

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/257684560>

# Reconfiguración Dinámica de Sistemas Distribuidos en Tiempo-Real Basada en Agentes

**Article** in *Revista iberoamericana de automática e informática industrial (RIAI)* · July 2012

DOI: 10.1016/j.riai.2012.05.014

CITATIONS

3

READS

1,695

2 authors:



**Oscar A. Esquivel-Flores**

Instituto Tecnológico y de Estudios Superiores de Monterrey, México.

29 PUBLICATIONS 58 CITATIONS

[SEE PROFILE](#)



**Héctor Benítez-Pérez**

Universidad Nacional Autónoma de México

137 PUBLICATIONS 704 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



PI day [View project](#)



Network control systems [View project](#)

## Reconfiguración Dinámica de Sistemas Distribuidos en Tiempo-Real Basada en Agentes

O. Esquivel-Flores<sup>a,\*</sup>, H. Benítez-Pérez<sup>b</sup>

<sup>a</sup>Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, México D.F., México.

<sup>b</sup>Departamento de Ingeniería de Sistemas Computacionales y Automatización IIMAS, Universidad Nacional Autónoma de México. Apdo. Postal 20-726, Del. A. Obregón, México, D.F. C.P. 01000, México.

### Resumen

Actualmente los sistemas distribuidos están presentes en diversas aplicaciones dentro de la industria e investigación. Este tipo de sistemas está restringido al cumplimiento de condiciones temporales lo que los convierte en sistemas de tiempo crítico que ante la presencia de fallas es necesario tomar decisiones que eviten la pérdida de plazos, disminución de desempeño, inestabilidad o la suspensión total del sistema. Una estrategia para lograr lo anterior es la reconfiguración dinámica de la estructura del sistema con el objetivo de contrarrestar el efecto de las fallas. En este artículo se presenta una propuesta de reconfiguración dinámica y distribuida que se basa en considerar a cada elemento del sistema distribuido como un agente con capacidad comunicación y cooperación que decida en conjunto con otros agentes una acción de reconfiguración. Se propone la modificación de periodos de muestreo de los agentes que sensan los estados del sistema, la disminución de la actividad de los agentes con falla y/o la utilización de algoritmos de voto. Para alguna posible falla en el sistema se hace corresponder un tipo particular de reconfiguración con base en el intercambio de información entre agentes para tomar un acuerdo consensado. El esquema de reconfiguración dinámico que se propone es implementado en el control de vuelo de un prototipo de helicóptero de dos hélices en el que se incluye un sistema distribuido que utiliza un medio de comunicación común entre sus componentes para formar el lazo cerrado de control de un sistema. La respuesta del sistema reconfigurado es explorada por medio de simulaciones numéricas realizadas en Matlab y Simulink utilizando TrueTime como herramienta de simulación de tiempo real. Copyright © 2012 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

### Palabras Clave:

Sistemas Distribuidos, Reconfiguración, Tiempo Real, Agentes.

### 1. Introducción

Algunas características que tienen los sistemas de cómputo distribuidos son la capacidad de mantener consistencia y la de recuperación ante eventos que afecten su funcionamiento sin suspender su ejecución. Estos sistemas están comúnmente sujetos al cumplimiento de restricciones temporales que los enmarcan dentro de los sistemas de tiempo real. Los sistemas de control han aprovechado los beneficios de los sistemas distribuidos para adquirir flexibilidad y adaptabilidad lo que ha resultado en la implementación de estrategias de control distribuido en las que se plantea la división del sistema en entidades individuales que realizan alguna parte del lazo de control. Por otra parte, la

reconfiguración dinámica consiste en mudar o cambiar incrementalmente la configuración original a otra necesaria en tiempo de ejecución (Kramer and Magee, 1985). Estudios recientes consideran la reconfiguración como un cambio de la estructura del sistema que conduce a una nueva representación de sus estados (Benítez-Pérez et al., 2010). En entornos de tiempo real se requiere la reconfiguración de forma dinámica y que afecte lo menor posible a la ejecución del sistema si se considera que el retardo de esta transición es despreciable.

Un sistema distribuido es débilmente acoplado y tiene un alto costo el conocer el estado global del sistema dada la comunicación entre elementos. La planificación en cada agente permite el cumplimiento de las restricciones de tiempo y la toma de decisiones que controlen el acceso a los recursos de forma independiente; sin embargo, esto puede desembocar en decisiones que de manera global no son del todo certeras. Por tanto, los criterios para evaluar cada planificación y el acceso a los recursos dependerán de la coordinación entre planificadores

\*Autor en correspondencia.

Correos electrónicos: [oaefmcc@hotmail.com](mailto:oaefmcc@hotmail.com) (O. Esquivel-Flores), [hector@uxdea4.iimas.unam.mx](mailto:hector@uxdea4.iimas.unam.mx) (H. Benítez-Pérez)

URL: <http://zaz.iimas.unam.mx/~hector/> (H. Benítez-Pérez)

locales y qué cantidad de información del sistema se requiere para lograrlo. En la etapa de diseño de los sistemas de control distribuidos, también nombrados sistemas de control en red (SCR) se considera como elemento importante de análisis el determinar un rango de los periodos de muestreo en los que es sistema presenta un rendimiento aceptable. En este trabajo se propone una estrategia de reconfiguración dinámica que modifica los periodos de muestreo de los sensores del sistema con dos propósitos particulares: lograr la planificabilidad global del sistema distribuido y la planificabilidad del medio de comunicación común. Lo anterior considera los estados del sistema en términos de los retardos de tiempo ocasionados. La planificabilidad local de cada agente del sistema distribuido garantiza la planificabilidad global del sistema, lo que asegura que cada tarea que ejecuta un agente se cumpla dentro de los plazos establecidos.

Almeida (Almeida, 2001) propone un modelo de reconfiguración que se centra en el diseño previo del sistema y las modificaciones necesarias para alcanzar nuevos estados. En este modelo las acciones de reconfiguración se definen en términos de los cambios específicos y las restricciones por cumplir durante la reconfiguración. Estas modificaciones son aplicadas bajo la supervisión de un manejador de cambios que utiliza la información de los estados originales del sistema y la relación entre entidades; sin embargo esto conduce a una centralización de las decisiones de reconfiguración, una alternativa para esto es la reconfiguración distribuida que involucra la información local de todos los componentes del sistema para obtener un estado global.

Los trabajos sobre modelos de reconfiguración implementados reportan pocos resultados aplicