se mantiene la integridad de la red.
- La velocidad de la comunicación CAN no está fija. Cualquier velocidad de comunicación puede ser fijada para los dispositivos conectados al bus.
- Todos los dispositivos conectados al bus pueden detectar un error. El dispositivo que detecta un error inmediatamente avisa a los demás dispositivos.
- Pueden ser conectados muchos dispositivos al bus al mismo tiempo, no existe un límite de número de dispositivos. En la práctica, el número de unidades que pueden ser conectadas al bus es limitada por el tiempo de retardo del bus y la carga eléctrica.

Los datos en el bus CAN son diferenciales y pueden tener dos estados: dominante y recesivo. La figura 2.12 muestra los estados de voltajes en el bus. El bus define un bit lógico 0 como un bit dominante y un bit lógico 1 como un bit recesivo. Cuando hay arbitraje en el bus, el estado de bit dominante gana sobre el estado de bit recesivo. En el estado recesivo, el voltaje diferencial CANH y CANL es menos que el umbral mínimo (menos que la entrada recibida 0.5V y menos que la salida transmitida 1.5V). En el estado dominante, el voltaje diferencial CANH y CANL es más grande que el umbral mínimo [10].

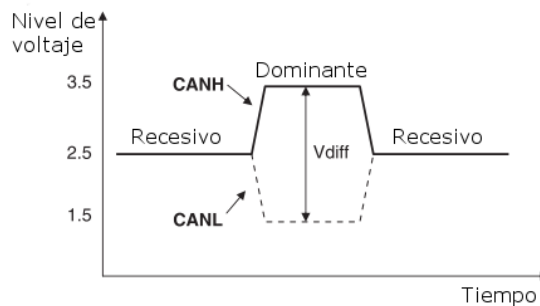


Figura 2.12: Estados lógicos del bus CAN.

La norma ISO-11898 del bus CAN especifica que un dispositivo en el bus debería ser capaz de funcionar a una distancia de 40 metros a una velocidad de 1Mb/s. Mientras más largo sea el bus, la velocidad disminuirá. La figura 2.13 muestra la variación de la longitud del bus con la velocidad de comunicación. Por ejemplo un bus con una longitud de 400 metros puede alcanzar una velocidad máxima de 100kb/s [10].

El bus CAN está diseñado para reducir los reflejos de señal en el bus. La ISO-11898 requiere que el bus tenga una impedancia característica de 120ohms. El bus puede ser diseñado con uno de los siguientes métodos:

- Terminación estándar
- Terminación dividida
- Terminación dividida sistemáticamente

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES
2.5. PROTOCOLO DE COMUNICACIÓN CAN-BUS

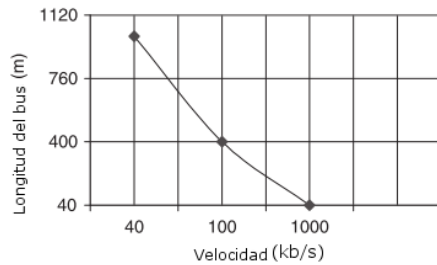


Figura 2.13: Velocidad del bus CAN y longitud del bus.

La *terminación estándar* es método más común, se usa un resistor de 120-ohm en cada punta del bus, como se ven en la figura 2.14(a), con esta terminación la cantidad de energía disipada en el envío de mensajes a través de la red es 0.25W [36]. En la *terminación dividida*, las puntas del bus están divididas y se usa un resistor de 60-ohm como se muestra en la figura 2.14(b). La *terminación dividida sistemáticamente* es muy parecida a la terminación dividida, excepto que se usan un circuito divisor de voltaje y un capacitor en ambas puntas del bus, se muestra en la figura 2.14(c) [10].

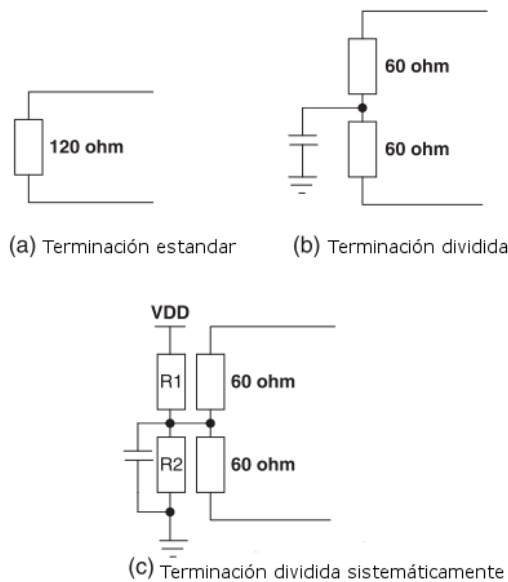


Figura 2.14: Métodos de terminación del bus.

2.5.1. Estructura de datos

La estructura de datos está en dos formatos: Estándar (tiene 11 bits de identificación) y la extendida (tiene 29 bits de identificación). La estructura de datos se usa en dispositivos transmisores para enviar datos a los dispositivos receptores, y la estructura de datos es la estructura más importante manipulada por el usuario. En la figura 2.15 se muestra como está constituida la estructura de datos. Una estructura de datos estándar inicia con el bit de inicio de la estructura (SOF), que es seguido por 11 bits de identificador, y el bit de petición de transmisión a distancia (RTR). El identificador y el RTR forman los 12 bits del campo de arbitraje. El campo de control es de un ancho de 6 bits y indica como muchos bytes de datos están en el campo de datos. El campo de

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES

2.5. PROTOCOLO DE COMUNICACIÓN CAN-BUS

datos puede ser de 0 a 8 bytes. El campo de datos es seguido por el campo CRC, que checa si la secuencia de bit recibida está corrompida. El campo ACK es usado por el transmisor para recibir reconocimiento de una estructura valida de algún receptor. El final del mensaje es indicado por 7 bits del campo, fin de estructura (EOF) [10].

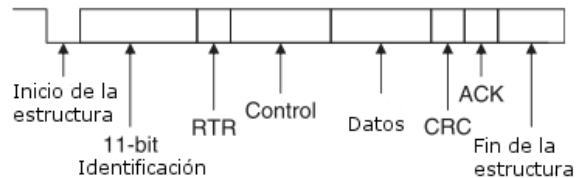


Figura 2.15: Estructura estándar de datos.

En una estructura de datos extendida, el campo de arbitraje es de un ancho de 32 bits (29 bits de identificador + 1 bit de IDE, para definir el mensaje como una estructura de datos extendida + 1 bit de SRR que es inutilizable + 1 bit de RTR). Ver figura 2.16 [10].

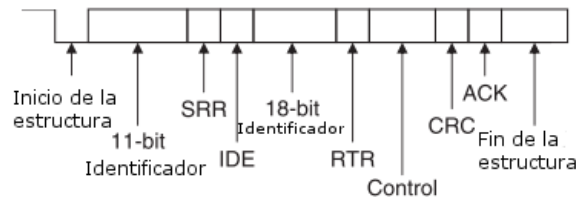


Figura 2.16: Estructura extendida de datos.

La estructura de datos está constituida de los siguientes campos:

2.5.1.1. Inicio de la estructura (SOF)

El inicio de la estructura indica cuando comienza la estructura de datos y es común para ambos formatos el estándar y el extendido [10].

2.5.1.2. Campo de arbitraje

El arbitraje es usado para resolver conflictos del bus que ocurren cuando varios dispositivos a la vez inician a enviar mensajes en el bus. El campo de arbitraje indica la prioridad de la estructura, es diferente entre el formato estándar y el formato extendido. En la versión estándar hay 11 bits, y arriba de 2032 IDs pueden ser fijados. El formato ID extendido, 11 bits de base más 18 para la extensión y arriba de 2032×2^{18} IDs pueden ser fijados [10].

Durante la etapa de arbitraje, cada dispositivo transmisor transmite su identificador y lo compara con el nivel del bus. Si los niveles son iguales, el dispositivo continúa la transmisión. Si el dispositivo detecta un nivel dominante en el bus mientras está intentando transmitir un nivel recesivo, éste finaliza y pasa a ser un dispositivo receptor. Después de un arbitraje, solo un transmisor es permitido en el bus, y éste dispositivo continúa enviando su campo de control, campo de datos, y otros datos.

El proceso de arbitraje se ilustra en la figura 2.17, se muestran los identificadores de tres nodos.

Se asume que el nivel recesivo corresponde a 1 y el nivel dominante a 0, el arbitraje es realizado de la siguiente manera:

- Todos los dispositivos inician a transmitir al mismo tiempo, primero envían el bit SOF.

CAPÍTULO 2. CONTEXTO Y MOTIVACIONES
2.5. PROTOCOLO DE COMUNICACIÓN CAN-BUS

Nodo 1: 11100110011 Nodo 2: 11100111111 Nodo 3: 11100110001

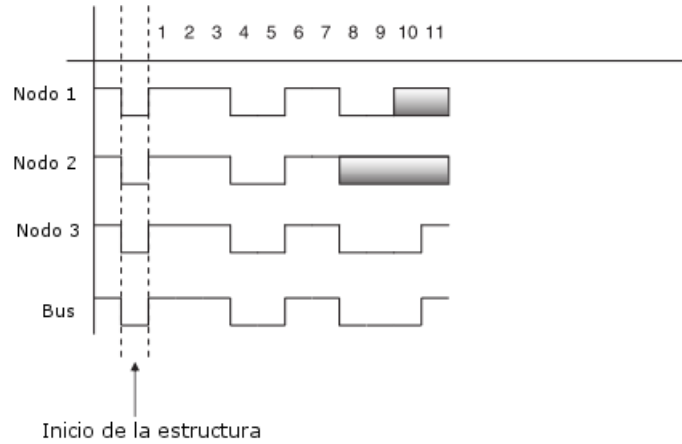


Figura 2.17: Ejemplo de arbitraje del bus CAN.

No. de bytes de datos	DCL3	DCL2	DCL1	DCL0
0	D	D	D	D
1	D	D	D	R
2	D	D	R	D
3	D	D	R	R
4	D	R	D	D
5	D	R	D	R
6	D	R	R	D
7	D	R	R	R
8	R	D o R	D o R	D o R

Tabla 2.1: Codificación del campo de control(Nivel dominante D, nivel recesivo R).

- Después envían sus bits de identificación. El octavo bit del nodo 2 está en estado recesivo, mientras que en los correspondientes bits del nodo 1 y nodo3 están en estado dominante. Por lo tanto el nodo 2 detiene la transmisión y regresa a modo de recepción. La fase de recepción es indicada por el campo gris.
- El bit 10 del nodo 1 se encuentra en estado recesivo, mientras que el nodo 3 está en estado dominante. Así el nodo 1 detiene la transmisión y regresa a modo receptor.
- El bus permite que el nodo 3 pueda enviar sus campos de control y de datos libremente.

Notablemente los dispositivos en el bus no tienen direcciones. De hecho todos los dispositivos detectan el mensaje, verifican el campo de identificación y si les corresponde el mensaje permiten que continúe transmitiendo los demás campos [10].

2.5.1.3. Campo de control

El campo de control es de un ancho de 6 bits, consiste en 2 bits reservados y un código de longitud de datos de 4 bits (DLC), indica el número de bytes de datos en el mensaje que comenzarán a transmitirse. Este campo es codificado como se muestra en la tabla 1.1 [10].

2.5.1.4. Campo de datos

El campo de dato transporta el contenido actual del mensaje. El tamaño del dato puede variar de 0 a 8 bytes [10].

2.5.1.5. Campo de CRC

El campo CRC consiste en una secuencia de 15 bits y un bit delimitador; es usado para comprobar la estructura y ver si existe un error. El cálculo CRC, incluye el inicio de la estructura, campo de arbitraje, campo de control, y campo de dato. La secuencia CRC calculada y la CRC recibida son comparadas; si no son iguales se asume un error [10].

2.5.1.6. Campo de ACK

El campo ACK indica que la estructura ha sido recibida con normalidad. Este campo consiste en 2 bits, un bit para ACK y otro para el delimitador de ACK.

2.6. TRUE TIME de MATLAB/Simulink

La mayoría de los controladores actuales están retroalimentados por red y la herramienta comúnmente usada MATLAB/Simulink. Los controladores retroalimentados por red son implementados como tareas en un núcleo en tiempo real y se comunican con otros nodos mediante la red. Consecuentemente se necesita tomar en cuenta a la hora de diseñar, las restricciones de la tarjeta del sistema, como la velocidad límite del CPU y el ancho de banda de la red.

La herramienta TRUE TIME fue desarrollado por Dan Henriksson, Anton Cervin, Karl-Erik Årzén para simular sistemas de control en tiempo real distribuidos. TRUE TIME hace esto posible para simular el comportamiento de núcleos en tiempo real, ejecutando tareas del controlador. También TRUE TIME simula protocolos de red y su influencia en lazos de control en red.

TRUE TIME consiste en bloques de núcleo y bloques de red, las variables son funciones escritas en C++. El bloque núcleo de TRUE TIME simula una computadora con un núcleo en tiempo real accionado en eventos, convertidores A/D y D/A, una interface de red, y canales de interrupción externos. El núcleo ejecuta tareas definidas por el usuario y gestores de interrupción, representan por ejemplo tareas I/O, algoritmos de control y tareas de comunicación. La ejecución es definida por un código de función definido por el usuario (función C++ ó archivos M) o gráficamente usando bloques ordinarios de Simulink (discretos). El tiempo de ejecución de la simulación del código puede ser modelado por una constante, aleatorio o incluso dependiente de datos. A demás, las políticas de programación en tiempo real del núcleo son arbitrarias y decididas por el usuario [13].

El bloque red de TRUE TIME es accionado en eventos y distribuye los mensajes entre los nodos de acuerdo con el modelo de red seleccionado. Actualmente TRUE TIME soporta seis de los protocolos de control de acceso medio (CSMA/CD(Ethernet), CSMA/AMP(CAN), token-ring, Switched Ethernet, FDMA, y TDMA). Es posible especificar los parámetro de la red como: la velocidad de transmisión, retardos después o antes del proceso, probabilidad de pérdida.

Actualmente TRUE TIME se usa en plataformas experimentales para investigación de aproximaciones flexibles, para programación e implementación en tiempo real de tareas de control [20].

2.6.1. El núcleo de TRUE TIME

El bloque núcleo es configurado a través del cuadro de diálogo que se muestra en la figura 2.18, con los siguientes parámetros [13](algunos parámetros también pueden ser establecidos con el comando `ttSetKernelParameter`):

Init function: Nombre del escrito de inicialización.

Init function argument: Argumento opcional para inicializar el escrito.

Battery: Habilitar este cuadro si el núcleo depende de una fuente de energía.

Clock drift: El tiempo fluctúa, 0.01 si el tiempo local debería ejecutarse 1% más rápido que el tiempo nominal (El tiempo actual de simulación).

Clock offset: Tiempo constante de compensación del tiempo nominal.

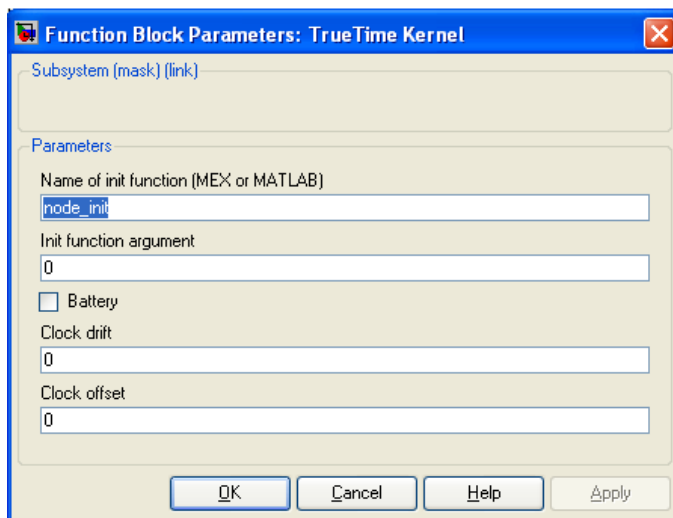


Figura 2.18: Cuadro de diálogo del bloque núcleo de TRUE TIME.

2.6.1.1. Voltaje dinámico escalado

Con el uso de `ttSetKernelParameter`(ver apéndiceA), la velocidad actual de ejecución del núcleo puede ser fijada y también la fuente de energía actual. Esto se hace posible para simular el voltaje dinámico escalado. Esta funcionalidad es muy usual junto con el bloque de batería [13].

2.6.2. La Red TRUE TIME

El bloque de Red de TRUE TIME simula acceso medio y transmisión de paquetes en una red de area local. Cuando un nodo intenta transmitir un mensaje(usando el comando `ttSendMsg`), una señal de activación se envía al bloque de Red en el canal de entrada correspondiente. Cuando la simulación de transmisión del mensaje finaliza, el bloque de Red envía una nueva señal de activación en el canal correspondiente de salida al nodo receptor. El mensaje transmitido es puesto en un buffer en el nodo ordenador receptor. Un mensaje contiene información acerca del nodo ordenador transmisor y receptor, arbitrariamente datos del usuario(típicamente señales de medición y de control), la longitud del mensaje y atributos opcionales de tiempo real como prioridad y fecha límite [13].

Soporta seis modelos simples de red: CSMA/CD(por ejemplo Ethernet), CSMA/AMP(por ejemplo CAN), Round Robin(por ejemplo Token Bus), FDMA, TDMA (por ejemplo TTP), y Switched Ethernet. Se ignora la propagación del retardo, este es típicamente muy pequeño en una

red de area local. Únicamente soporta la simulación de niveles de paquetes, asume que el más alto nivel de protocolo en el núcleo de nodo tienen dividido los mensajes largos en paquetes.

El bloque de la red es configurado a través del cuadro de diálogo, figura 2.19. Usando el comando `ttSetNetworkParameter`, ver la sección de comandos en el apéndice A, es posible cambiar algunos parámetros [13]. Los siguientes parámetros de red son comunes en todos los modelos:

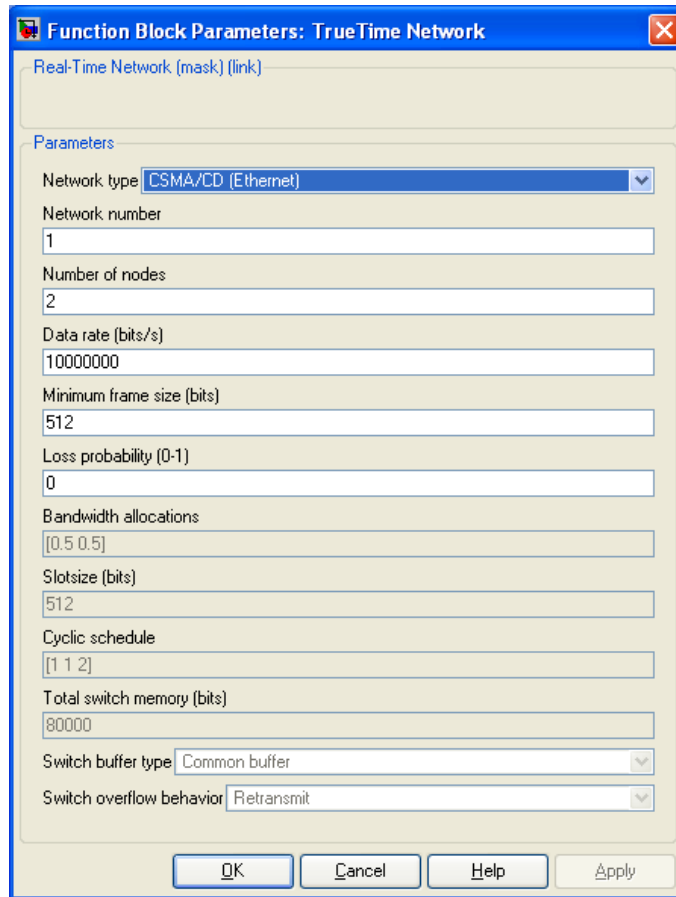


Figura 2.19: Cuadro de diálogo del bloque de red TRUE TIME.

Network number:Número del bloque en la red. La red tiene que ser numerada a partir de 1. Las redes mediante alambres y las inalámbricas no permiten usar el mismo número.

Number of nodes:Número de nodos que son conectados en la red. El número determinara el tamaño de Snd, Rcv, la entrada de programación y las salidas del bloque.

Data rate(bits/s) La velocidad de la red.

Minimum frame size (bits)Mensaje o marco mas corto, éste será rellenado para dar la longitud mínima. Denota el tamaño mínimo del marco.

Pre-processing delay(s)Tiempo que un mensaje es retardado por la interface de la red, en el final del envío. Éste es usado para modelar, una conexión serial baja entre la computadora y la interface de red.

Post-processing delay(s)Tiempo que un mensaje es retardado por la interface de la red en el final de la recepción.

Loss probability (0-1) La probabilidad de que un mensaje de red sea perdido durante la transmisión. Los mensajes consumirán ancho de banda de la red, pero nunca llegaran a su destino.

2.6.2.1. CSMA/AMP(CAN)

CSMA/AMP representa un portador de acceso multiple con arbitraje en la prioridad del mensaje. Si la red está ocupada, el nodo que está enviando esperará hasta que la red se encuentre libre. Si ocurre un choque(Dos transmisiones inician dentro de 1 microsegundo), el mensaje con la prioridad más alta (el número más bajo de prioridad)se seguirá transmitiendo. Si dos mensajes con la misma prioridad buscan ser transmitidos al mismo tiempo, una elección arbitraria sera hecha. (En aplicaciones reales de CAN, todos los nodos que están enviando, tienen un identificador único que sirve como la prioridad del mensaje)[13].

2.6.3. La batería de TRUE TIME

El bloque de batería tiene un parámetro, la energía inicial, que tiene que ser fijada usando la mascara de configuración. Para usar la batería se tiene que habilitar el cuadro en la mascara de configuración del núcleo y conectar la salida de la batería a la entrada E del bloque núcleo. En la entrada P del bloque batería se conecta la señal que indica el desgaste de la batería, comúnmente se conecta todo el flujo de energía de tal forma que la salida P (del bloque núcleo, modelos ordinarios de Simulink, y el bloque de red inalámbrica) a la entrada P de la batería [13]. El cuadro de diálogo del bloque batería se muestra en la figura 2.20, el único parámetro que se carga es la energía inicial de la batería, está dado en términos de Watt segundos(Joule).

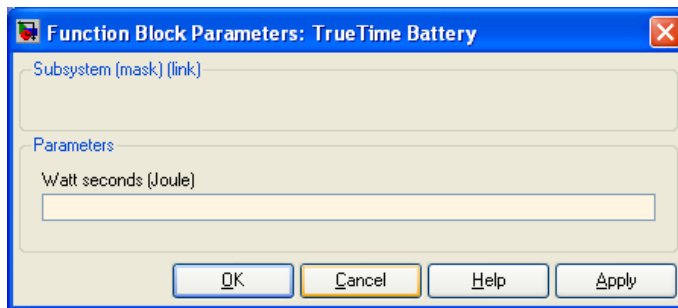


Figura 2.20: Cuadro de diálogo del bloque batería de TRUE TIME.

Capítulo 3

Control PID basado en eventos

En este capítulo, en la primera sección se analiza la estrategia propuesta por K.E.Årzén, se describe las condiciones de detección de eventos y la implementación en código. En la siguiente sección se describen las cuatro nuevas estrategias(sin condición de límite de seguridad, saturación de la ganancia integral, factor exponencial olvidado del intervalo de muestreo, híbrida) propuestas por Sylvain Durand , se analiza su condición de detección de eventos, las condiciones lógicas para cada una de ellas, se muestra gráficamente el comportamiento de la ganancia integral en el aumento del producto $he(.)$. En la última sección se describen los índices IAE, IAEP y IAD.

3.1. Control PID basado en eventos propuesto por Årzén

La decisión de calcular una nueva señal de control es tomada cuando se cruza cierto nivel. El controlador propuesto por Årzén se basa en dividir el controlador en dos partes:1) la lógica de detección de eventos y 2) la lógica de control. La estructura de este control se muestra en la figura3.1 [7].

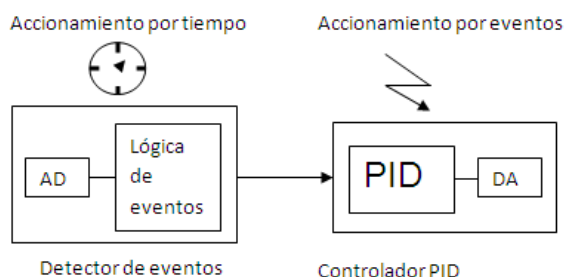


Figura 3.1: Control PID basado en eventos de Årzén.

El detector de eventos se ocupa de indicar a la lógica de control que debe producir una acción de control debido a que se dan las circunstancias adecuadas, es decir cuando se cruza el nivel propuesto. La lógica de detección de eventos es síncrona y está sujeta a un periodo de muestreo nominal de forma que cada h_{nom} se examina si se cumplen las condiciones necesarias para activar la lógica de control [7]. Las dos condiciones de activación del control PID basado en eventos de Årzén son las siguientes:

$$|e - e_s| \geq \delta \quad \text{ó} \quad (t_k - t_s) \geq h_{max}$$

Las condiciones anteriores expresan que el cálculo de una nueva acción de control puede producirse ante dos situaciones: por error o por tiempo. La primera de las condiciones analiza si el valor

CAPÍTULO 3. CONTROL PID BASADO EN EVENTOS
3.1. CONTROL PID BASADO EN EVENTOS PROPUESTO POR ÅRZÉN

<pre> // Precálculo b_i := k_p/T_i; //Lógica de detección de eventos r := ADIn(ch1); y := ADIn(ch2); e := r - y; h_{act} := h_{act} + h_{nom}; IF(abs(e - e_s) > δ)OR(h_{act} >= h_{max})THEN e_s := e; a_d := T_d/(T_d + N * h_{act}); // Cálculo de la señal de control </pre>	<pre> u_p := k_p * (β * r - y); u_d := a_d * u_d - a_d * k_p * N * (y - y_{old}); u := u_p + u_i + u_d; //Enviar acción de control al actuador DAOOut(u, ch3); //Actualización del estado u_i := u_i + b_i * h_{act} * (r - y); y_{old} := y; h_{act} := 0,0; ENDIF; </pre>
--	---

Tabla 3.1: Código del controlador PID de Årzen.

absoluto de la diferencia entre el valor actual del error e , y el valor del error en que se disparó el cálculo de la última acción de control e_s supera un umbral δ . La segunda condición es una medida de seguridad que obliga a producir una nueva acción de control cuando el tiempo transcurrido desde que se calculó la última supera un límite h_{max} figura3.2 [7].

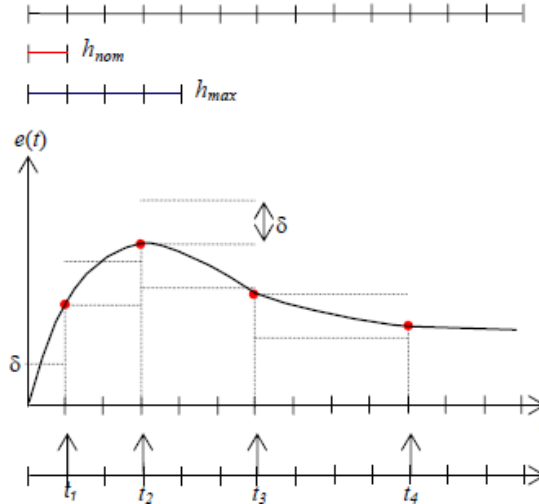


Figura 3.2: Muestreo en el controlador PID de Årzen.

Esta doble condición produce también un doble efecto: el control se realizará a la frecuencia máxima $1/h_{nom}$ durante transitorios (cambios de referencia, perturbaciones); y se realizará a la frecuencia mínima $1/h_{max}$ durante los periodos estacionarios [7].

Por lo que respecta a la lógica de control, el algoritmo utilizado es un controlador PID tradicional que se ha discretizado utilizando una aproximación por diferencias hacia adelante en el término integral y hacia atrás en el término derivativo [7].

A continuación en la tabla 3.1 se presenta el código completo del controlador PID. No puede ser precalculado una gran parte del código debido a que el periodo de muestreo h_{act} , que se utiliza para generar la acción de control, es variable ya que oscila de un mínimo h_{nom} a un máximo h_{max} . En el código del controlador se puede apreciar que un valor elevado de h_{act} produce un efecto análogo a la aplicación de un valor reducido de T_i y, recíprocamente, un valor pequeño de h_{act} produce el mismo efecto que un T_i elevado [7].

La estrategia del controlador PID basado en eventos de Årzen reduce de manera importante la utilización de CPU en el controlador digital [7].

3.2. Nuevas estrategias de Control PID basado en eventos

El trabajo original de Årzén tiene un periodo máximo de seguridad h_{max} que fuerza el cálculo de la señal de control, si la señal medida permanece sin cambios [7]; esta condición parece haber sido agregada por razones de inestabilidad, para que se cumpla la condición del teorema de muestreo de Nyquist-Shanon. Sin embargo en [11] se propone remover el límite de periodo máximo de seguridad gracias a la detección de nivel, ya que la condición de muestreo de Nyquist-Shannon no es muy consistente [11]. En [12] se muestran los resultados experimentales. En esta sección se estudian cuatro de las estrategias de control PID basado en eventos que eliminan el re cálculo de la señal de control [11].

En el trabajo de Årzén, si es cambiada la señal de referencia después de un largo periodo de estado estacionario, aparecerán importantes sobre tiros. La parte integral de la ecuación 3.1 del controlador PID basado en eventos, es responsable de estos sobre tiros. Donde t_a es el instante actual donde se detectó un evento, t_{a-1} es el instante anterior donde se detectó un evento.

$$u_i(t_a) = u_i(t_{a-1}) + K_i \cdot h(t_a) \cdot e(t_a) \quad (3.1)$$

Después de esperar a un evento durante mucho tiempo, el valor de $h(\cdot)$ se va haciendo muy grande [11]. Mientras no ocurre un evento $u_i(t_a)$ crece desmedidamente, el valor de $e(\cdot)$ aumenta drásticamente y así termina el estado estacionario. El cálculo de la parte integral necesita del producto $he(\cdot)$ que es llamado *ganancia integral*, este producto explota al final del intervalo del estado estacionario. Para eliminar este problema, el estado estacionario se analiza detalladamente, en la figura 3.3 se muestra como se puede dividir en dos partes, que son: *i*) el intervalo de tiempo donde la señal está realmente en estado estacionario y *ii*) el intervalo de tiempo requerido para detectar un nuevo cruce de nivel. La primera parte inicia después del calculo de la última señal de control, es en el instante de muestreo t_{a-1} y termina justamente después de que la señal de referencia cambia.

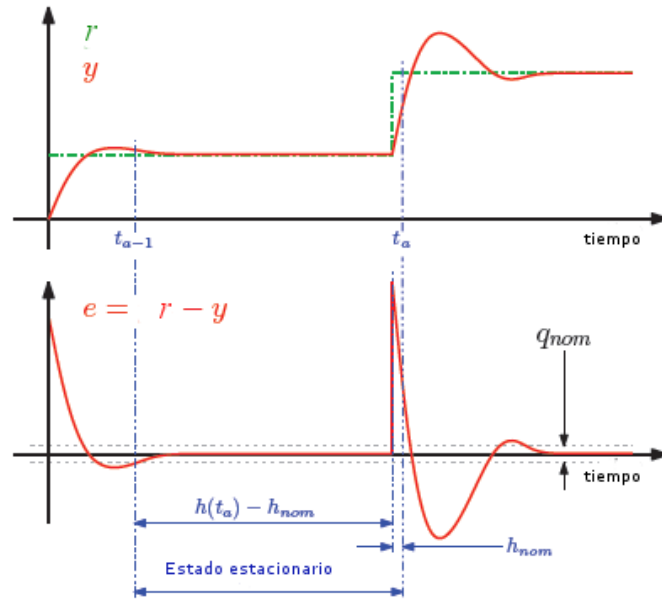


Figura 3.3: Descomposición de un intervalo de estado estacionario.

Entonces mientras el error llega a ser más grande que el nivel de detección q_{nom} , la segunda parte inicia. Se envía una petición y finalmente se calcula una nueva señal de control en el tiempo t_a . Sin embargo el tiempo exacto del cruce de nivel podría ser entre dos instantes de muestreo

CAPÍTULO 3. CONTROL PID BASADO EN EVENTOS
3.2. NUEVAS ESTRATEGIAS DE CONTROL PID BASADO EN EVENTOS

accionados por tiempo sucesivos t_{k-1} y t_k (por que el detector de eventos es accionado por tiempo t_k). El peor caso ocurre cuando la señal de referencia cambia en el tiempo $t_{k-1} + \varepsilon$, donde ε es muy pequeño. En este caso, la petición es enviada únicamente en el segundo instante de muestreo t_k y eventualmente la medida del error $e(t_a)$ es grande durante casi todo el periodo de muestreo h_{nom} . El tiempo exacto no podría ser conocido, pero la ganancia integral $he(\cdot)$ se puede limitar superiormente de cualquier modo durante el tiempo de detección de un transitorio por el producto $h_{nom} \cdot e(t_a)$. Consecuentemente, la primera parte corre únicamente del instante de tiempo t_{a-1} a $t_a - h_{nom}$ que es igual al intervalo de tiempo $h(t_a) - h_{nom}$. Durante este periodo, el error permanece muy pequeño, el error es menor que el nivel de detección q_{nom} de lo contrario el estado estacionario no sería logrado, y así es el producto $he(\cdot)$. De esta manera, la ganancia integral también se limita superiormente por el producto $(h(t_a) - h_{nom}) \cdot q_{nom}$. Para resumir, un intervalo de estado estacionario puede ser dividido en *i*) la primera parte donde el intervalo de muestreo aumenta mucho, pero el error permanece pequeño y *ii*) una segunda parte donde el error llega a ser muy grande pero únicamente durante pocos instantes. Por lo tanto, el producto $he(\cdot)$ no explotara desde que $h(\cdot)$ y $e(\cdot)$ se compensan entre sí. Para incluir un valor más preciso a la ganancia integral, en [20] se propone la ecuación 3.2.

$$u_i(t_a) = u_i(t_{a-1}) + K_i \cdot he(t_a) \quad (3.2)$$

Donde $he(t_a) \leq (h(t_a) - h_{nom}) \cdot q_{nom} + h_{nom} \cdot e(t_a)$

Se puede notar que esta desigualdad, que inicialmente estuvo construida por el intervalo de estado estacionario es finalmente correcta por el funcionamiento completo. De hecho, la ecuación 3.2 viene de la ecuación 3.1 durante un transitorio, desde que en este caso $h(t_a) = h_{nom}$. Basado en esta suposición, en [20] se proponen seis algoritmos sin condición de límite de seguridad. Pero en esta tesis, solo se estudiarán cuatro de ellos.

Considerando el mecanismo de cruce de nivel, Årzén sugirió actualizar la señal de control cuando el valor absoluto de la diferencia entre el error actual medido y el error pasado cruza el límite q_{nom} , $abs(e(t_a) - e(t_{a-1})) > q_{nom}$, éste es modificado por una nueva propuesta, y por eso se propone cambiar el mecanismo de detección de eventos, sustituyendo la condición relativa por la absoluta. Una nueva señal de control es calculada tan pronto como el error actual medido cruza el nivel de detección $abs(e(t_a)) > q_{nom}$. Con este método el número de muestreos aumenta inevitablemente durante los transitorios, pero el error entre el sistema y la referencia es seguro que sea menor que q_{nom} durante el periodo de estado estacionario. Este método es requerido por controladores basados en eventos sin condición límite de seguridad, para asegurar que un estado estacionario es realmente activado antes de decidir no actualizar la señal de control mientras la referencia no cambia, o no ocurre una perturbación. Como un resultado la señal medida y tiene que rastrear la referencia esperada r durante el intervalo de estado estacionario, que significa un pequeño error (y no un pequeño error relativo, por que el error relativo podría ser pequeño, $abs(e(t_a) - e(t_{a-1})) > q_{nom}$, mientras el sistema esté lejos de la referencia).

En la primera estrategia propuesta en [20] solo es quitada la condición de límite de seguridad h_{max} . La segunda es donde la parte integral se modifica, para reducir el impacto después de un largo intervalo de estado estacionario, que es llamada saturación de la ganancia integral. La tercera es la del factor exponencial olvidado del intervalo de muestreo. La cuarta es una estrategia híbrida, entre la segunda y tercera estrategia. Las aproximaciones propuestas son de algún modo similares al mecanismo anti-windup, donde el error inducido por la saturación tiene que ser compensado. Finalmente, el procedimiento para implementar tal control es el mismo para las diferentes propuestas. El algoritmo resultante está representado en la figura 3.4, donde la ganancia integral $he(\cdot)$ depende de la estrategia escogida.

3.2.1. Algoritmo 1: Únicamente sin condición de límite de seguridad

Este algoritmo corresponde al de Årzén donde la condición de límite de seguridad $h(t_a) \geq h_{max}$ es removida. Para este primer algoritmo, la parte integral queda expresada en la ecuación 3.3.

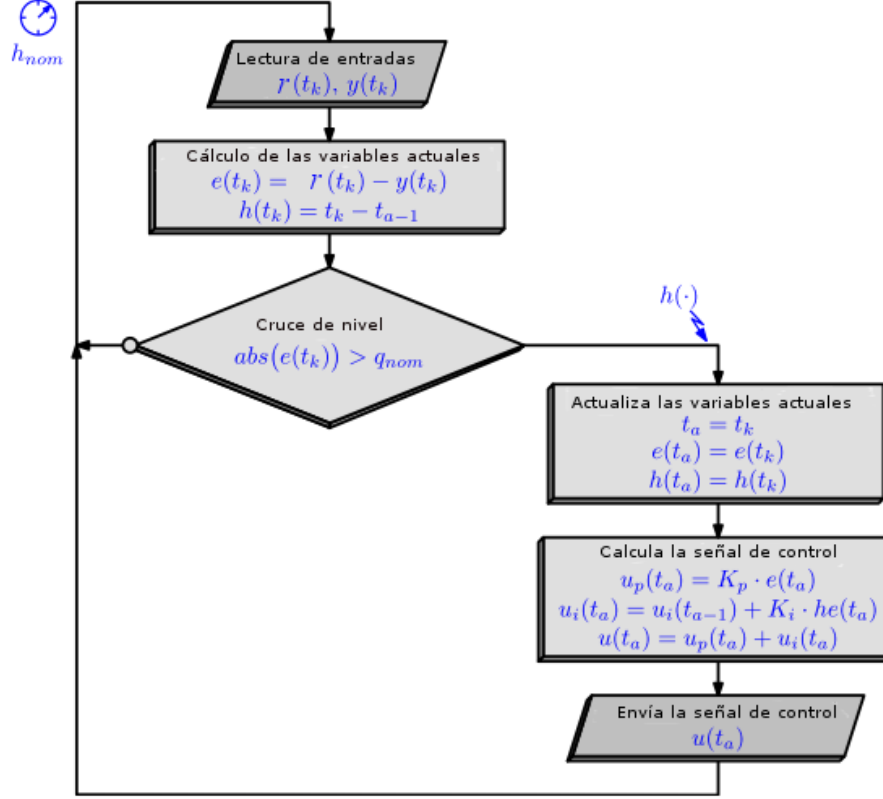


Figura 3.4: Algoritmo de los controladores PID basados en eventos sin condición de límite de seguridad.

$$u_i(t_a) = u_i(t_{a-1}) + K_i \cdot h e(t_a) \quad (3.3)$$

Donde $h e(t_a) = h(t_a) \cdot e(t_a)$

Se esperan importantes sobre tiros después de un largo intervalo de estado estacionario [20].

3.2.2. Algoritmo 2: Saturación de la ganancia integral

Este segundo algoritmo consiste en reducir el producto $h e(\cdot)$ después de un largo intervalo de estado estacionario, en orden para reducir los sobre-impulsos. De esta manera la ganancia integral es limitada conforme al principio descrito en la introducción de ésta sección. Se observa que la parte integral de este algoritmo viene de la ecuación 3.2. La parte integral queda definida por la ecuación 3.4 [20].

$$u_i(t_a) = u_i(t_{a-1}) + K_i \cdot h e_{sat}(t_a) \quad (3.4)$$

Donde $h e_{sat}(t_a) = (h(t_a) - h_{nom}) \cdot q_{nom} + h_{nom} \cdot e(t_a)$

3.2.3. Algoritmo 3: Factor exponencial olvidado del intervalo de muestreo

Otro método consiste en reducir el impacto del intervalo de muestreo que principalmente aumenta durante el intervalo de estado estacionario. Por eso se agrega el factor olvidado del intervalo

CAPÍTULO 3. CONTROL PID BASADO EN EVENTOS
3.2. NUEVAS ESTRATEGIAS DE CONTROL PID BASADO EN EVENTOS

de muestreo. Se elige una función exponencial para disminuir este impacto cuando el tiempo del estado estacionario aumenta, que está dada por la ecuación 3.5 [20].

$$h_{exp}(t_a) = h(t_a) \cdot \exp(\alpha \cdot (h_{nom} - h(t_a))) \quad (3.5)$$

Donde el parámetro α permite un grado de libertad o disminuye el intervalo exponencial de muestreo. Esto produce un intervalo de muestreo exponencial $h_{exp}(\cdot)$ en la parte integral del controlador como el siguiente:

$$\begin{aligned} u_i(t_a) &= u_i(t_{a-1}) + K_i \cdot h_{exp}(t_a) \\ \text{Donde } h_{exp}(t_a) &= h_{exp}(t_a) \cdot e(t_a) \end{aligned} \quad (3.6)$$

Esta función permite tener *i*) un intervalo de muestreo nominal durante los transitorios, donde $h(t_a) = h_{nom}$, el intervalo de muestreo exponencial es $h_{exp}(t_a) = h_{nom}$ y *ii*) una disminución exponencial del intervalo de muestreo durante los intervalos de muestreo de estado estacionario.

3.2.4. Algoritmo 4: Estrategia Híbrida

Finalmente este algoritmo es una mezcla entre la saturación de la ganancia integral y el factor exponencial olvidado del intervalo de muestreo. En realidad, en el primer algoritmo el producto $he(\cdot)$ aumenta con respecto al intervalo de muestreo $h(\cdot)$ y el error medido $e(\cdot)$ como es descrito en la figura 3.5(a). El segundo algoritmo minimiza el impacto del producto $he(\cdot)$ en el término integral, limitando la ganancia integral; pero todavía aumenta con respecto a $h(\cdot)$ y $e(\cdot)$, como se muestra en la figura 3.5(b). Finalmente el tercer algoritmo agrega un factor exponencial olvidado del intervalo de muestreo en tal forma que la ganancia integral disminuye. Sin embargo, el producto $he(\cdot)$ es más grande cuando el intervalo de muestreo es pequeño, como se ve en la figura 3.5(c). La idea es tener un menor impacto del intervalo de muestreo todo el tiempo. Por esa razón, el factor exponencial olvidado dado en la ecuación 3.5 se usa dentro del algoritmo de saturación [20]. Así el producto $he(\cdot)$ es ahora limitado superiormente por $(h_{exp}(t_a) - h_{nom}) \cdot q_{nom} + h_{nom} \cdot e(t_a)$ y la parte integral llega a ser:

$$\begin{aligned} u_i(t_a) &= u_i(t_{a-1}) + K_i \cdot h_{hybrid}(t_a) \\ \text{Donde } h_{hybrid}(t_a) &= (h_{exp}(t_a) - h_{nom}) \cdot q_{nom} + h_{nom} \cdot e(t_a) \end{aligned} \quad (3.7)$$

La evolución de la ganancia integral de este algoritmo híbrido está graficado en la figura 3.5 (d).

3.2.5. Integral absoluta como índice de desempeño del error

Cada una de las cuatro estrategias anteriormente descritas se pueden evaluar por tres diferentes integrales absolutas [23].

La primera es la integral IAE, da la información del seguimiento de la señal de referencia.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (3.8)$$

La segunda es la integral IAEP, compara la respuesta del sistema de control basado en tiempo con la del control basado en eventos.

$$IAE = \int_0^{\infty} |y_{basada_tiempo}(t) - y_{basada_eventos}(t)| dt \quad (3.9)$$

La tercera integral IAD, compara el índice IAE basado en tiempo con el basado en en eventos.

$$IAE = \int_0^{\infty} |IAE_{basado_tiempo}(t) - IAE_{basado_eventos}(t)| dt \quad (3.10)$$

CAPÍTULO 3. CONTROL PID BASADO EN EVENTOS
3.2. NUEVAS ESTRATEGIAS DE CONTROL PID BASADO EN EVENTOS

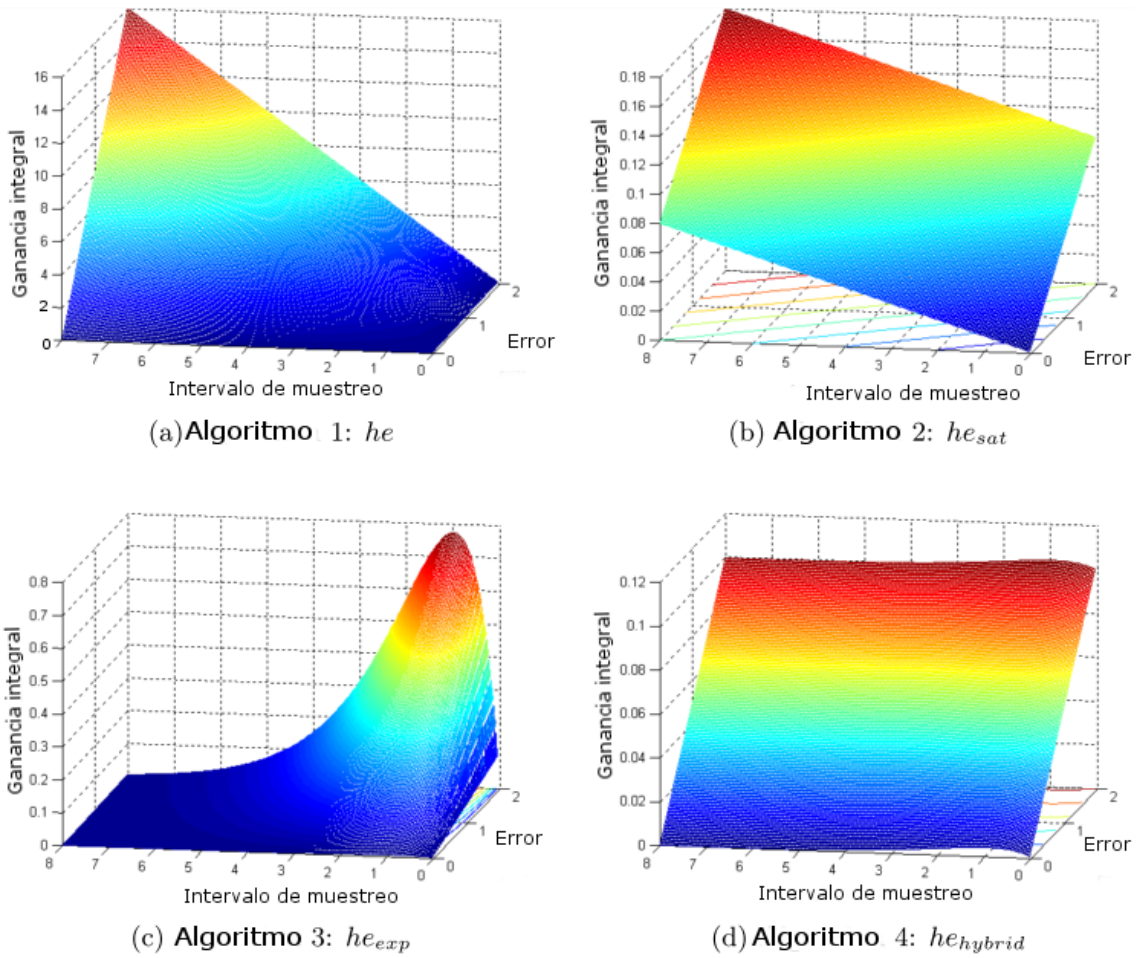


Figura 3.5: Dinámica de la ganancia integral ($h_{nom} = 0,05s, q_{nom} = 0,01, \alpha = 1$) de las estrategias de control basado en eventos sin condición de límite de seguridad.

Capítulo 4

Implementación y estudio del desempeño de las estrategias de control PI basadas en eventos, en un sistema de control retroalimentado por una red

De acuerdo con los sistemas lineales de primer orden analizados en la sección 2 del capítulo 1, es posible ver que ellos permiten representar la dinámica de una gran cantidad de sistemas que encontramos en la naturaleza, tanto en el ámbito científico como tecnológico. Por tal motivo, y sin pérdida de generalidad, en este capítulo es abordado el modelo matemático que representa la dinámica de la velocidad del motor DC, es tomado el atrevimiento de suponer que los resultados encontrados para el control de este sistema, pueden ser aplicados a otros sistemas más o menos complejos que se representen por una ecuación diferencial de primer orden.

Este capítulo estudia el motor de DC controlado por un control PI que en lazo cerrado resulta un *sistema de segundo orden*, éste cuenta con *características de desempeño* bien establecidas, específicamente se analizan el sobre impulso máximo (M_p) y el tiempo de asentamiento (t_s). Para controlar la velocidad del motor de DC se usa la estrategia de control basada en tiempo, la estrategia de control basada en eventos propuesta por K.E.Árzén [6] y las cuatro nuevas estrategias de control basado en eventos propuestas por S. Durand [20].

En particular se describe la obtención de la función de transferencia de la velocidad del motor de DC seleccionado. Es descrita la forma en que se sintoniza el control PI tomando como base la respuesta transitoria de un sistema de segundo orden a la entrada escalón unitario. Se muestra como se implementa en diagrama a bloques del control PI con la herramienta TRUE TIME de MATLAB/ SIMULINK. También se describe como se implementa en simulación cada una de las estrategias (basada en tiempo y basadas en eventos) de control PI para controlar la velocidad del motor de DC. Se realiza el estudio de cada una de las estrategias de control (basada en tiempo y basadas en eventos), se evalúa la condición de detección de eventos relativa y la absoluta. Y por último se prueba que tan robusta (perturbaciones en la red y perturbaciones externas) es la estrategia de control PI basada en tiempo, la estrategia de control PI basada en eventos de Árzén, y la mejor de las nuevas estrategias de control PI basado en eventos [20].

Cada estrategia de control PI será evaluada gráficamente, con las características de desempeño y con los índices: IAE, IAEP, IAD. También es evaluado el número de mensajes enviados por el nodo de control (Menv), y la duración aproximada en días de la batería (Dtb) al voltaje mínimo de

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI

4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

funcionamiento de un micro controlador. Con gráficas comparativas y tablas se muestra cual es la mejor estrategia de control, las abreviaturas utilizadas se describen en el apéndice C.

4.1. Identificación y control del sistema

El sistema a implementar en simulación en TRUE TIME de MATLAB/ SIMULINK, es un control PI lazo cerrado para controlar la velocidad de un motor de DC. La retroalimentación es mediante una red de comunicación CAN-BUS. Para poder llevar a cabo una buena sintonización del control, lo primero es identificar de manera experimental los parámetros del sistema a controlar, método que se describe en la siguiente sección.

4.1.1. Identificación de la función de transferencia de un sistema de primer orden

La caracterización del motor de DC se hace de una forma práctica, mediante un encoder y un circuito electrónico, que proporciona en términos de voltaje la velocidad del motor. La vista en el osciloscopio de la respuesta de este sistema se muestra en la figura 4.1.



Figura 4.1: Vista en el osciloscopio de la respuesta de un motor de DC, al alimentarlo con 12v.

La ecuación dinámica del motor de DC, es vista en la ecuación 4.1. ω es la velocidad del motor, u es la señal de control, K_m es la constante de par motor, R es la resistencia de la armadura, J_m es la inercia del rotor.

$$\dot{\omega} = -\frac{K_m^2}{J_m R} \omega + \frac{K_m}{J_m R} u \quad (4.1)$$

Aplicando la transformada de laplace a la ecuación 4.1 se obtiene una función de transferencia de primer orden para el motor de DC, esta función de transferencia se ve en la ecuación 4.2.

$$H(S) = \frac{K}{\tau S + 1} \quad (4.2)$$

Las constantes K y τ se obtienen experimentalmente. En las siguientes líneas se describe la obtención de cada una de las constantes:

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

Para calcular la constante K se toma en cuenta que la frecuencia de giro del motor está dada por la relación de la frecuencia del encoder que es la que indica el osciloscopio entre el número de ranuras del encoder, esto se muestra en la ecuación 4.3.

$$\frac{300Hz}{90} = 3 \cdot 3Hz \quad (4.3)$$

El resultado se convierte a radianes sobre segundo de la siguiente manera:
(3 · 3Hz)(2π) = 20 · 7345rad/s.

La velocidad de un motor se define por la ecuación 4.4 de primer orden.

$$\bar{\Omega} = KA \quad (4.4)$$

Donde $\bar{\Omega}$ es la velocidad del motor, K es una constante que también aparece en la ecuación 4.2, 'A' representa el voltaje con el que es alimentado el motor.

De la ecuación 4.4 se despeja la constante K y queda la ecuación 4.5.

$$K = \frac{\bar{\Omega}}{A} \quad (4.5)$$

Se sustituye la velocidad en términos de rad/s y 12v que es el valor de 'A' en la ecuación 4.5 y se obtiene que K=1.728.

La constante τ , representa el tiempo en que el motor alcanza el 63,2% de su velocidad máxima. Partiendo de que la velocidad máxima del motor es representada en el osciloscopio por el voltaje máximo, se encuentra que al aplicar el máximo voltaje al motor (12v), la señal de respuesta alcanza 4v. El 63,2% de 4v es 2.528v que es alcanzado en 114ms figura 4.1, de aquí se obtiene que $\tau = 0,114s$.

Se sustituyen los valores de K y τ en la ecuación 4.2, y se obtiene la función de transferencia que representa la velocidad del motor, se muestra en la ecuación 4.6.

$$H(S) = \frac{1 \cdot 728}{0 \cdot 114S + 1} \quad (4.6)$$

Se divide la ecuación 4.6 entre 0.114, y se obtiene la ecuación 4.7.

$$H(S) = \frac{15 \cdot 3}{S + 8 \cdot 772} \quad (4.7)$$

4.1.2. Sintonización del controlador PI

Un controlador PI se usa para controlar la velocidad del motor de DC, el diagrama a bloques se muestra en la figura 4.2. Donde Vd es la velocidad deseada, el primer bloque representa el control PI, u(S) es la señal de control del controlador PI que es aplicada al motor de DC, Vm es la velocidad medida. El sistema es retroalimentado y muestreado con un tiempo constante, el error es la entrada del controlador PI.

Mediante el algebra de diagramas de bloques se obtiene la función de transferencia del sistema de control en lazo cerrado que se ve en la ecuación 4.8.

$$H_{LC}(S) = \frac{K_P(ST_i + 1)15 \cdot 3}{T_i(S^2 + S(8 \cdot 772 + K_P15 \cdot 3) + \frac{K_P15 \cdot 3}{T_i})} \quad (4.8)$$

La función de transferencia en lazo cerrado de la respuesta transitoria de un sistema de segundo orden con una entrada escalón, está dada por la ecuación 4.9 [35].

$$H_{LC}(S) = \frac{W_n^2}{S^2 + 2\zeta W_n S + W_n^2} \quad (4.9)$$

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

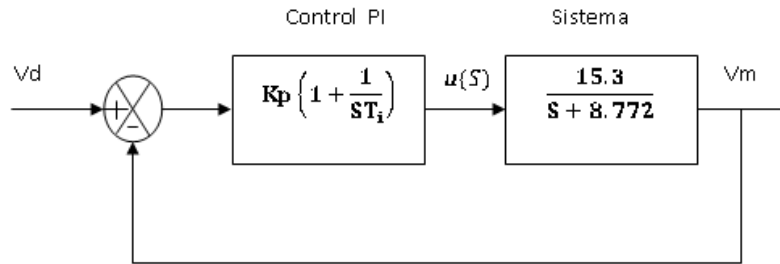


Figura 4.2: Diagrama a bloques del sistema de control de velocidad del motor de DC.

Igualando las ecuaciones 4.8y 4.9, se obtienen las ecuaciones 4.10y 4.11, de estas se obtiene el valor de K_p y T_i .

$$8 \cdot 772 + K_P 15 \cdot 3 = 2W_n \zeta \quad (4.10)$$

$$\frac{K_P 15 \cdot 3}{T_i} = W_n^2 \quad (4.11)$$

El valor que se toma para el factor de amortiguamiento es $\zeta = 0,707$. El tiempo de asentamiento t_s está dado por la ecuación 4.12(criterio del 2%)[35], en esta ecuación se sustituye el valor de ζ y el tiempo de asentamiento t_s de 1 segundo.

$$t_s = \frac{4}{\zeta W_n} \quad (4.12)$$

De la ecuación 4.12 se despeja la frecuencia natural W_n , y se encuentra que $W_n = 5 \cdot 6577$. Los valores de ζ y W_n se sustituyen en las ecuaciones 4.10 y 4.11 y se encuentra que las ganancias del controlador PI toman el valor de: $T_i = -0 \cdot 0241$ y $K_p = -0 \cdot 0505$.

Aplicando el teorema de valor final a la función de transferencia obtenida, se tiene que la velocidad maxima que puede alcanzar el motor, es $20,93rad/s$.

4.1.3. Implementación en diagrama a bloques del sistema de control PI retroalimentado por red

El diagrama a bloques se implementa con la herramienta TRUE TIME de MATLAB/ SIMULINK. El sistema tiene salidas a pantallas y gráficas que sirven para medir las variables que posteriormente serán estudiadas. La implementación del sistema de control lazo cerrado se muestra en la figura 4.3.

El diagrama a bloques de la red se describe a continuación:

- **Bloque de red:**La red se configura con un número de 4 nodos, una velocidad de datos de 80000bits/s, tamaño mínimo de 40 bits para la armadura de datos.
- **Bloque del sistema:** Este bloque es la función de transferencia del motor de DC, como entrada tiene la señal que viene del nodo actuador, su salida está conectada al nodo sensor.
- **Bloque del nodo de interferencia:** Éste se encarga de enviar una señal de interferencia a la red, que afecta el ancho de banda de la red según el porcentaje que se elija. Se verán mayormente afectados los sistemas a medida que se incrementa el porcentaje del ancho de banda de este nodo en la red. Así se podrá evaluar que tan robusta es cada una de las estrategias de control.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

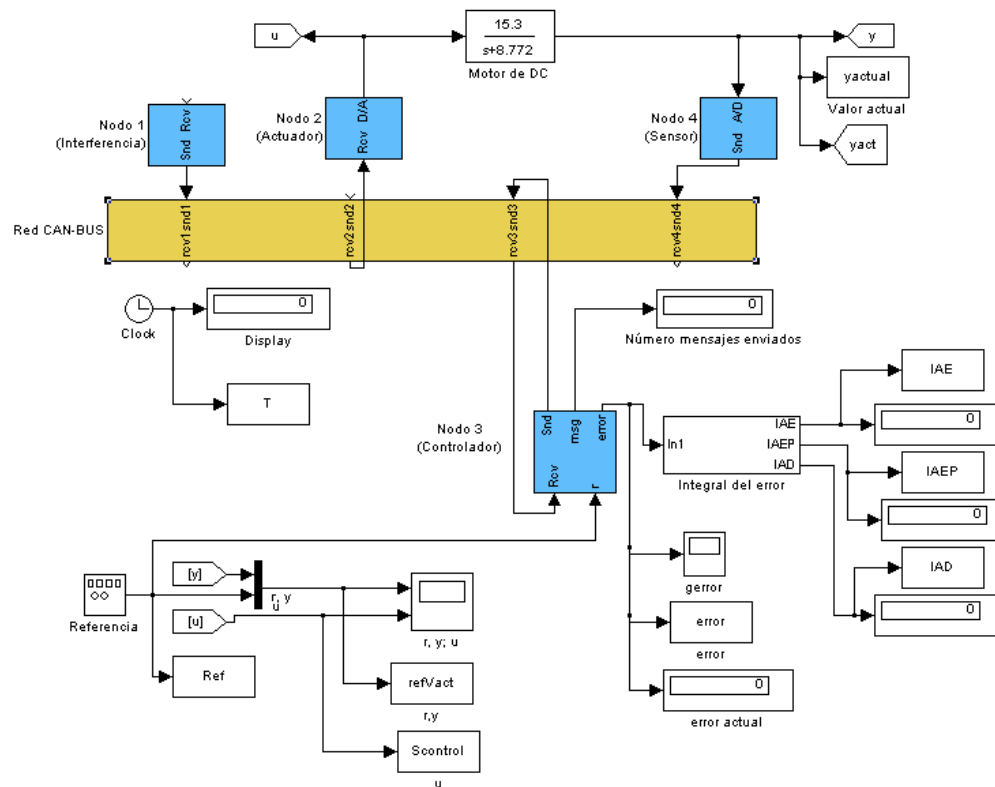


Figura 4.3: Diagrama a bloques del control de la velocidad de un motor de DC, sistema lazo cerrado por una red CAN-BUS.

- **Bloque del nodo actuador:** Este nodo recibe la señal de control enviada por el nodo control y la aplica al sistema.
- **Bloque del nodo sensor:** Mide la señal del sistema a evaluar y la envía al nodo control por medio de la red.
- **Bloque del nodo control:** Este nodo calcula la señal de control, aquí es donde se aplican cada una de las estrategias de control PI a evaluar.
- **Bloque de la integral del error:** El diagrama interno de este bloque se muestra en la figura 4.4. Se calculan los tres índices IAE, IAEP, IAD.

4.1.4. Implementación en código de la estrategia de control PI basado en tiempo

El código de función para la estrategia de control basada en tiempo se muestra a continuación.

En el apéndice A se describen los comandos utilizados.

```
function [exectime, data] = ctrlcode(seg, data)
```

```
switch seg,
```

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

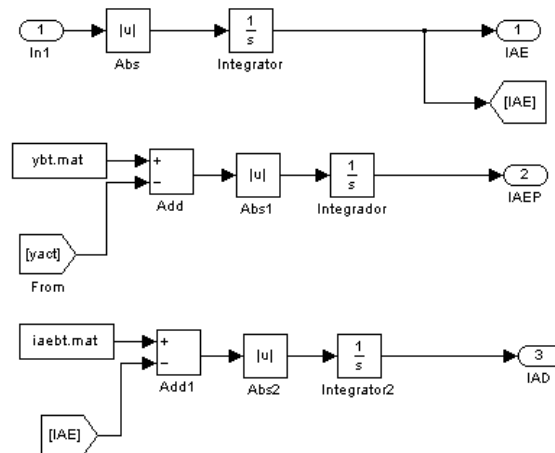


Figura 4.4: Diagrama a bloques del sub sistema del bloque *integral del error*.

case 1,

```
y = ttGetMsg; % Obtiene de la red la señal actual del sistema.
r = ttAnalogIn(1); % Obtiene de la entrada del bloque control la señal de referencia.

data.e=r-y; % Calcula el valor actual del error.
```

```
P = data.K*(r-y); % Calcula la parte proporcional del controlador.
I = data.Iold ; % Guarda el valor anterior de la parte integral del controlador.
```

```
data.c = P + I; %Calcula la señal de control.
```

```
% Calcula la señal saturada del control.
```

```
if data.c>12
data.u=12;
elseif data.c< -12
data.u=-12;
else data.u=data.c;
end
```

```
% Calcula la parte integral del controlador y le suma el anti-windup.
```

```
data.Iold = data.Iold + (data.K*data.h*(r-y))/data.Ti +(data.K*data.h*data.ks
*(-data.c+data.u))/data.Ti;
```

```
data.cont= data.cont+1; % Cuenta la cantidad de veces que funciona el control.
exectime = 0.0005;
```

case 2,

```
ttAnalogOut(1, data.cont); % Saca la señal del contador a la salida msg del bloque control.
ttAnalogOut(2, data.e); % Saca la señal de error a la salida error del bloque de control.
```

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

Variable	Descripción
r	Valor de la señal de referencia.
y	Valor actual de la respuesta del sistema.
data.e	Valor actual del error.
P	Parte proporcional del controlador.
data.K	Constante de la ganancia proporcional del controlador. Su valor es de -0.0505.
I	Parte integral del controlador.
data.old	Valor anterior de la parte integral del controlador.
data.c	Valor de la señal de control.
data.u	Valor saturado de la señal de control.
data.h	Constante del periodo del controlador. Está ajustada a 0.008.
data.Ti	Constante de la ganancia integral del controlador. Su valor es de - 0.0241.
data.ks	Constante de saturación de la señal de control. Su valor es de 8.
data.cont	Cantidad de veces que se calculó la señal de control.

Tabla 4.1: Descripción de las variables para implementar la estrategia de control PI basado en tiempo, para controlar la velocidad del motor de DC.

```
ttSendMsg(2, data.u, 80); % Envía la señal saturada del error al bloque del nodo actuador.
exectime = -1; % Finaliza
```

```
end
```

Las variables se describen en la tabla 4.1.

4.1.5. Implementación en código de las estrategias de control PI basado en eventos

Las estrategias de control PI basado en eventos [7],[11] que se implementan en simulación son:

Estrategia propuesta por Årzén.

Estrategia sin condición de límite de seguridad.

Estrategia de saturación de la ganancia integral.

Estrategia del factor exponencial olvidado del intervalo de muestreo.

Estrategia Híbrida.

El código de implementación de las estrategias de control PI basado en eventos es muy similar al de la estrategia basada en tiempo. En el apéndice A se describen los comandos utilizados.

Para comprender como se implementa cada una de las estrategias, a continuación se muestra el **código de la estrategia de saturación de la ganancia integral**:

```
function [exectime, data] = ctrlcode(seg, data)
```

```
switch seg,
```

```
case 1,
```

```
y = ttGetMsg; % Obtiene de la red la señal actual del sistema.
```

```
r = ttAnalogIn(1); % Obtiene de la entrada del bloque control la señal de referencia.
```

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

```
data.e=r-y;% Calcula el valor actual del error.

elim=abs(0.01*r); % Calcula el valor actual del error límite. El valor 0.01 es para obtener el
porcentaje de la señal de referencia.
data.hact=data.hact+data.h;% Calcula el valor del periodo actual.

if (abs(data.e)>elim) % Condición de detección de eventos.

P = data.K*(data.beta*data.e);%Calcula la parte proporcional del control PI.

% Código exclusivo de la estrategia saturación de la ganancia integral

if data.hact>=data.hmax
data.he=(data.hact-data.h)*elim+data.h*data.e;
else
data.he=data.hact*data.e;
end

I = data.Iold;% Calcula la parte integral del controlador.

data.c = P + I;%Calcula la señal de control.

% Satura la señal de control a 12.

if data.c>12
data.u=12;
elseif data.c<-12
data.u=-12;
else data.u=data.c;
end

% Calcula la parte integral del controlador.

data.Iold = data.Iold + (data.K*data.he)/data.Ti + (data.K*data.h*data.ks*
(-data.c+data.u))/data.Ti;

data.es=data.e;% Actualiza el valor anterior del error.
data.cont= data.cont+1; % Incrementa el valor del contador.
data.hact=0; % Inicializa el valor del periodo actual.
ttSendMsg(2, data.u, 80);% Envía la señal de control saturada al bloque del nodo control.

end

exectime = 0.0005;
case 2,

ttAnalogOut(1, data.cont);% Saca la señal del contador a la salida msg del bloque control.
ttAnalogOut(2, data.e);% Saca la señal de error a la salida error del bloque de control.
exectime = -1;
end
```

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI

4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

En el escrito del código anterior **se resaltan las partes que cambian en cada estrategia**. A continuación se enumeran las partes donde cambia la implementación de cada estrategia de control PI basado en eventos del motor de DC:

1. Estrategia de Árzén

- *Condición de detección de eventos:*

```
if (abs(data.e-data.es)>elim)||((data.hact>=data.hmax)
```

- *Código exclusivo de la estrategia:* Esta estrategia no lleva ningún código exclusivo.
- *Parte integral del controlador:*

```
data.Iold = data.Iold + (data.K*data.hact*data.e)/data.Ti+  
+(data.K*data.h*data.ks*(-data.c+data.u))/data.Ti;
```

- *Valor anterior del error:* En esta estrategia si se actualiza el valor anterior del error.

2. Estrategia sin condición de límite de seguridad

- *Condición de detección de eventos:*

Puede ser la condición relativa :if (abs(data.e-data.es)>elim) ó la condición absoluta: if (abs(data.e)>elim)

- *Código exclusivo de la estrategia:*Esta estrategia no lleva ningún código exclusivo.
- *Parte integral del controlador:*

```
data.Iold = data.Iold + (data.K*data.hact*data.e)/data.Ti+  
(data.K*data.h*data.ks*(-data.c+data.u))/data.Ti;
```

- *Valor anterior del error:* El valor anterior del error solo se debe actualizar cuando se trabaja con la condición relativa, mientras que con la condición absoluta no se debe actualizar.

3. Estrategia de saturación de la ganancia integral

- *Condición de detección de eventos:*

Puede ser la condición relativa :if (abs(data.e-data.es)>elim) ó la condición absoluta: if (abs(data.e)>elim)

- *Código exclusivo de la estrategia:*

```
if data.hact>=data.hmax  
data.he=(data.hact-data.h)*elim+data.h*data.e;  
else  
data.he=data.hact*data.e;  
end
```

- *Parte integral del controlador:*

```
data.Iold = data.Iold + (data.K*data.he)/data.Ti+ (data.K*data.h*data.ks*(-  
data.c+data.u))/data.Ti;
```

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI

4.1. IDENTIFICACIÓN Y CONTROL DEL SISTEMA

- *Valor anterior del error:* El valor anterior del error solo se debe actualizar cuando se trabaja con la condición relativa, mientras que con la condición absoluta no se debe actualizar.

4. Estrategia del factor exponencial olvidado del intervalo de muestreo

- *Condición de detección de eventos:*

Puede ser la condición relativa :if (abs(data.e-data.es)>elim) ó la condición absoluta: if (abs(data.e)>elim)

- *Código exclusivo de la estrategia:*

```
data.hact_i=data.hact*exp(data.h-data.hact);
```

- *Parte integral del controlador:*

```
data.Iold = data.Iold + (data.K*data.hact_i*data.e)/data.Ti+  
(data.K*data.h*data.ks*(-data.c+data.u))/data.Ti;
```

- *Valor anterior del error:* El valor anterior del error solo se debe actualizar cuando se trabaja con la condición relativa, mientras que con la condición absoluta no se debe actualizar.

5. Estrategia Híbrida

- *Condición de detección de eventos:*

Puede ser la condición relativa :if (abs(data.e-data.es)>elim) ó la condición absoluta: if (abs(data.e)>elim)

- *Código exclusivo de la estrategia:*

```
if data.hact>=data.hmax  
data.hact_i=data.hact*exp(data.h-data.hact);  
data.he=(data.hact_i-data.h)*elim+data.h*data.e;  
else  
data.he=data.hact*data.e;  
end
```

- *Parte integral del controlador:*

```
data.Iold = data.Iold + (data.K*data.he)/data.Ti+ (data.K*data.h*data.ks*(-  
data.c+data.u))/data.Ti;
```

- *Valor anterior del error:* El valor anterior del error solo se debe actualizar cuando se trabaja con la condición relativa, mientras que con la condición absoluta no se debe actualizar.

Las variables del código de la implementación de las estrategias de control PI basado en eventos en su mayoría son las mismas a las ocupadas en la implementación de la estrategia basada en tiempo, pero se añaden unas variables exclusivas, estas se muestran en la tabla 4.2.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

Variable	Descripción
elim	Valor del error límite.
data.hact	Valor del periodo actual.
data.hmax	Valor del periodo máximo.
data.es	Valor del error anterior.
data.he	Valor para saturar la ganancia integral.
data.hact_i	Factor exponencial olvidado del intervalo de muestreo

Tabla 4.2: Descripción de las variables exclusivas para implementar las estrategias de control PI basado en eventos , para controlar la velocidad del motor de DC.

4.2. Estudio de las estrategias de control PI

La estrategia de control basada en tiempo a dado excelentes resultados en las tareas de control que se le han encomendado, sus *características de desempeño* son muy buenas y el índice IAE. Pero cuando el sistema se retro alimenta con una red, y en esta red existe muchos más nodos que generan retraso en el envío de los paquetes de datos de la señal de control, la señal de control sufre un retraso que afecta en gran manera la respuesta del sistema a controlar.

En los últimos años el interés del área de control a aumentado por proporcionar una estrategia de control PID basada en eventos [20], donde la acción de control se realiza cuando es necesario. La estrategia basada en eventos debe tener las siguientes características: tener un buen resultado en sus *características de desempeño*, ser robusta ante perturbaciones en la red y perturbaciones externas, tener excelentes valores en los índices IAE, IAEP, IAED, reducir las veces que se actualiza la señal de control se ve reflejada en el uso del micro controlador, cantidad de mensajes que envía el nodo control, consumo de energía y costos. Las características anteriormente mencionadas se evalúan en la respuesta del sistema de primer orden con cada una de las estrategias de control PI(estrategia basada en tiempo, estrategia basada en eventos de Árzén , y las cuatro nuevas estrategias de control PI basado en eventos propuestas por Sylvain Durand [20]).

El diagrama a bloques mencionado en sub sección 4.1.3, es utilizado para el estudio de las estrategias de control PI basada en tiempo y basada en eventos. El tiempo total de simulación es 20s. La referencia es una señal escalón que va de 6 rad/s y en los 10s cambia a 17 rad/s.

Para estudiar el consumo de energía del sistema con cada una de las estrategias de control, el nodo control se alimenta con una batería virtual de 5v (Apéndice B) en TRUE TIME de MATLAB/SIMULINK, que en un sistema real esta batería alimentaría el micro controlador que es el nodo control de un sistema de control retroalimentado por una red. Solo se analiza el consumo de energía de la actualización de la señal de control cuando el nodo control envía mensajes a la red, a pesar de que el micro controlador también aumenta su consumo de energía cuando se actualiza la señal de control [37]. En cada estrategia de control es obtenido el valor Dtb, que es el valor aproximado en días del desgaste de la batería de 5V a 4.2V(Voltaje mínimo de funcionamiento a alta frecuencias de un micro controlador 18F458 de Microchip con modulo CAN-BUS).

Para llevar a cabo el estudio de las estrategias de control PI en un sistema retroalimentado por una red CAN-BUS, primero se estudia el caso nominal(0% en el ancho de banda del nodo de interferencias, y sin perturbaciones externas), se obtienen los resultados de la : estrategia basada en tiempo, estrategia de control basada en eventos de Árzén con las dos condiciones de detección de eventos(relativa y absoluta), los resultados de las cuatro nuevas estrategias de control PI propuestas por Sylvain Durand. Mediante gráficas y tablas se comparan los resultados de cada una de las estrategias de control PI, y se seleccionan las tres con mejores resultados.

Finalmente para comprobar que tan robusta es cada una de las tres mejores estrategias de control PI, se evalúan los resultados cuando el sistema de control retroalimentado por una red CAN-BUS tiene perturbaciones en la red y perturbaciones externas(81 % del ancho de banda del nodo de interferencias y con perturbaciones externas). Los resultados son comparados gráficamente.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1. Análisis caso nominal

4.2.1.1. Estrategia basada en tiempo

En la figura 4.5 se muestran las señales de respuesta de la estrategia de control PI basado e tiempo. En a) se observa la señal de referencia contra la señal actual. Cuando el valor de la referencia es igual a 6rad/s t_s pasa por 0.16s el valor de t_s de diseño, M_p es de 2.76%. Cuando el valor de referencia es igual a 17rad/s t_s es igual a 1s que es el valor de diseño de t_s , M_p disminuye a 1.76% que es menor que el criterio del 2%. b) Índice IAE, el valor máximo es 6.25. c) La cantidad de mensajes enviados (M_{env}) por el nodo control son 2000 mensajes durante los 20s que dura la simulación. d) Lo que tarda la batería para disminuir su voltaje a 4.2V (D_{tb}), son 5.3 días.

Los valores obtenidos con esta estrategia se compararán con las estrategias de control basado en eventos a estudiar, ya que la estrategia basada en tiempo se ha utilizado en los últimos años, y ha dado buenos resultados.

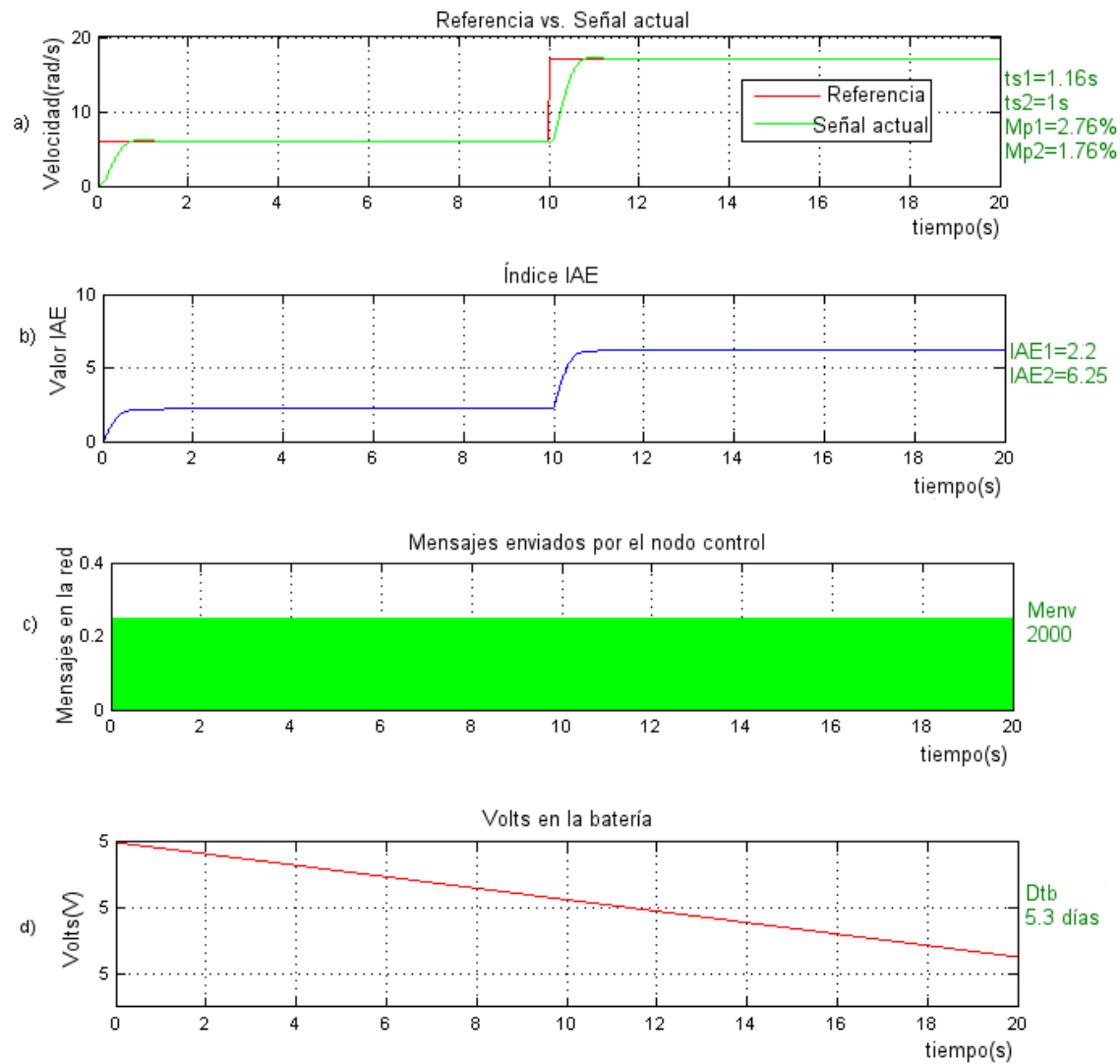


Figura 4.5: Señales de respuesta de la estrategia de control PI basada en tiempo.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1.2. Estrategia basada en eventos de Árzén

- Cuando el elim es 0.1 % del valor de la referencia, y hmax es 0.04s.

En la figura 4.6: a) Se observa que la señal actual de la condición de detección de absoluta y la condición de detección de eventos relativa van a la par, los valores de t_s y M_p , se incrementan muy poco respecto a los valores de diseño cuando el valor de la referencia es 6rad/s, ya cuando el valor de la referencia es 17rad/s el valor de t_s toma el valor de diseño y M_p está dentro del rango del criterio del 2% del error. La señal de control b) en ambos valores de la referencia toma valores por debajo de 12V que es el valor límite de la señal de control. En d) se muestra que la condición de detección de eventos relativa envía 104 mensajes menos que la absoluta c).

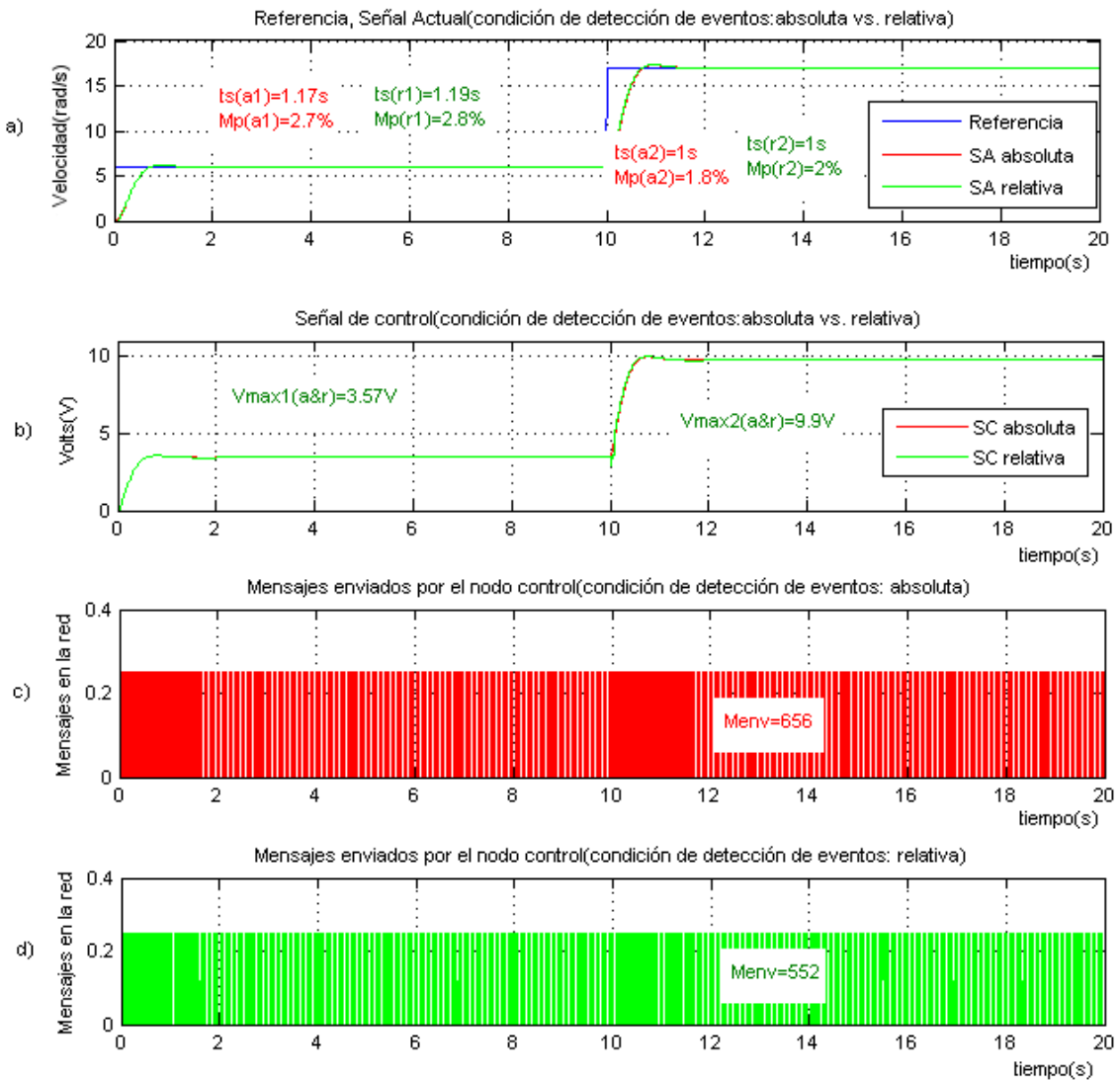


Figura 4.6: Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=0.04s.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

En la figura 4.7: En a) el valor de Dtb indica que con la condición relativa la batería dura aproximadamente 3 días más que con la condición absoluta, los índices IAE b) de ambas son casi iguales, pero en c) y d) comienza a notarse que los valores de los índices (IAEP, IAD) con la condición absoluta son menores.

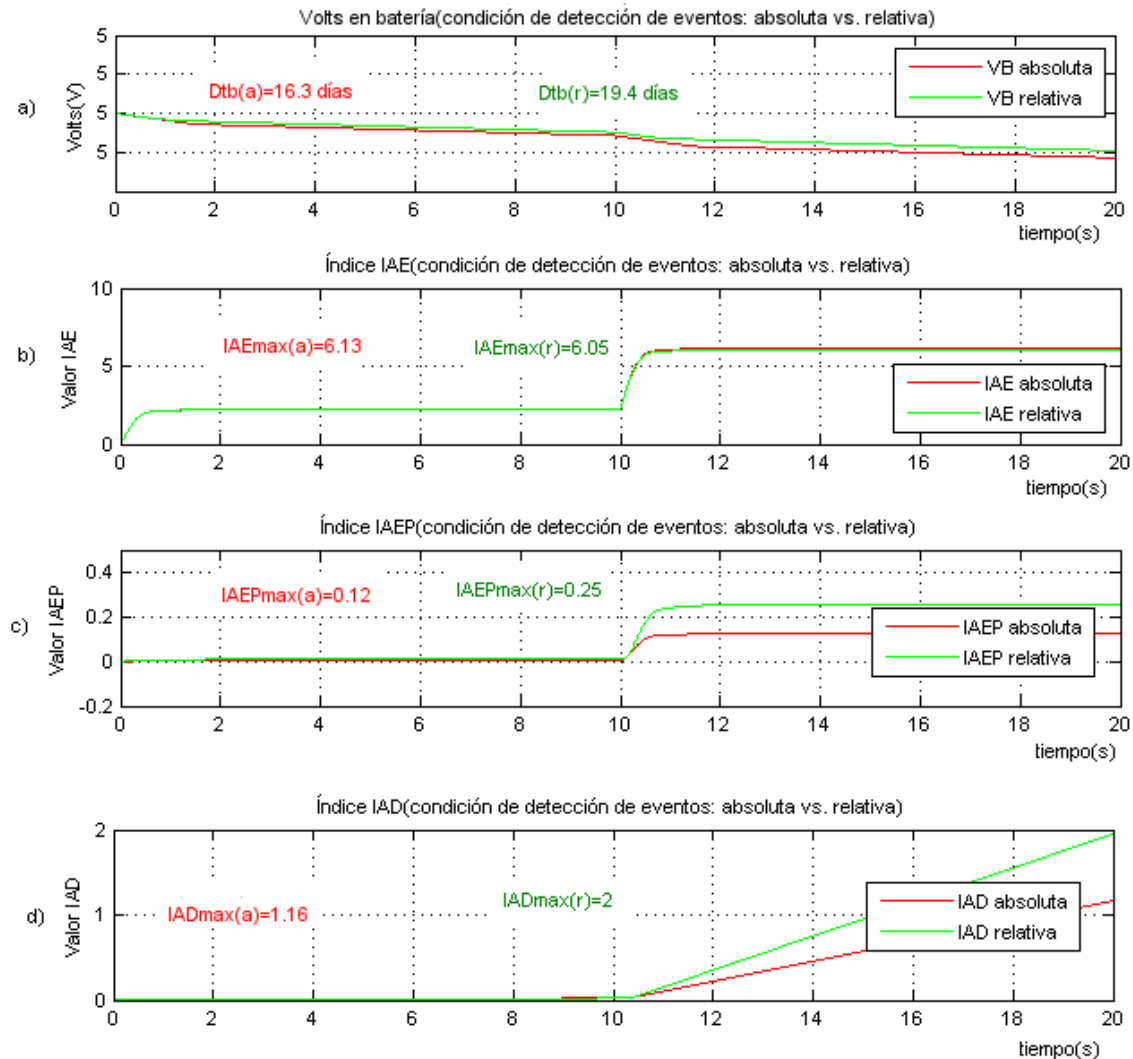


Figura 4.7: Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=0.04s.

Nota: Con estos valores de elim y hmax, parece indicar que la condición relativa es mejor por que envía menos mensajes a la red, la batería dura más, tiene el índice IAE más bajo. Pero los índices IAEP y IAD más bajos son de la condición absoluta.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

- Cuando el elim es 1 % del valor de la referencia, y hmax es 0.04s

En la figura 4.8: a) Cuando la señal de referencia es 6rad/s, en ambas condiciones el aumento de t_s es insignificante, pero el aumento de M_p es más notable en la condición relativa; cuando la referencia es 17rad/s t_s toma el valor de diseño y M_p queda dentro del rango del criterio del 2%. b) La señal de control se encuentra por debajo del límite, no se satura. En d) se muestra que con la condición relativa se envían 115 mensajes menos que con la condición absoluta c).

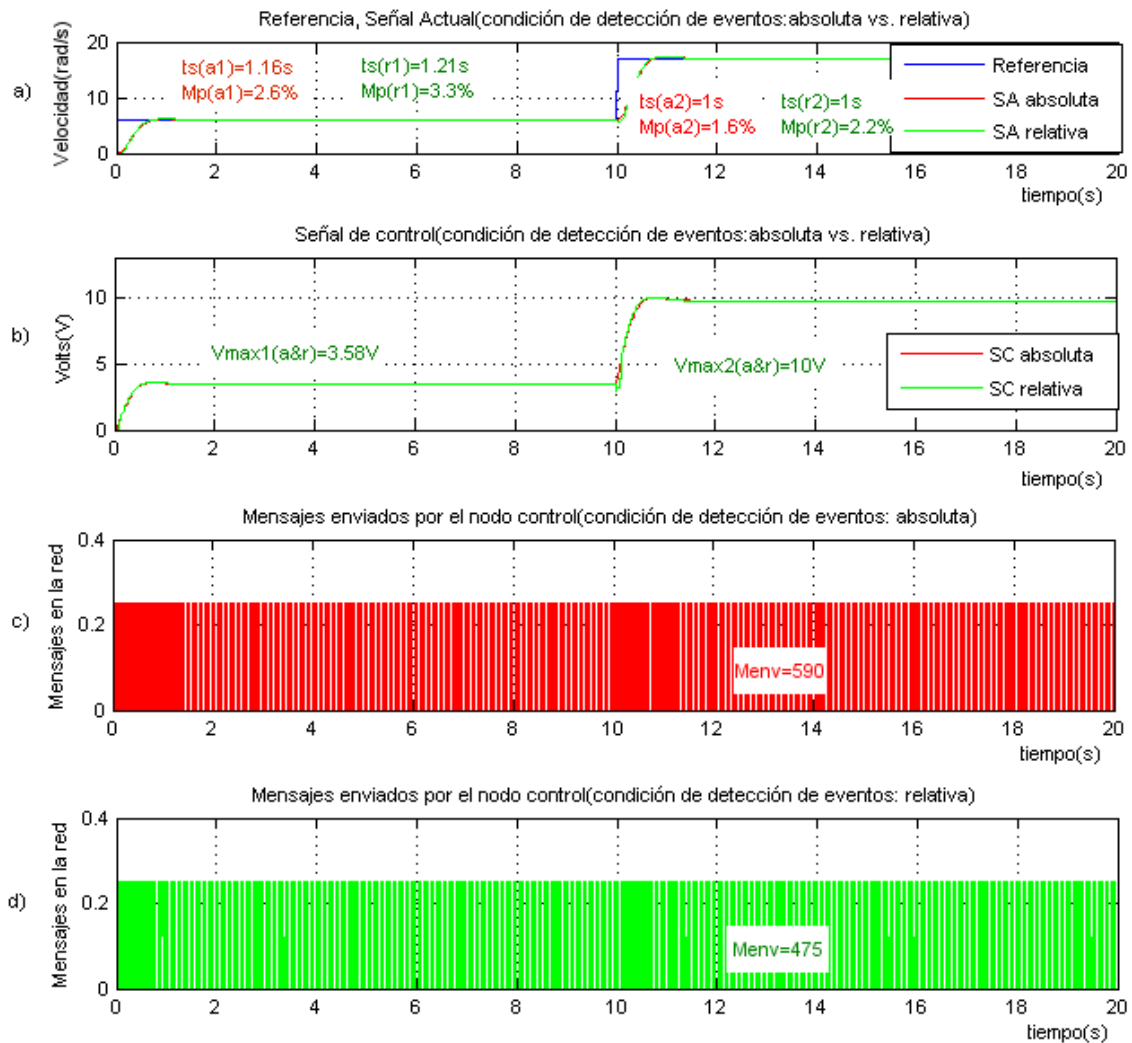


Figura 4.8: Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Arzén, cuando el elim es 1 % del valor de la referencia, y $h_{max} = 0.04s$.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

En la figura 4.9: a) Con la condición de detección de eventos relativa la duración de la batería (Dtb) es aproximadamente 4.5 días más que con la condición absoluta. b) En ambas condiciones de detección de eventos el índice IAE es casi el mismo, el de la condición absoluta es más chico por 0.11 que la condición relativa. c) El índice IAEP de la condición absoluta muestra ser más chico por 0.23 que de la condición relativa. d) El índice IAD con la condición relativa es más chico por 0.75 que con la condición absoluta.

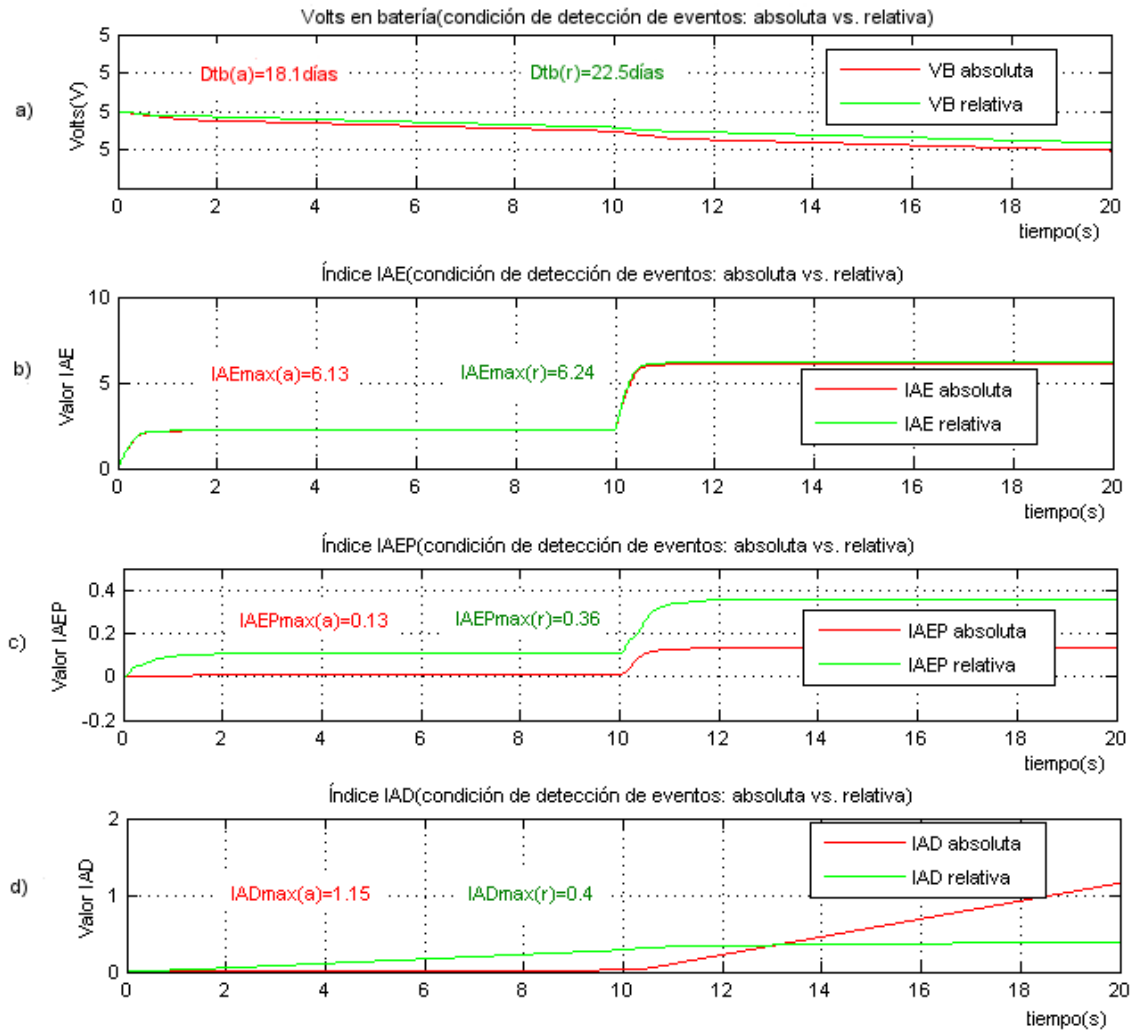


Figura 4.9: Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1% del valor de la referencia, y hmax=0.04s.

Nota: Al aumento de elim al 1% de la referencia y hmax permanece en 0.04s, el número de mensajes enviados (Menv) con ambas condiciones se reducen en comparación con los valores anteriores de elim=0.1% de la referencia y hmax=0.04s. Los índices IAE, IAEP, IAD son mejores con la condición absoluta, a pesar de que con la condición relativa sea menor la cantidad de mensajes enviados (Menv) a la red.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

- Cuando el elim es 0.1 % del valor de la referencia, y hmax es 1s

En la figura 4.10: a) El aumento de t_s y M_p es muy poco en ambas condiciones de detección de eventos cuando la señal de referencia es 6rad/s. Pero cuando la señal de referencia es 17rad/s la señal actual con la condición relativa comienza a oscilar de tal manera que t_s es indefinido, y la señal de control vista en b) tiene que ser saturada a 12v, y así M_p también aumenta desmedidamente. La condición absoluta con la referencia a 17rad/s también aumenta presenta un M_p del 18.8% que para nada es grato, pero su tiempo de asentamiento t_s es 1s, el valor de diseño. La condición absoluta envía 62 mensajes menos que la condición relativa.

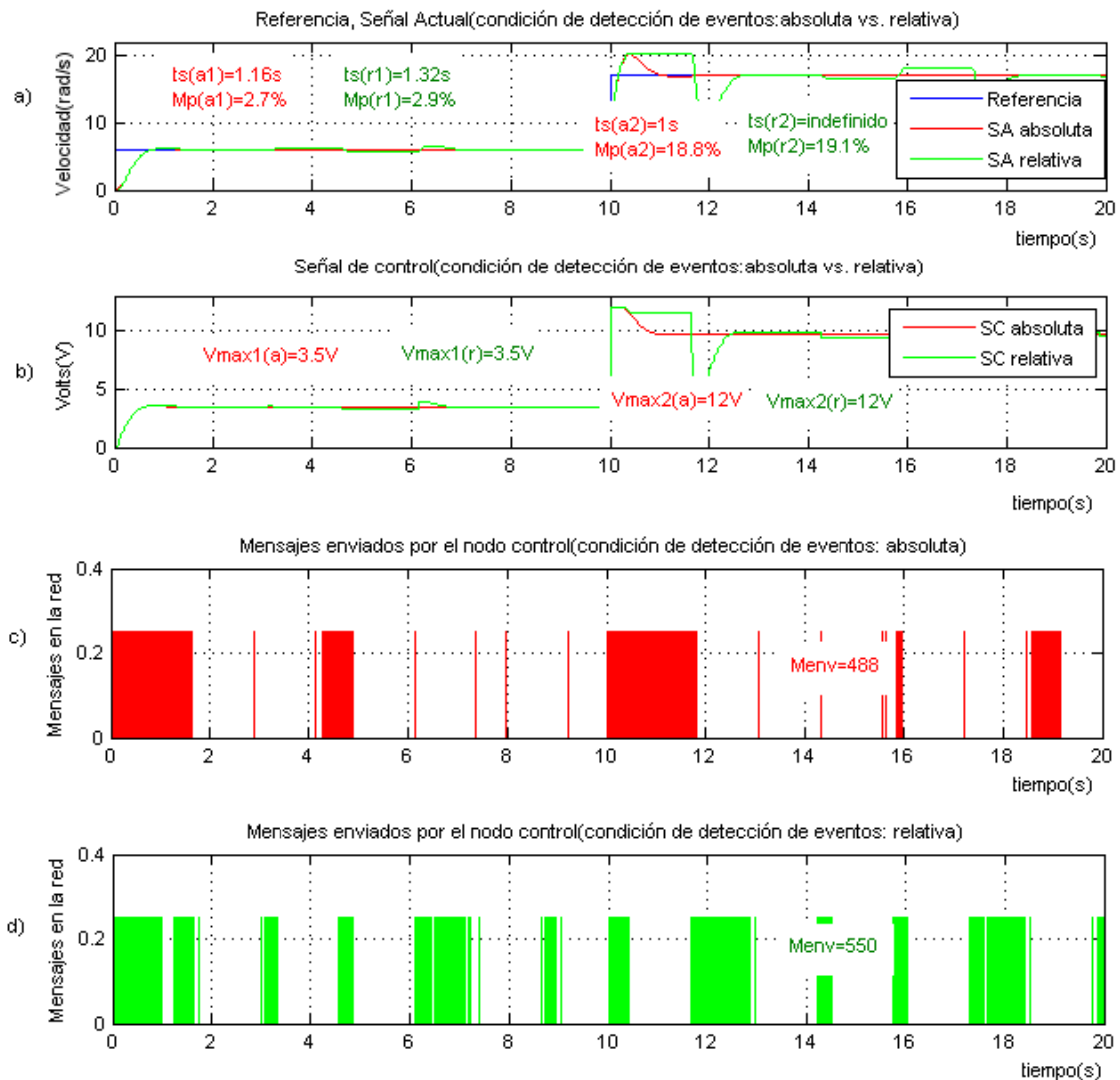


Figura 4.10: Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=1s.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

En la figura 4.11: a) La condición absoluta permite que Dtb tenga 2.5 días más que con la condición relativa. Los índices a)IAE, b)IAEP, c)IAD con la condición de detección de eventos relativa son más grandes que con la condición absoluta.

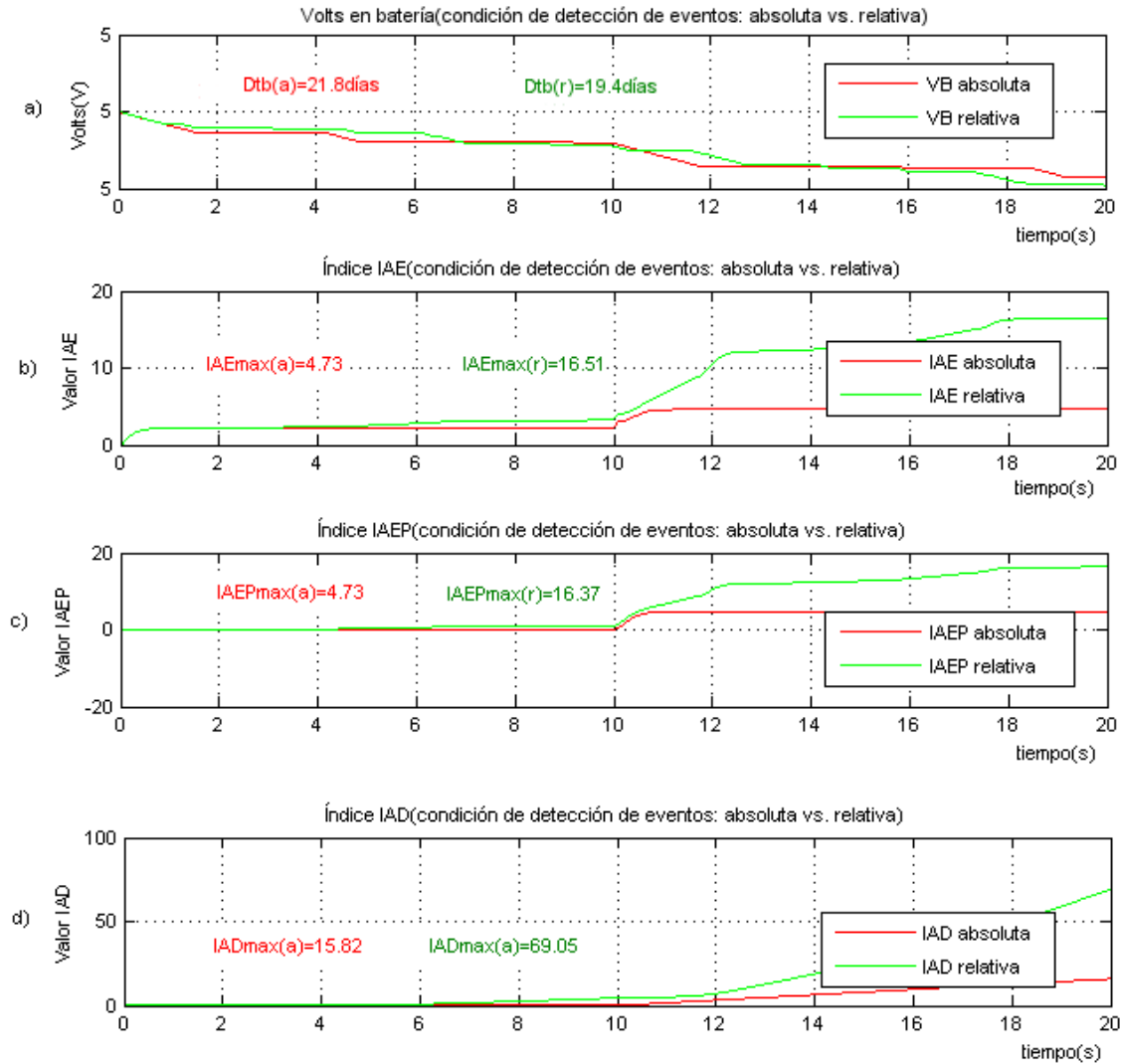


Figura 4.11: Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 0.1 % del valor de la referencia, y hmax=1s.

Nota: El aumento de hmax a 1s y elim igual al 0.1 % de la referencia, permite que con la condición de detección de eventos absoluta exista una disminución considerable en Menv, y un aumento en Dtb. Los índices IAE, IAEP, IAD son más bajos con la condición absoluta. Al cambio de la señal de referencia a 17rad/s la condición relativa comienza a oscilar de tal manera que ts es indefinido y Mp=19.1 %, mientras que con la condición absoluta solo existe un sobre impulso (Mp) del 18.8 %, ts es igual al valor de diseño.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

- Cuando el elim es 1% del valor de la referencia, y h_{max} es 1s

En la figura 4.12: a) La condición de detección de eventos relativa presenta oscilaciones desde el momento cuando la referencia es 6rad/s, ya cuando al referencia es 17rad/s las oscilaciones incrementan. Mientras que con la condición absoluta aparece un sobre impulso (M_p) del 19.4% cuando la referencia cambia a 17rad/s, el valor de t_s es el valor de diseño. b) La señal de control con la condición relativa se satura a 12V en cada oscilación, con la condición absoluta la señal de control solo se satura cuando la señal de referencia cambia de 6 a 17 rad/s. La cantidad de mensajes enviados (M_{env}) con la condición absoluta c) es 12 mensajes menor que la condición relativa d).

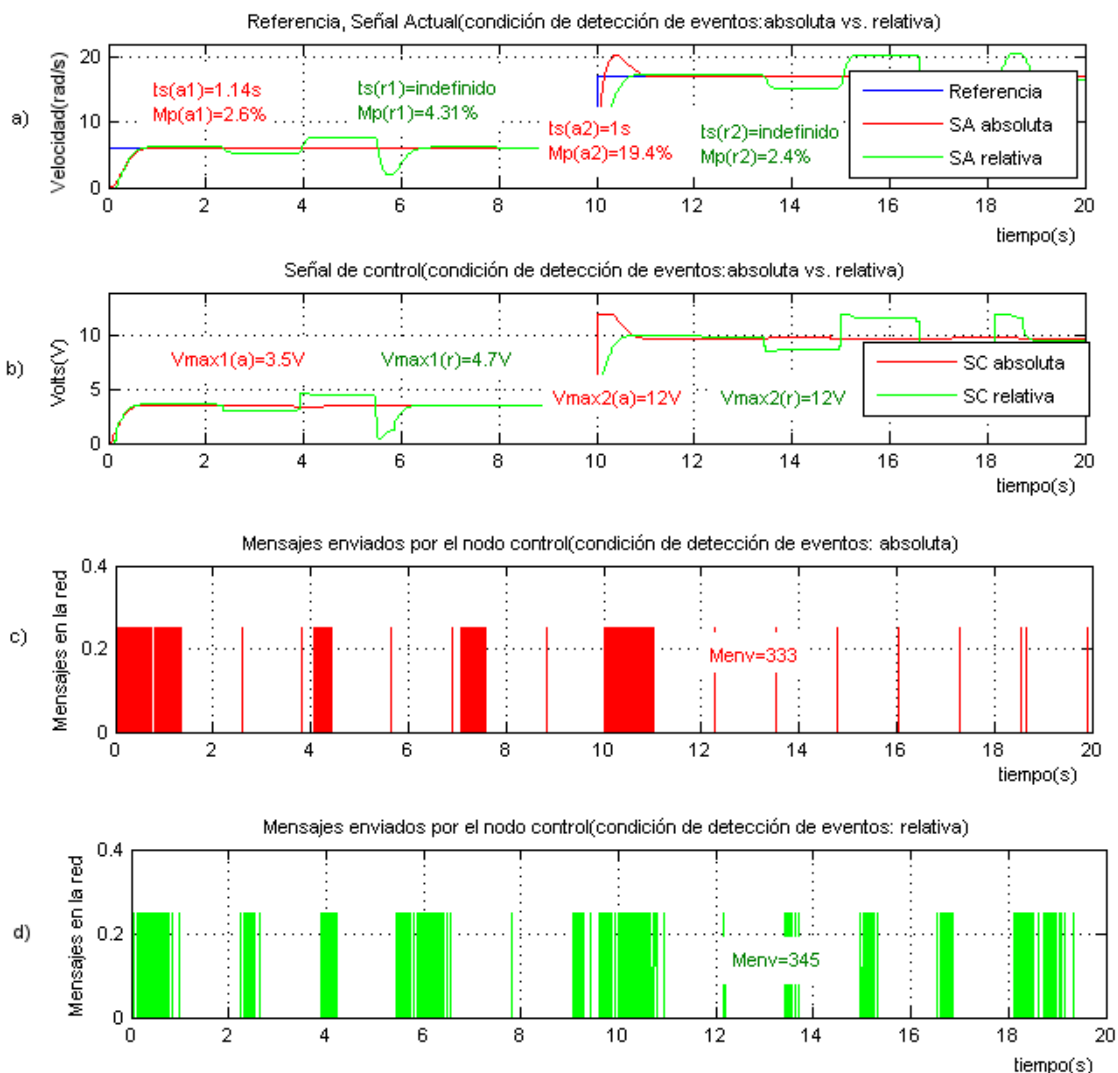


Figura 4.12: Bloque de gráficas (a) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1% del valor de la referencia, y $h_{\text{max}}=1\text{s}$.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

En la figura 4.13: a) La duración de la batería Dtb es casi la misma en ambas condiciones de detección de eventos, solo Dtb de la condición absoluta es 1 día más que la condición relativa que es 31 días. Los índices b)IAE, c)IAEP, d) IAD son mucho menores con la condición absoluta que los de la condición relativa.

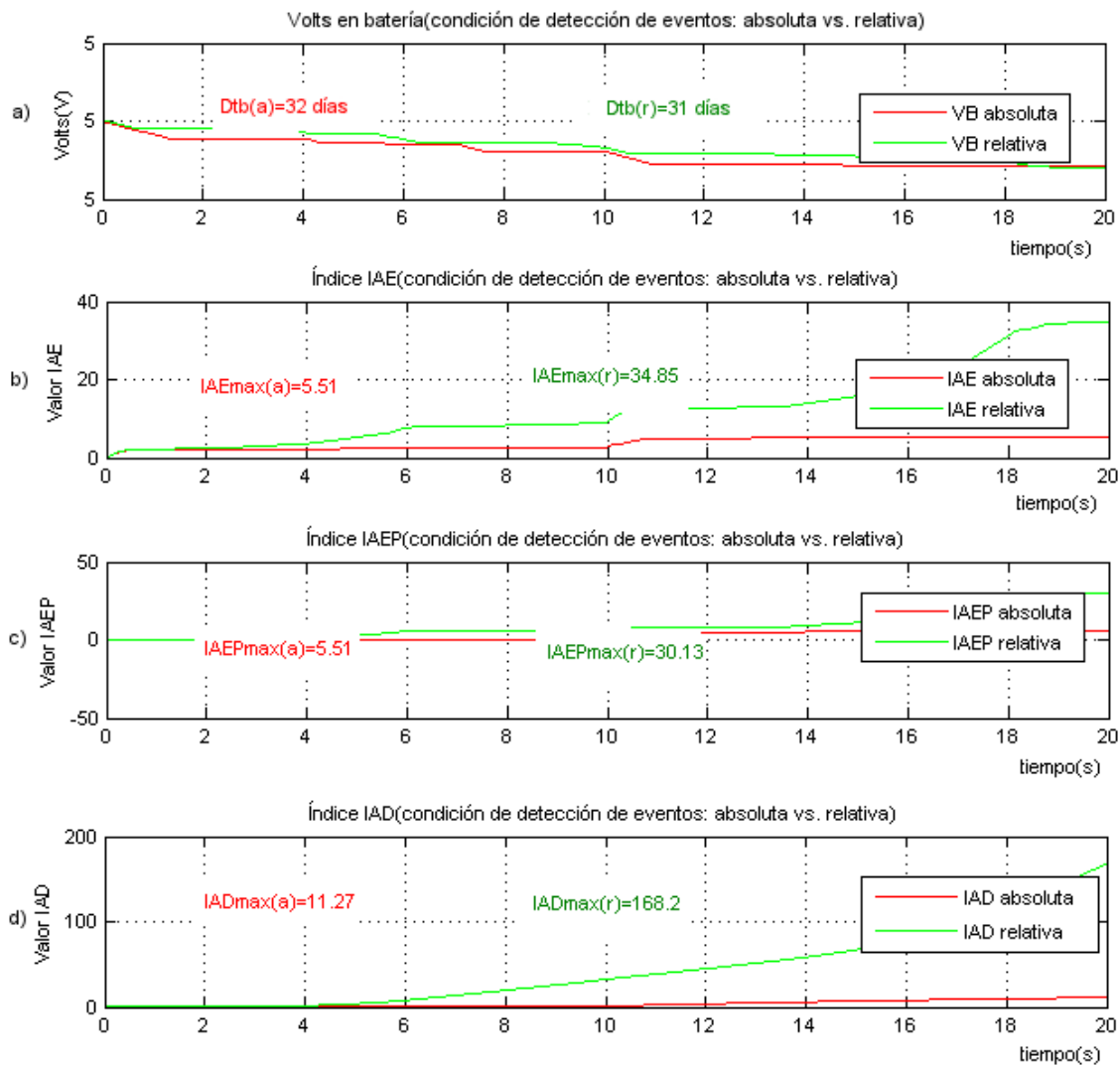


Figura 4.13: Bloque de gráficas (b) de las señales de respuesta de la estrategia de control PI basado en eventos propuesta por Árzén, cuando el elim es 1% del valor de la referencia, y hmax=1s.

Nota: Cuando el valor de elim aumenta a 1% de la referencia y hmax es 1s existe una mayor reducción de M_{env} que con los valores anteriores de elim y hmax. La condición de detección de eventos absoluta da mejores resultados de índices (IAE, IAEP, IAD), ts llega a 1s tal y como se diseñó, solo presenta un sobre impulso M_p del 19.4% al cambio de la referencia. La condición relativa también reduce M_{env} y aumenta Dtb, solo que presenta grandes sobre impulsos después de un periodo largo de muestreo y en el cambio de la señal de referencia; sus índices (IAE, IAEP, IAD) aumentan desmedidamente.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

■ **Tablas comparativas de la variación de elim y hmax en la estrategia de Árzén**

Para comprender mejor la evolución del control PI basado en eventos propuesto por Árzén al cambio de los valores de elim y hmax, a continuación se presenta una tabla comparativa de la condición de detección de eventos relativa y absoluta. Se comparan la variables vistas gráficamente solo cuando la referencia es 17 rad/s, ya que es donde se presenta el caso más caótico.

	ts		Mp		SC (Vmax)		Menv	
	a	r	a	r	a	r	a	r
elim=0.001*r hmax=0.04s	1s	1s	1.8%	2%	9.9V	9.9V	656	552
elim=0.01*r hmax=0.04s	1s	1s	1.6%	2.2%	10V	10V	590	475
elim=0.001*r hmax=1s	1s	Indf	18.8%	19.1%	12V	12V	488	550
elim=0.01*r hmax=1s	1s	Indf	19.4%	2.4%	12V	12V	333	345

	Dtb(días)		IAE		IAEP		IAD	
	a	r	a	r	a	r	a	r
elim=0.001*r hmax=0.04s	16.3	19.4	6.13	6.05	0.12	0.25	1.16	2
elim=0.01*r hmax=0.04s	18.1	22.5	6.13	6.24	0.13	0.36	1.15	0.4
elim=0.001*r hmax=1s	21.8	19.4	4.73	16.51	4.73	16.37	15.82	69.05
elim=0.01*r hmax=1s	32	31	5.51	34.85	5.51	30.13	11.27	168.2

Observaciones:

- Mientras hmax permanece en 0.04s no existe gran diferencia entre los resultados de la condición de detección de eventos relativa y la condición absoluta. Por el resultado de Menv, Dtb se puede decir que la condición relativa es mejor.
- Cuando hmax=1s el valor de Menv y Dtb mejoran con ambas condiciones de detección de eventos, pero el valor de ts, IAE, IAEP, IAD es *mucho mejor con la condición absoluta* ya que con la condición relativa se observan importantes sobre impulsos en la señal actual.
- Con la estrategia de Árzén con condición de detección de eventos absoluta con elim=0.01*r y hmax=1s se consiguen buenos resultados, pero presenta un sobre impulso máximo (Mp) del 19.4% que no es para nada conveniente. Las cuatro nuevas estrategias de control PID basado en eventos propuestas por Sylvain Durand [20] serán evaluadas con estos mismos valores de elim y hmax para probar si realmente son más eficientes que la estrategia de Árzén.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1.3. Estrategia sin condición de límite de seguridad

En la figura 4.14:a) Cuando la referencia es 6rad/s la respuesta del sistema es muy buena, pero cuando la referencia cambia a 17rad/s el valor de Mp crece a 21.4% mientras que ts es 1s igual al valor de diseño. b) La señal de control se satura a 12V cuando existe el sobre impulso máximo. c) Los índices: IAE, IAEP, IAD, no sobrepasan el valor de 8.3. d) El valor de Dtb es 45 días. e) Se envían 237 mensajes durante el tiempo de simulación, los mensajes solo se envían cuando existe un evento(cambio en la señal de referencia).

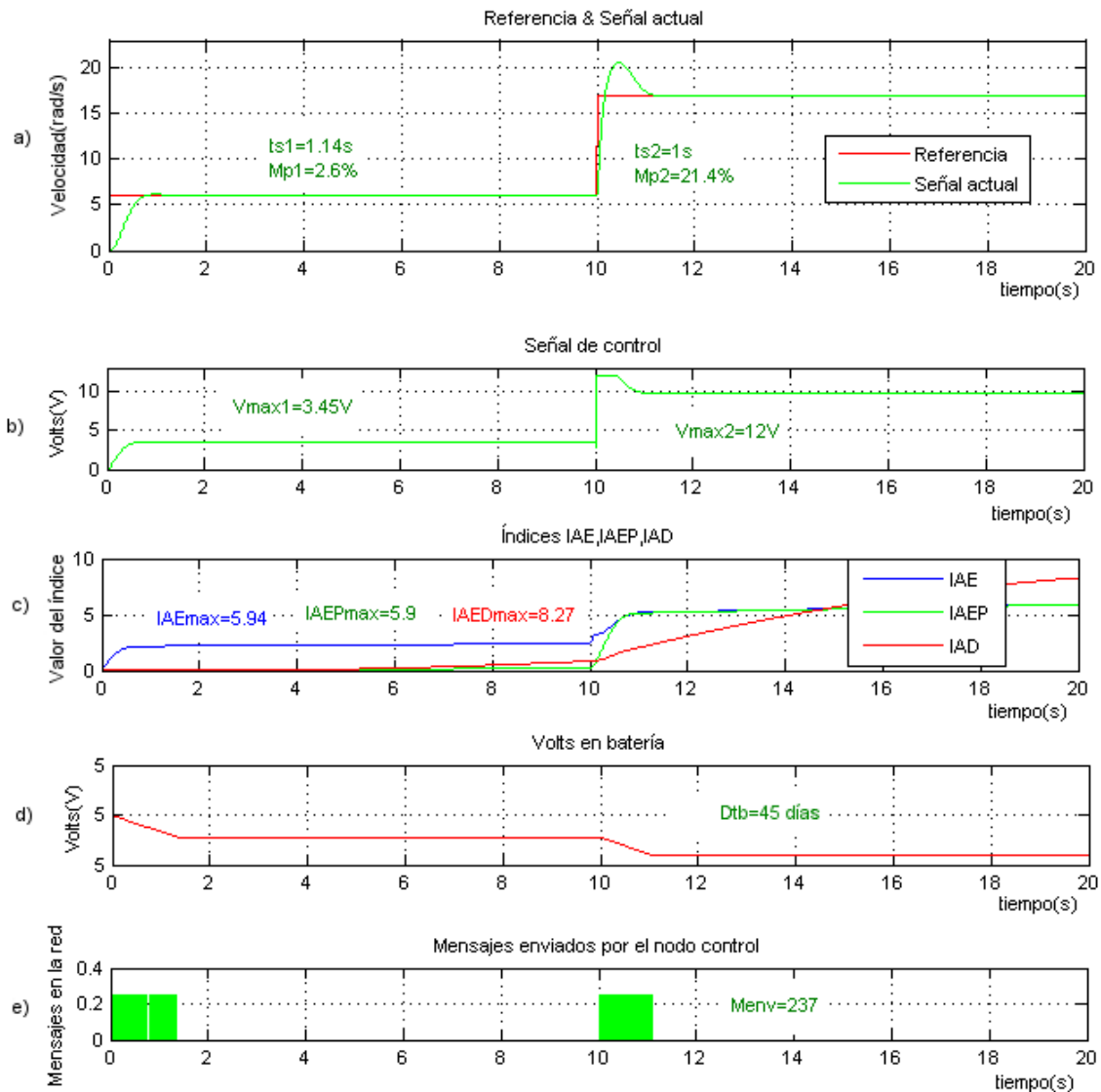


Figura 4.14: Señales de respuesta de la estrategia de control PI basada en eventos, sin condición de límite de seguridad.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1.4. Estrategia de saturación de la ganancia integral

En la figura 4.15: a) Cuando la referencia es 6rad/s el valor de MP y t_s es aproximadamente el valor de diseño. Cuando la referencia es 17rad/s el valor de t_s es el valor de diseño y el valor de Mp es igual al criterio del 2%. b) El valor máximo de la señal de control es 9.79V, no es saturado. c) El valor de los índices IAE, IAEP, IAD está por debajo de 9.3. d) El valor de Dtb es 46.6 días. e) Los mensajes enviados solo son 229, son enviados únicamente cuando el valor absoluto del error actual es mayor al valor del error límite(elim).

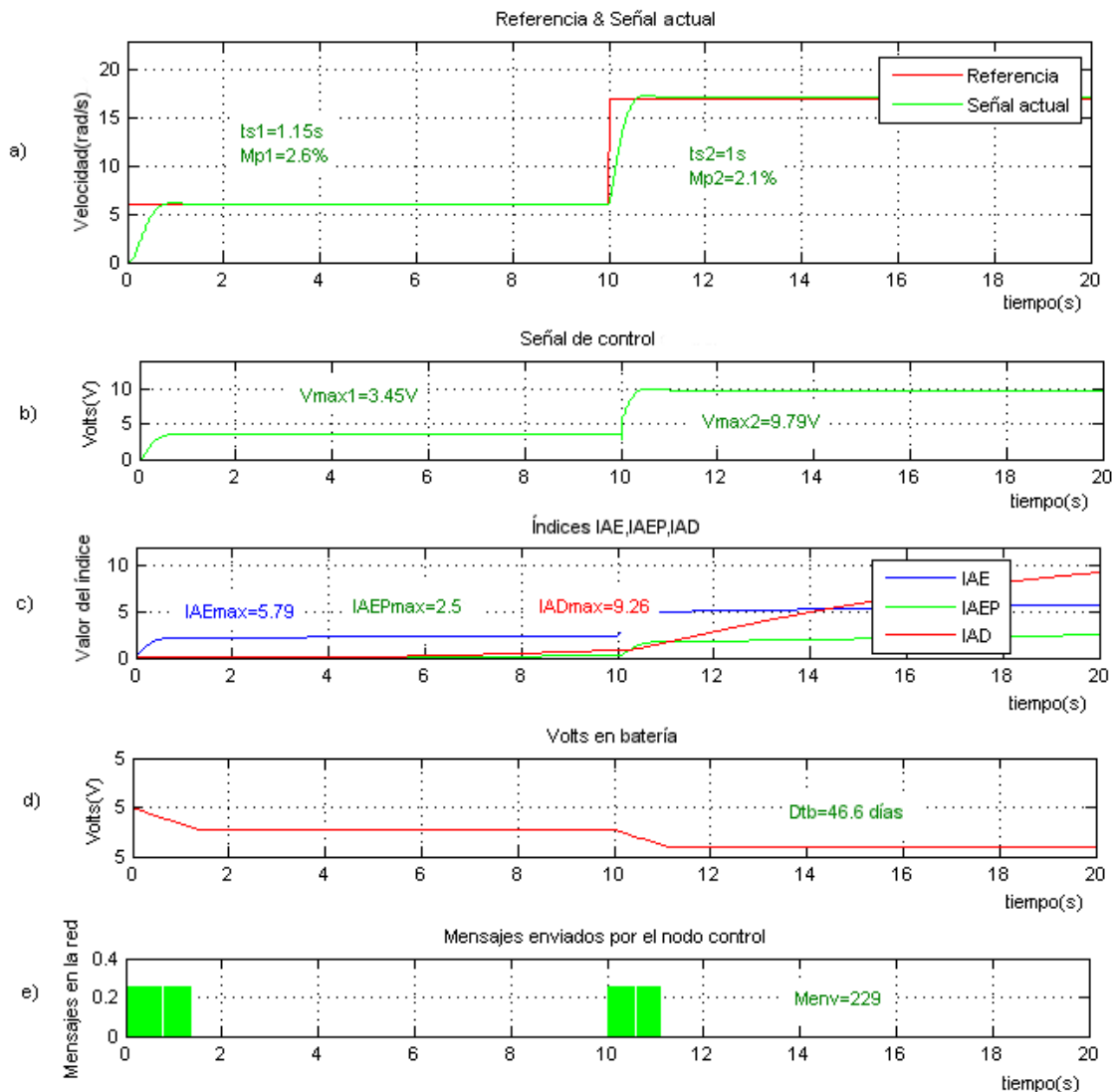


Figura 4.15: Señales de respuesta de la estrategia de control PI basada en eventos, saturación de la ganancia integral.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1.5. Estrategia del factor exponencial olvidado del intervalo de muestreo

En la figura 4.16: a) Cuando la referencia es 6rad/s los valores de t_s y M_p pasan por muy poco su valor de diseño. Cuando la referencia es 17rad/s el valor de t_s es 1s igual a el valor de diseño y M_p se encuentra 0.5% por debajo del criterio del 2%. b) La señal de control no es saturada ya que su valor está dentro del límite. c) El valor de IAE, IAEP, IAD no pasa el valor de 7.4. d) El valor de D_{tb} es 45.2 días. e) El valor de M_{env} es 236, se envían mensajes solo cuando se actualiza la señal de control.

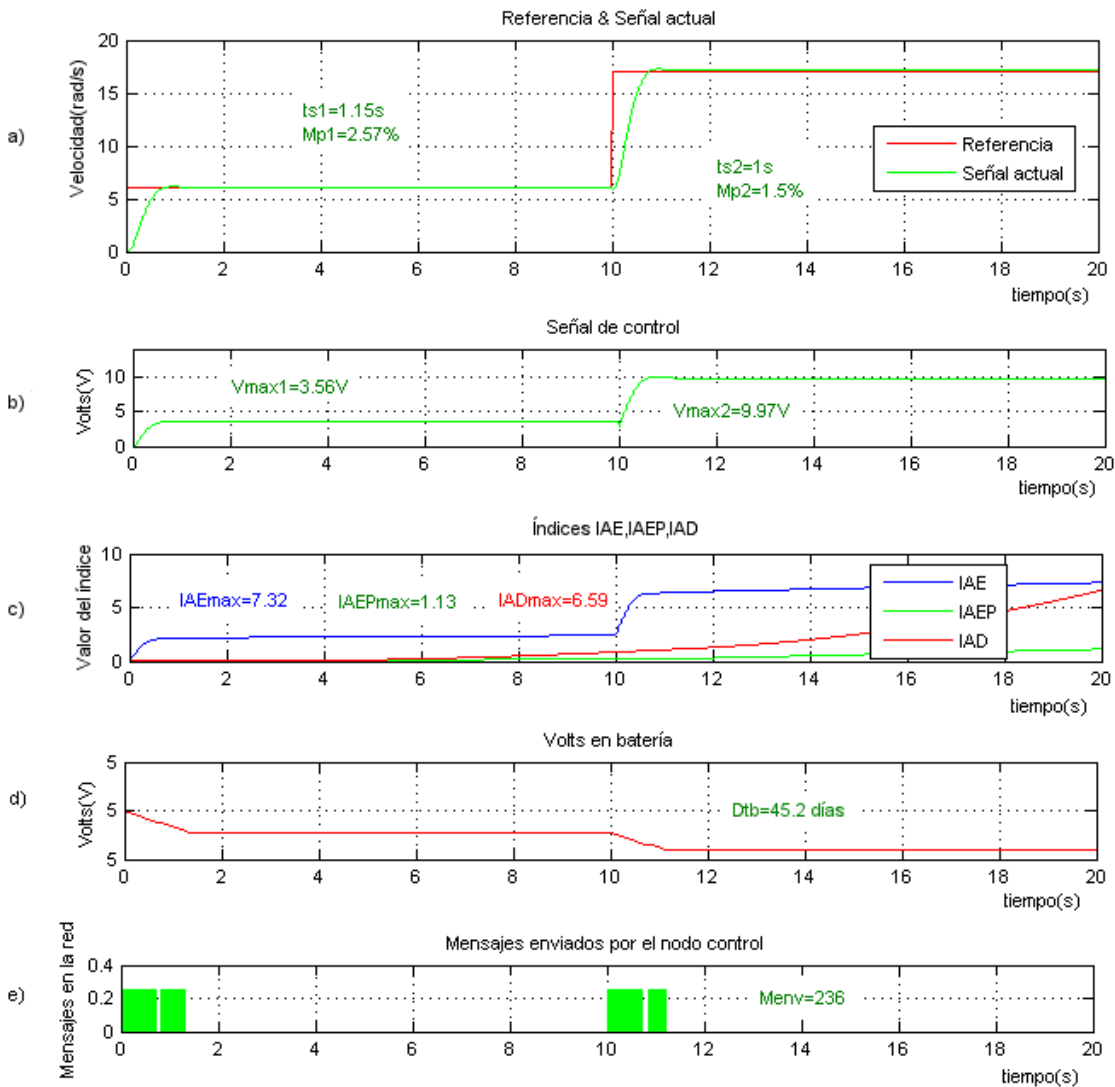


Figura 4.16: Señales de respuesta de la estrategia de control PI basada en eventos, factor exponencial olvidado de la ganancia integral.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1.6. Estrategia Híbrida

En la figura 4.17: a) Cuando la referencias es 6rad/s el valor de t_s y M_p pasa por muy poco el valor de diseño. Cuando la referencia es 17rad/s el valor de t_s es igual al valor de diseño, M_p se encuentra por debajo del criterio del 2% del error. b) La señal de control no es saturada ya que se encuentra por debajo de 12V. c) El valor de IAE, IAEP, IAD no pasa el valor de 7.4. d) El valor de D_{tb} es 45.4 días. e) Se envía un total de 235 mensajes solo cuando ocurre un evento.

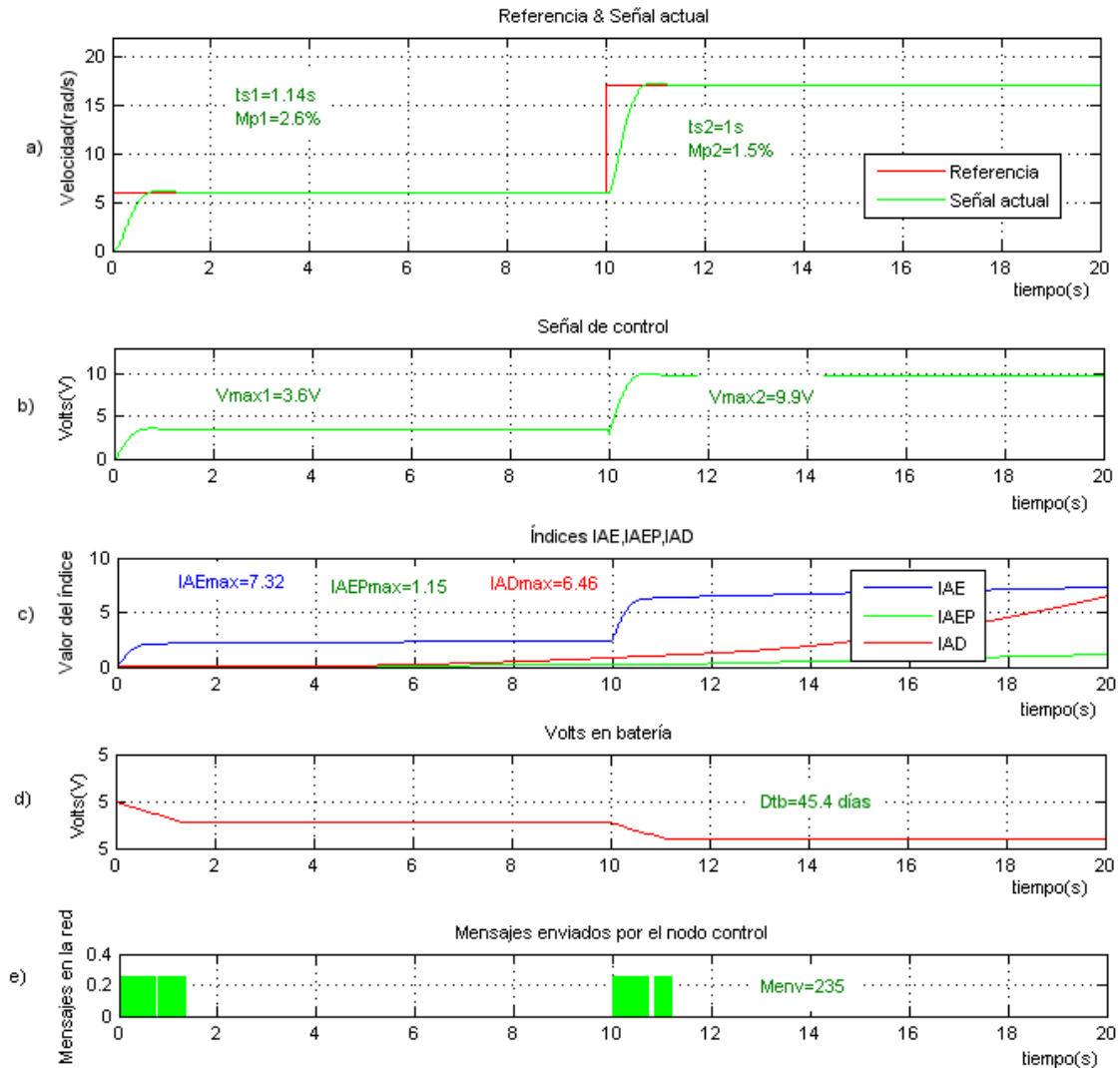


Figura 4.17: Señales de respuesta de la estrategia de control PI basada en eventos, estrategia híbrida.

**CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS
ESTRATEGIAS DE CONTROL PI**
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.1.7. Comparación de la estrategia de control basada en tiempo vs. estrategias basadas en eventos

En la siguiente tabla se muestran los resultados de la estrategia de control PI basado en tiempo y las estrategias de control PI basado en eventos, las abreviaturas utilizadas se definen en el Apéndice C. El valor de t_s , M_p , IAE, IAEP, IAD solo se toman cuando la referencia es 17rad/s, ya que es donde existe el mayor valor de M_p . La estrategia basada en tiempo tiene los mejores valores de t_s y M_p (son los valores de diseño), a excepción de M_{env} que es muy grande y el valor de D_{tb} es muy pequeño. El valor IAE es 6.25, es comparado con el valor obtenido en las otras estrategias.

Estrategia de control:	t_s	M_p	M_{env}	$D_{tb}(\text{días})$	IAE	IAEP	IAD
Basado en tiempo	1s	1.76 %	2000	5.3	6.25	-	-
Basado en eventos de Árzén	1s	19.4 %	333	32	5.51	5.51	11.27
Basado en eventos sin condición de límite de seguridad	1s	21.4 %	237	45	5.94	5.9	8.27
Basado en eventos de saturación de la ganancia integral	1s	2.1 %	229	46.6	5.79	2.5	9.26
Basado en eventos del factor exponencial olvidado del intervalo de muestreo	1s	1.5 %	236	45.2	7.32	1.13	6.59
Basado en eventos híbrida	1s	1.5 %	235	45.4	7.32	1.15	6.46

En todas las estrategias basadas en eventos existe una notable disminución en M_{env} y un aumento en D_{tb} , el valor de t_s es 1s tal como se diseñó. Existe un gran aumento en el valor de M_p en las dos primeras estrategias basadas en eventos, en la estrategia de Árzén M_p es 19.4 % y en la estrategia sin condición de límite de seguridad M_p es 21.4 %. Estos valores de M_p son inconvenientes en gran manera para el control del sistema de primer orden.

En la estrategia de saturación de la ganancia integral y la estrategia del factor exponencial olvidado del intervalo de muestreo, el valor de M_p disminuye al igual que el valor de M_{env} . Pero la estrategia híbrida muestra mejores resultados, t_s es 1s igual al valor de diseño, M_p es 1.5 % que es menor al criterio del 2 %, M_{env} disminuye un 88.3 % en comparación con la estrategia basada en tiempo, D_{tb} aumenta aproximadamente 8.6 veces su valor con respecto a la estrategia basada en eventos. Los valores de IAE, IAEP, IAD también son mejores en la estrategia basada en eventos híbrida. La estrategia híbrida presenta un perfecto equilibrio en toda las variables presentadas en la tabla.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

A continuación en las figuras 4.18, 4.19 se comparan gráficamente los resultados de las tres mejores estrategias que son: Estrategia basada en tiempo (tiene una teoría bien establecida [7]), estrategia basada en eventos de Arzén (es de las primeras estrategias de control PID basado en eventos [7]) y la estrategia híbrida propuesta por Sylvain Durand (promete mejores resultados [20]).

En la figura 4.18 la señal actual a) de la estrategia de control PI híbrida sigue casi con perfección a la señal actual de la estrategia basada en tiempo, el índice b) IAE se acerca más al valor IAE de la estrategia basada en tiempo, tiene el valor más bajo en los índices c) IAEP y d) IAD. Es demasiado notable el valor de Mp del 19.4% en la estrategia basada en eventos de Arzén, éste valor de Mp podría causar grandes desperfectos en un sistema real.

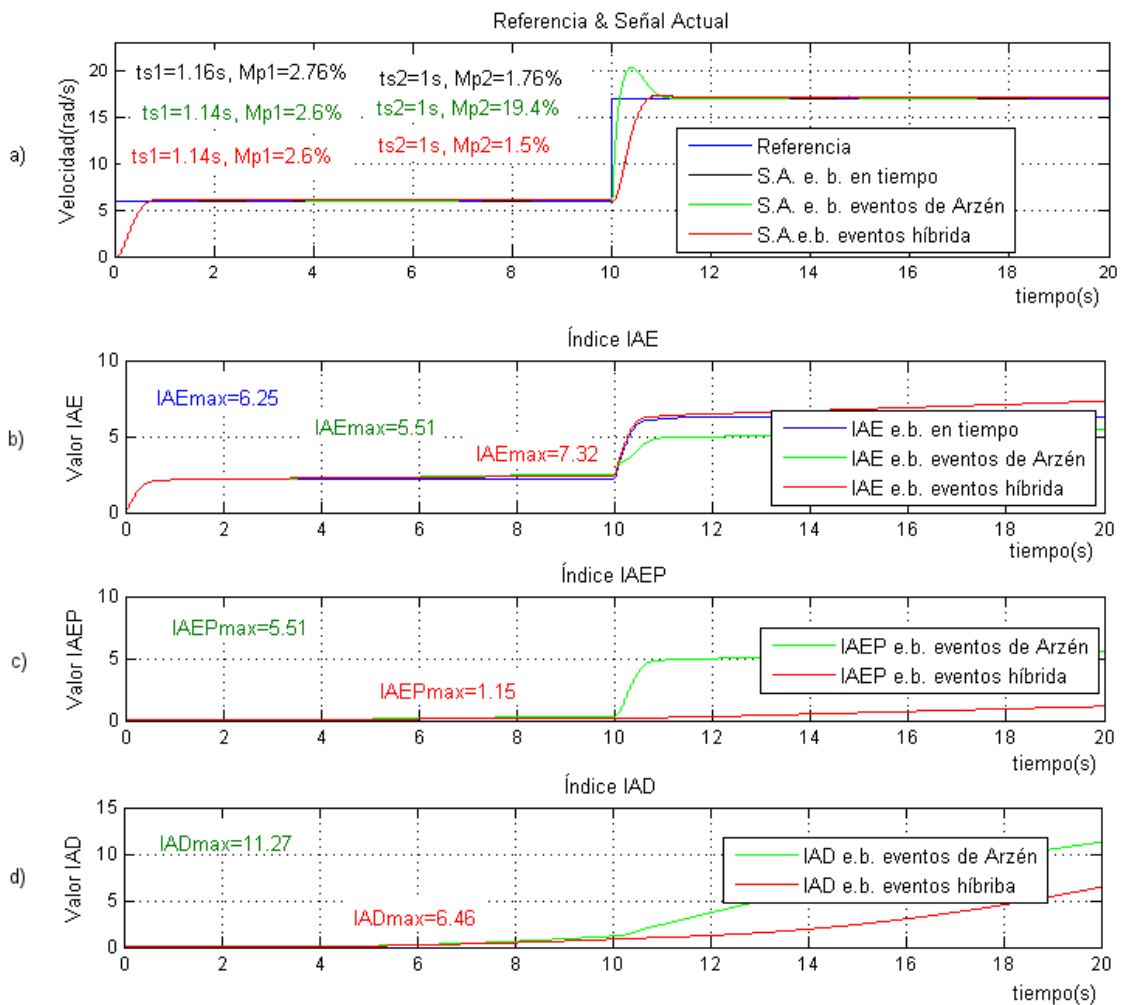


Figura 4.18: Comparación de t_s , M_p , IAE, IAEP, IAD, en las tres mejores estrategias.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

En la figura 4.19 el valor de a) Menv en la estrategia basada en eventos híbrida es 88.3% menor que en la estrategia basada en tiempo. La diferencia del valor de b) Dtb en las tres estrategias es muy notable, pero el valor de Dtb en la estrategia basada en eventos híbrida supera por aproximadamente 40 días a el valor Dtb de la estrategia basada en tiempo.

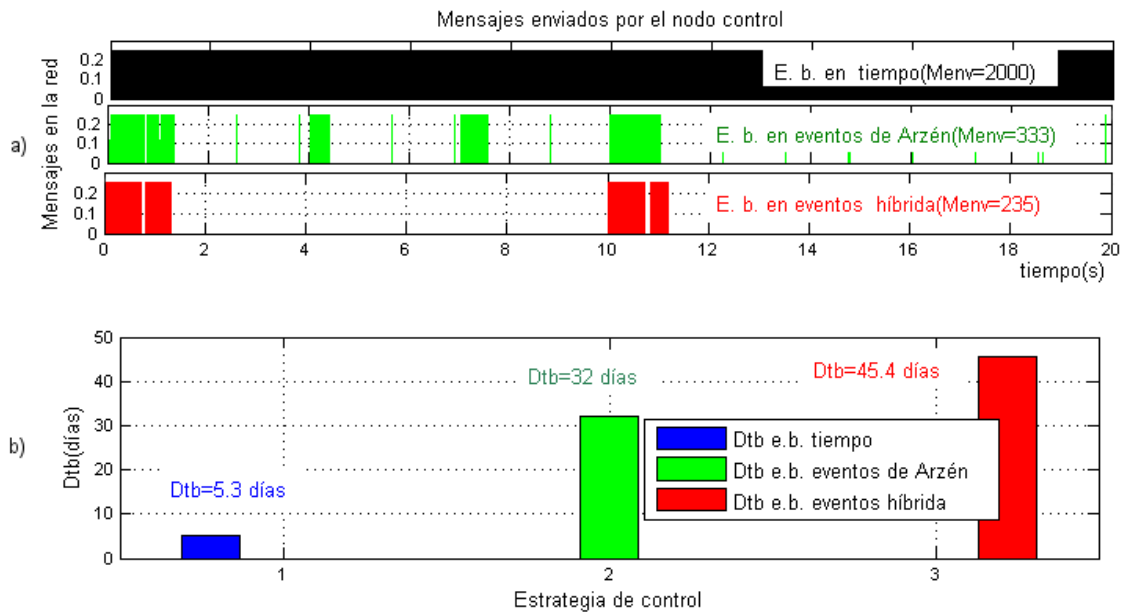


Figura 4.19: Comparación de Menv y Dtb en las tres mejores estrategias.

Estos resultados fueron obtenidos cuando el ancho de banda del nodo de interferencias era 0% y sin perturbaciones. La estrategia híbrida mostró resultados prometedores es decir, se redujo la cantidad de Menv, se aumento Dtb sin alterar las características de desempeño (t_s , M_p) y teniendo los mejores valores en los índices IAE, IAEP, IAD.

Ahora en la siguiente sub sección las tres mejores estrategias son puestas a prueba ante perturbaciones en la red y en la referencia, para comprobar si los resultados que favorecen a la estrategia basada en eventos híbrida se siguen manteniendo.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

4.2.2. Análisis de robustez respecto a interferencia en la red y perturbaciones externas

Las estrategias de control PI(basada en tiempo, basada en eventos de Árzén y la basada en eventos híbrida propuesta por Sylvain Durand) que dieron mejores resultados en el caso nominal, ahora es puesta a prueba la robustez de cada una de ellas ante perturbaciones externas al sistema(en 4s y 14s) e interferencia en la red, con un 81 % del ancho de banda del nodo de interferencias.

En las figuras 4.20 y 4.21 se muestran los resultados. La estrategia basada en tiempo, cuando la referencia es 6rad/s da buenos resultados tanto en características de desempeño, como en el valor del índice IAE, pero cuando la referencia cambia a 17rad/s se vuelven indefinidas las características de desempeño. La estrategia basada en eventos de Árzén parece tener los mejores resultados solo que cuando la referencia cambia a 17rad/s se presenta un sobre impulso máximo(Mp) del 19.7%, mientras que la estrategia híbrida solo da un 5.7% del Mp en este valor de referencia. Menv es 659 en la estrategia de Árzén y Menv es 593 en la estrategia híbrida. En las estrategias basadas en eventos de Árzén y híbrida la cantidad de mensajes enviados(Menv) por el nodo control aumenta en comparación con el valor de Menv en el caso nominal, el aumento de Menv se debe a que la señal de control tiene que actualizarse cuando aparecen las perturbaciones externas y por el retraso causado por el nodo de interferencias en el envío de datos, c) de la figura 4.20.

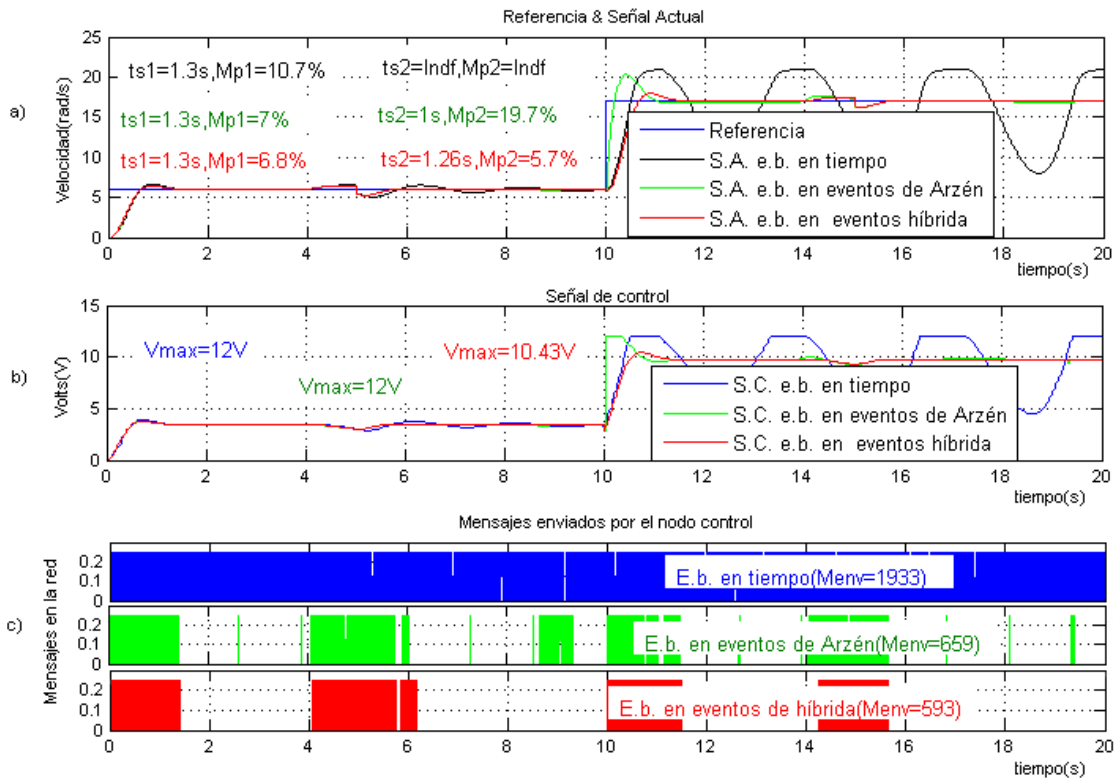


Figura 4.20: Respuesta del sistema con las tres mejores estrategias de control PI(basada en tiempo, B. eventos de Árzén, B. eventos híbrida) con el 81 % del ancho de banda del nodo de interferencias, con perturbaciones externas en 4s y 14s.

CAPÍTULO 4. IMPLEMENTACIÓN Y ESTUDIO DEL DESEMPEÑO DE LAS ESTRATEGIAS DE CONTROL PI
4.2. ESTUDIO DE LAS ESTRATEGIAS DE CONTROL PI

En la figura , a) el valor de IAE en la estrategia basada en tiempo pasa aproximadamente por 40 el valor IAE de las estrategias de control basado en eventos(las perturbaciones le afectaron mucho). b) El valor de IAEP en la estrategia basada en eventos de Årzén es 2.5 más grande que IAEP de la estrategia híbrida. c) El valor IAD de la estrategia de Årzén pasa por 21 el valor IAD de la estrategia híbrida, este valor ya es muy exagerado.

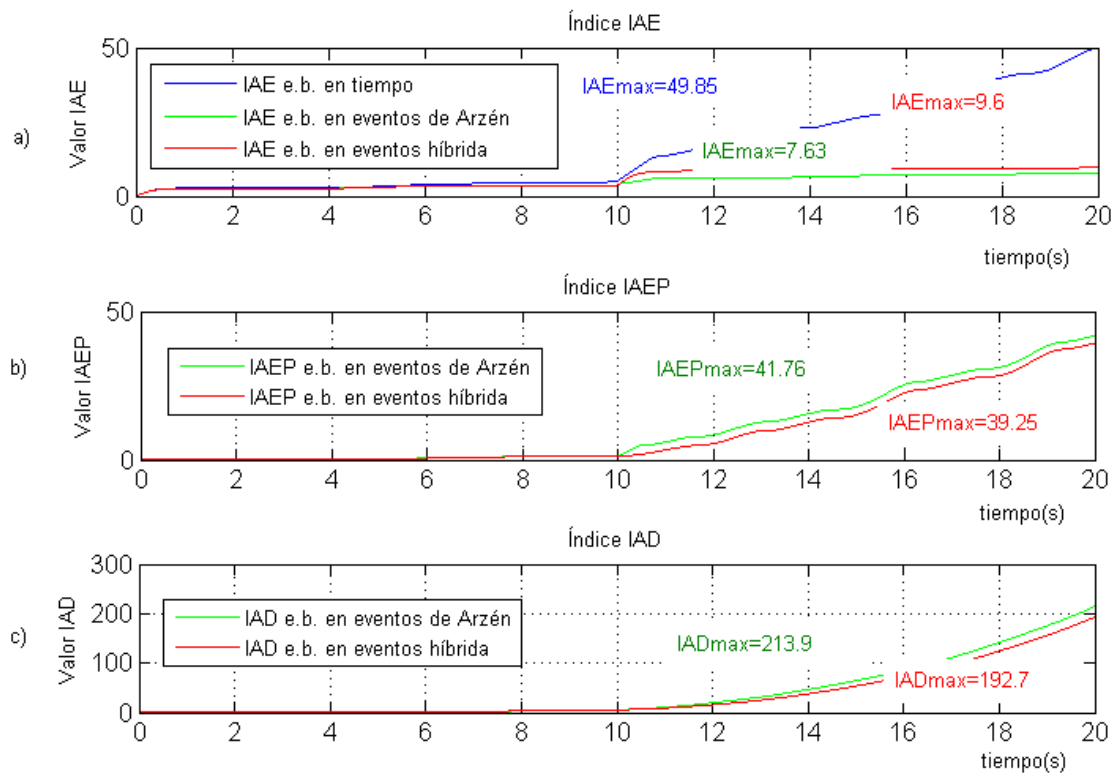


Figura 4.21: Índices IAE, IAEP, IAD de las tres mejores estrategias de control PI(basada en tiempo, B. eventos de Årzén, B. eventos híbrida), con el 81% del ancho de banda del nodo de interferencias, con perturbaciones externas en 4s y 14s.

Las tres mejores estrategias de control PI se ven afectadas por el 81% del ancho de banda del nodo de interferencias y las perturbaciones externas; la estrategia basada en tiempo es la más afectada, presenta importantes oscilaciones cuando la referencia es 17rad/s, la estrategia basada en eventos de Årzén presenta un sobre impulso máximo(Mp) del 19.7% en el cambio de la señal de referencia. La estrategia híbrida presenta un Mp del 5.7% en el cambio de referencia lo cual es preferible en comparación con el Mp de la estrategia de Årzén.

Capítulo 5

Conclusiones

- Se concluye que la estrategia de control PI basado en tiempo en un sistema retroalimentado por red, da muy buenos resultados con respecto a t_s , M_p , IAE, siempre y cuando no existan perturbaciones en la red, ya que cuando existen perturbaciones en la red comienzan a aparecer oscilaciones importantes que no son convenientes. Su respuesta es muy buena ante perturbaciones externas. En comparación con las estrategias de control PI basado en eventos, la cantidad de veces que se actualiza la señal de control es mucho mayor, así mismo incrementa la cantidad de mensajes enviados y aumenta el consumo de energía.
- En las estrategias de control PI basado en eventos, la condición de detección de eventos absoluta da mejores resultados (t_s , M_p , IAE, IAEP, IAD) que la condición de detección de eventos relativa. En la condición relativa se presentan importantes oscilaciones cuando se incrementa el valor de h_{max} y q_{nom} .
- En la estrategia de control PI basado en eventos de Årzén (2008) en un sistema retroalimentado por red, cuando existe una perturbación o cambio en la señal de referencia después de un periodo largo aparece un gran incremento en el valor de M_p .
- La estrategia híbrida de control PI basado en eventos (S. Durand 2011) en un sistema retroalimentado por red, da mejores resultados que la estrategia basada en tiempo y las otras estrategias basadas en eventos, ya que demuestra ser más robusta (t_s y M_p se mantienen cerca del valor de diseño; IAE, IAEP y IAD son bajos) ante perturbaciones en la red y perturbaciones externas. La cantidad de veces que se actualiza la señal de control es menor que en las demás estrategias, reduce hasta un 88 % la cantidad de mensajes enviados en la red y disminuye el consumo de energía.
- En sistemas basados en red que requieren el uso de más de una red de comunicación (automóvil), la estrategia híbrida de control PI promete disminución en costos y energía, ya que el ancho de banda de una red es optimizado.
- En un sistema de control de retroalimentación simple (que no está retroalimentado por red), la estrategia de control PI basado en eventos híbrida también promete buenos resultados desde el punto de vista de t_s , M_p , D_{tb} , IAE, IAEP, IAD. A pesar de que en la actualización de la señal de control no está implicada la transmisión de datos en una red, se implica el uso del micro controlador; es decir entre más se actualice la señal de control el micro controlador procesara más tareas [10] que finalmente repercuten en el consumo de energía. En realidad es muy pequeño el consumo de energía por tarea realizada (en [37] se propone medir de forma experimental el consumo de energía por tarea realizada en un microcontrolador), pero en un largo periodo repercutirá en un ahorro o desgaste significativo de energía.

Bibliografía

- [1] K. J. Åström, R. M. Murray, "Feedback Systems", Copyright ©2008 by Princeton University Press.
- [2] K.B.M. Krishna, B.V.S. Goud, "Speed Control of Dc Motor through Position and Speed Tracking System Using Neural Networks", International Journal of Electronic Engineering Research, ISSN 0975-6450, Vol. 5, No. 1, pp. 87-103, 2013.
- [3] Y. Tipsuwan and M.Y. Chow, "Control methodologies in networked control systems", Control Eng. Practice, Vol. 11, pages 1099-1111, 2003.
- [4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems", Proc. IEEE, vol. 95, no. 1, pp. 138-162, Jan. 2007.
- [5] T. C. Yang, "Networked control system: A brief survey", IEEE Proc. Control Theory Appl, vol. 153, no. 4, pages 403-412, Jul. 2006.
- [6] K.E. Årzén, "A simple event-based PID controller", In Preprints of the 14th World Congress of IFAC, Beijing, P.R. China, 1999.
- [7] S. Dormido, J. Sánchez, E. Kofman, "Muestreo, control y comunicación basado en eventos", Revista iberoamericana de automática e informática industrial (RIAI), ISSN 1697-7912, Vol. 5, No. 1, págs. 5-26, Enero 2008.
- [8] R. Kelly, J. Moreno, "Learning PID structures in an introductory course of automatic control", IEEE Transactions on Education, vol 44, No.4, November 2001.
- [9] A. Roca, "Control de procesos con prácticas de simulación y análisis por ordenador PC", 3a edición, AlfaOmega, 03/03/2010.
- [10] D. Ibrahim, "Advanced PIC microcontroller projects in c: from USB to RTOS with the PIC18F series", Newnes, 2008.
- [11] S. Durand and Marchand, "Further Results on Event-Based PID Controller", Proceedings of the European Control Conference 2009, Budapest Hungary, August 23-26, 2009.
- [12] S. Durand, J.F. Guerrero-Castellanos and N. Marchand, "Event-Based PID Controllers without Safety Limit Condition Experimental Results", Reporte interno GIPSA-LAB y FCE-BUAP, Julio 2011.
- [13] M. Ohlin, D. Henriksson and A. Cervin, "TRUETIME1.5-Reference Manual", Department of Automatic Control, Lund University, January 2007.
- [14] M. B. Hoagland and B. Dodson, "The Way Life Works", Times Books, New York, 1995.
- [15] K. J. Åström, "Control System Design", Copyright ©2002 by Princeton University Press.

- [16] Stanford Research Systems, "Small Instrumentation Modules", www.thinkSRS.com.
- [17] Enfield Technologies, "Pneumatic Systems Controller Datasheet", DS1001; rev 20100315, Copyright © 2004-2010 Enfield Technologies, LLC.
- [18] BRAINCHILD Electronic, "BTC-8300. Self-Tune Fuzzy / PID Process / Temperature Controller", <http://www.bcmfg.com>.
- [19] K. Åström and B. Wittenmark, "Computer Controlled Systems", 3rd Edition. Prentice Hall, 1997.
- [20] S. Durand, "Reduction of the energy consumption in embedded electronic devices with low control computational cost", Tesis doctoral, version 1, Universidad de GRENOBLE, 18 de Abril 2011.
- [21] H. Van Gageldonk, K. van Berkel, A. Peeters, D. Baumann, D. Gloor, and G. Stegmann. "An asynchronous low-power 80C51 microcontroller", In Proceedings of the 4th International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 96-107, 1998.
- [22] J. Sánchez, M. Guarnes, and S. Dormido, "On the application of different event-based sampling strategies to the control of a simple industrial process", *Sensors*, 9:6795-6818, 2009.
- [23] J. Sánchez, M. Guarnes, S. Dormido, and A. Visioli. "Comparative study of event-based control strategies: An experimental approach on a simple tank", In Proceedings of the 10th European Control Conference, 2009.
- [24] J. Sandee, W. Heemels, and P. van den Bosch, "Event-driven control as an opportunity in the multidisciplinary development of embedded controllers", In Proceedings of American Control Conference, pages 1776-1781, 2005.
- [25] K. Åström and B. Bernhardsson, "Comparison of periodic and event based sampling for first-order stochastic systems", In Preprints 14th World Congress of IFAC, page 301U306, 1999.
- [26] K. Åström and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems", In Proceedings of the 41st IEEE Conference on Decision and Control, 2002.
- [27] N. Marchand, "Stabilization of Lebesgue sampled systems with bounded controls: the chain of integrators case", In Proceedings of the 17th IFAC World Congress, 2008.
- [28] W. Heemels, J. Sandee, and P. van den Bosch, "Analysis of event-driven controllers for linear systems", *International journal of control*, 81:571-590, 2009.
- [29] J. Lunze and D. Lehmann, "A state-feedback approach to event-based control", *Automatica*, 46:211-215, 2010.
- [30] M. Donkers and W. Heemels, "Output-based event-triggered control with guaranteed l_1 gain and improved event-triggering", In Proceedings of the IEEE Conference on Decision and Control, 2010.
- [31] A. Eqtami, D. Dimarogonas, and K. Kyriakopoulos, "Event-triggered control for discrete-time systems", In Proceedings of the IEEE American Control Conference, 2010.
- [32] O. Sename, D. Simon, and D. Robert, "Feedback scheduling for real-time control of systems with communication delays", In Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, volume 2, 2003.

- [33] D. Simon, D. Robert, and O. Sename, "Robust control/scheduling co-design: application to robot control", In Proceedings of the IEEE Symposium on Real-Time and Embedded Technology and Applications, pages 118-127, 2005.
- [34] M. J. Mazo, A. Anta, and P. Tabuada, "On self-triggered control for linear systems: Guarantees and complexity", In Proceedings of the 10th European Control Conference, 2009.
- [35] K. Ogata, "Ingeniería de control moderna", Tercera edición, PRENTICE HALL, 1998.
- [36] A. Granja, F. Javier, H. Esteban y J. Francisco, "Desarrollo de una interface USB-CAN", Tesis de ingeniería, Universidad de Valladolid, Junio 2012.
- [37] G. Calandrini, A. Gardel, I. Bravo, P. Revenga, J. L. Lázaro, F. J. Toledo-Moreno, "Power Measurement Methods for Energy Efficient Applications", Revista sensors 2013, ISSN 1424-8220, Junio 2013.

Parte I
Apéndice

Apéndice A

Algunos comandos de TrueTime

Estos son los comandos mas usados, y en especial para el desarrollo de los sistemas propuestos en ésta tesis. Están basados en la bibliografía [13].

A.1. `ttAnalogIn(TH)`

Propósito

Lee un valor del canal analógico de entrada.

Sintaxis en Matlab

Valor=`ttAnalogIn` (inpChan)

Argumento

inpChan El canal de entrada en el que se leerá la información.

Descripción

Esta función es usada para leer una entrada analógica del medio ambiente. El canal de entrada debe ser entre 1 y el número de canales de entrada del bloque núcleo especificado por *ttInitKernel*.

A.2. `ttAnalogOut(TH)`

Propósito

Escribe un valor a un canal analógico de salida.

Sintaxis en Matlab

`ttAnalogOut`(outpChan,value)

Argumento

outpChan El canal de salida en el que se va a escribir.

value El valor que se va a escribir.

Descripción

Esta función es usada para escribir una salida analógica al medio ambiente. El canal de salida debe ser entre 1 y el número de canales de salida especificados en *ttInitKernel*.

A.3. `ttCallBlockSystem(TH)`

Propósito

Llama un diagrama a bloques de Simulink dentro de un código de funciones.

Sintaxis en Matlab

out= `ttCallBlockSystem`(nbroutp, inp, blockename)

Argumento

<i>nbrinp</i>	Número de entradas del diagrama a bloques.
<i>nbroupt</i>	Número de salidas del diagrama a bloques.
<i>inp</i>	Vector de valores de entrada.
<i>outp</i>	Vector de valores de salida.
<i>blockname</i>	El nombre del diagrama a bloques de Simulink.

Descripción

Esta función es usada para llamar un diagrama a bloques de Simulink dentro de un código de función. En cada llamada, se realiza una segunda simulación del bloque, usando los estados antiguos como valores iniciales. Los estados del diagrama a bloques son almacenados internamente por el núcleo entre llamadas. Consecuentemente, los diagramas a bloques podrían contener únicamente bloques discretos y los tiempos de muestreo que deberían ser fijados a uno. Las entradas y salidas son definidas por los puertos de entrada y de salida de Simulink.

A.4. ttCreateInterruptHandler(I)

Propósito

Crea un gestor de interrupciones en TrueTime.

Sintaxis en Matlab

ttCreateInterruptHandler(name, priority, codeFcn)

ttCreateInterruptHandler(name, priority, codeFcn, data)

Argumento

<i>name</i>	Nombre del gestor el cual debe ser único.
<i>priority</i>	Prioridad del gestor. Debe ser un valor mas grande que cero, el número mas pequeño representa la prioridad mas alta .
<i>codeFcn</i>	Código de función del gestor, donde codeFcn es una cadena (nombre de un archivo .m)
<i>data</i>	Una estructura de datos arbitraria representa la memoria local de un gestor.

Descripción

Esta función es usada para crear un gestor que será ejecutado en respuesta a interrupciones. Los gestores de interrupción pueden ser asociados con temporizadores, el canal de recepción de la red, canal de interrupción de la red, o vincular a las tareas como gestores de desbordamiento. Cada únicamente gestor puede ser asociado con una fuente de interrupción.

A.5. ttCreateJob(ITH)

Propósito

Crea el trabajo de una tarea.

Sintaxis en Matlab

ttCreateJob(taskname)

Argumento

taskname Nombre de la tarea.

Descripción

Esta función es usada para crear trabajo a trabajos de tareas. Si ya está un trabajo activo para la tarea, el trabajo es puesto en una cola de espera y sirve tan pronto como sea posible. ttCreateJob debe ser llamado para activar una tarea aperiódica, es decir tareas creadas usando ttCreateTask. Una llamada a ttCreateJob accionará el gancho de llegada de la tarea. Si no hay trabajos activos el gancho del descargo también se llamará. En otro caso, el gancho del descargo será llamado cuando el trabajo se activa después de la cola del trabajo.

A.6. ttCreateMailbox(I)

Propósito

Crea un buzón de TrueTime para la comunicación entre tareas.

Sintaxis en Matlab

```
ttCreateMailbox(mailboxname)  
ttCreateMailbox(mailboxname, maxsize)
```

Argumento

mailboxname Nombre del buzón. Debe ser único.
maxsize Tamaño del buffer asociado con el buzón(el predefinido es INT _ MAX).

Descripción

Esta función es usada para la comunicación entre tareas. El buzón TrueTime implementa mensajes asíncronos pasando con nombramiento indirecto. Un buffer es utilizado para almacenar mensajes entrantes, y el tamaño máximo del buffer puede ser especificado por maxsize.

A.7. ttCreatePeriodicTask(I)

Propósito

Crea una tarea periódica de TrueTime.

Sintaxis en Matlab

```
ttCreatePeriodicTask(name, offset, period, priority, codeFcn)  
ttCreatePeriodicTask(name, offset, period, priority, codeFcn, data)
```

Argumento

name Nombre de la tarea. Debe ser único.
offset Libera el tiempo para el primer trabajo de la tarea periódica.
period Periodo de la tarea.
priority Prioridad de la tarea. Debe ser un valor más grande que cero, donde un número pequeño representa una prioridad alta. Relevante únicamente para prioridad fija de programación.
codeFcn Código de función de la tarea, donde codeFcn es una cadena (nombre de un archivo.m).
data Una estructura de dato arbitrario representa la memoria local de la tarea.

Descripción

Esta función se usa para crear tareas periódicas para ser ejecutadas en el núcleo de TRUE TIME. La periodicidad es implementada internamente por el núcleo usando un temporizador periódico.

A.8. ttCreatePeriodicTimer(ITH)

Propósito

Crea un temporizador periódico y lo asocia a un gestor de interrupción.

Sintaxis en Matlab

```
ttCreatePeriodicTimer (timername, start, period, handlername)
```

Argumento

timername Nombre del temporizador. Debe ser único.
start El tiempo para el primer vencimiento del temporizador.
period Periodo del temporizador.
handlername Nombre del gestor de interrupciones asociado con el temporizador.

Descripción

Esta función se usa para crear temporizadores periódicos. Cada vez que el tiempo vence el gestor de interrupciones asociado se activa y se fija para la ejecución.

A.9. ttCreateTask(I)

Propósito

Crea una tarea de TrueTime.

Sintaxis en Matlab

```
ttCreateTask(name,deadline, priority, codeFcn)
```

```
ttCreateTask(name,deadline, priority, codeFcn, data)
```

Argumento

<i>name</i>	Nombre de la tarea el cual debe ser único.
<i>deadline</i>	Plazo fatal relativo de la tarea.
<i>priority</i>	Prioridad de la tarea. Debe ser un valor mas grande que cero, un número pequeño representa una prioridad alta. Relevante únicamente para prioridad fija de programación.
<i>codeFcn</i>	Código de función de la tarea, donde <i>codeFcn</i> es una cadena (nombre de un archivo .m).
<i>data</i>	Una estructura de datos arbitraria representa la memoria local de la tarea.

Descripción

Esta función es usada para crear una tarea para ejecutarla en el núcleo de TrueTime. Observa que un trabajo de tarea no es creado por ésta función. Esto es hecho por el código *ttCreateJob*. El peor caso del tiempo de ejecución de la tarea es fijarlo igual a *deadline* de la tarea.

A.10. ttGetMsg(TH)

Propósito

Obtiene el mensaje que ha sido recibido de la red. Ésta función es usada en redes alambradas y en inalámbricas.

Sintaxis en Matlab

```
[msg, signalPower]=ttGetMsg
```

```
[msg, signalPower]=ttGetMsg(network)
```

Argumento

<i>network</i>	Interface de red del cual el mensaje debería ser recibido. El número de red predefinido es 1.
<i>signalPower</i>	Valor de la señal de poder recibida correspondiente a este mensaje. Se usa únicamente en las redes inalámbricas.

Descripción

Esta función se usa para recuperar un mensaje que ha sido recibido de la red. Típicamente, se notificara por una interrupción que existe un mensaje en la cola de entrada de la interface de red, pero esto también es posible para interrogar por un nuevo mensaje. Si no existe un mensaje, la función regresará una matriz vacía(Matlab).

A.11. ttInitKernel(I)

Propósito

Inicializa el núcleo de TrueTime.

Sintaxis en Matlab

ttInitKernel (nbrInp, nbrOutp, prioFcn)

ttInitKernel (nbrInp, nbrOutp, prioFcn, cs _ oh)

Argumento

<i>nbrInp</i>	Número de canales de entrada, es decir el tamaño de los puertos Analógico/Digitales del bloque del núcleo.
<i>nbrOutp</i>	Número de canales de salida, es decir el tamaño de los puertos Digital/Analógicos del bloque del núcleo.
<i>prioFcn</i>	Política de programación usadas por el núcleo.
<i>cs _ oh</i>	Tiempo superior para una interrupción de contexto total. A menos que sea especificado, un cero sera asociado con los interruptores de contexto.

Descripción

Esta función realiza la inicialización necesaria del bloque núcleo y primero debe ser llamada en todo el escrito de inicialización. La prioridad de la función en Matlab debería ser alguna de los siguientes casos; 'prioFP', 'prioRM', 'prioDM', o 'prioEDF'.

A.12. ttInitNetwork(I)

Propósito

Inicializa la interface de red de TRUE TIME. Si el núcleo debe ser conectado a varias redes, esta función tiene que ser llamada varias veces. Esta función es usada en redes alambradas e inalámbricas.

Sintaxis en Matlab

ttInitNetwork(nodenum, handlername)

ttInitNetwork(network, nodenum, handlername)

Argumento

<i>network</i>	Número del bloque de red de TRUE TIME. El número de red predefinido es 1
<i>nodenum</i>	Dirección del nodo en la red. Tiene que ser un número entre 1 y el número de nodos que se especificaron en el cuadro de diálogo del bloque de red correspondiente de TRUE TIME.
<i>handlername</i>	Nombre del gestor de interrupción, que debe ser llamado cuando un mensaje llega a la red .

Descripción

La interface de la red debe ser inicializada usando este comando antes de que algún mensaje pueda ser enviado o recibido. La inicialización se suspenderá si no hay bloques de red de TRUE TIME en el modelo Simulink.

A.13. ttSendMsg(TH)

Propósito

Envía un mensaje por la red.

Sintaxis en Matlab

ttSendMsg(receiver, data, length)

ttSendMsg(receiver, data, length, priority)

APÉNDICE A. ALGUNOS COMANDOS DE TRUETIME

A.14. TTSETKERNELPARAMETER(ITH)

ttSendMsg([network receiver], data, length)
ttSendMsg([network receiver], data, length, priority)

Argumento

<i>network</i>	Interface de red en que el mensaje es enviado. El número de red predefinido es 1.
<i>receiver</i>	El número del nodo receptor(un número entre 1 y el número de nodos). A este nodo se le permite enviarse mensajes a el mismo. Especificar el número receptor 0 para transmitir un mensaje a todos los nodos en la red.
<i>data</i>	Estructura de datos arbitraria representa el contenido del mensaje.
<i>length</i>	Longitud del mensaje, en bits. Determina el tiempo que le tomara transmitir el mensaje.
<i>priority</i>	Prioridad del mensaje(relevante únicamente para redes CSMA/AMP). Si no se especifica la prioridad será dada por el número de nodo de envío, es decir los mensajes enviados del nodo 1 tendrán una prioridad más alta por defecto.

Descripción

La interface de red debe ser inicializada usando *ttInitNetwork* antes de que un mensaje pueda ser enviado.

A.14. ttSetKernelParameter(ITH)

Propósito

Fija un parámetro específico del núcleo.

Sintaxis en Matlab

ttSetKernelParameter(parameter, value)

Argumento

<i>parameter</i>	Nombre del parámetro a ser cambiado.
<i>value</i>	Nuevo valor del parámetro.

Descripción

Esta función hace cambiar los parámetros del núcleo de forma rápida. Soporta los siguientes parámetros:

- *cpuscaling*
- *energyconsumption*

El valor prefijado para el parámetro *cpuscaling* es 1, *energyconsumption* es 0. Establece *cpuscaling* en 2, hará que el bloque núcleo ejecute todo dos veces más rápido. Hará esto posible para experimentar con diferentes métodos de escalar voltaje dinámico. Al mismo tiempo el consumo de energía puede ser fijado a un valor realista y ser conectado a un bloque batería.

A.15. ttSetNetworkParameter(ITH)

Propósito

Fija un parámetro de red específico en una base por nodo.

Sintaxis en Matlab

ttSetNetworkParameter(parameter, value)

ttSetNetworkParameter(networkNbr, parameter, value)

Argumento

<i>parameter</i>	Nombre del parámetro a ser cambiado.
<i>value</i>	Nuevo valor del parámetro.
<i>networkNbr</i>	La interfaz de la red en que el parámetro debe ser cambiado. El número de red prefijada es 1.

Descripción

Esta función cambia los parámetros de red en una base por nodo. Son soportados los siguientes parámetros:

<i>transmitpower</i>	El parámetro <i>transmitpower</i> es valido únicamente cuando se usan redes inalámbricas y predefinido para cualquiera que sea fijado en la máscara del bloque.
<i>predelay</i>	El tiempo que un mensaje es retardado por la interface de red en el fin del envío. Por ejemplo se puede usar para modelar una conexión serial lenta entre la computadora y la interface de red. El valor predefinido es cero.
<i>postdelay</i>	El tiempo que un mensaje es retardado por la interface de red en el fin de recepción. El valor predefinido es cero.

A.16. ttSleepUntil(T)

Propósito

Pone a dormir la tarea en ejecución hasta cierto punto en el tiempo.

Sintaxis en Matlab

ttSleepUntil(time)

Argumento

time Tiempo en que la tarea debe despertar.

Descripción

Esta función se usa para hacer dormir una tarea en ejecución hasta un punto específico en el tiempo.

Apéndice B

Parámetros de batería

Esta batería recargable es simulada y alimenta el nodo control del sistema de control. Los datos técnicos se describen a continuación:

Nombre del producto	Mobile power	Modelo del producto	MP4000-829
Material de la batería	Polymerlithiumbatteries	Capacidad	4000mAh
Entrada	5V/1A	Salida	5V/1A
Tiempo de carga	5-6 horas	Recyclable	≥ de 500 veces
Temperatura de operación	-5°C ~ 40°C	Temperatura de almacenamiento	-10°C ~ 50°C
Volumen	116*65*7.5mm	Peso	135g
Dispositivos de aplicación	CELULAR,PSP,IPOD,IPHONE,IPAD,MP4,PMP,GPS,CÁMARA etc.		

Apéndice C

Abreviaturas utilizadas en las gráficas y tablas para el estudio de las estrategias del control PI.

ts	Tiempo de asentamiento, es una característica de desempeño de la respuesta transitoria de un sistema de segundo orden a la entrada escalón unitario.
Mp	Sobre impulso máximo, es una característica de desempeño de la respuesta transitoria de un sistema de segundo orden a la entrada escalón unitario.
IAE	Integral absoluta del error, que da la información del seguimiento de la señal actual a la señal de referencia.
IAEmax	Índice IAE máximo.
IAEP	Integral absoluta, que compara la señal actual del sistema de control basado en tiempo con la del control basado en eventos.
IAEPmax	Índice IAEP máximo.
IAD	Integral absoluta, que compara el índice IAE basado en tiempo con el basado en eventos.
IADmax	Índice IAD máximo.
Vmax	Es el voltaje máximo que alcanza el valor de la señal de control del controlador PI.
Menv	Representa la cantidad de mensajes enviados por el nodo control en la red CAN-BUS.
Dtb	Valor estimado de duración en días del tiempo de descarga de la batería de 5V a 4.2V, que es el valor mínimo de voltaje para el funcionamiento a altas frecuencias de un micro controlador PIC18F458 de Microchip con módulo CAN-BUS.
SC	Señal de control.
SA	Señal actual.
VB	Volts en la batería.
Indf	Valor indefinido. Se utiliza cuando no pueden ser calculados los valores de ts ó Mp.
E.b	Se utiliza para abreviar la frase: Estrategia basada.

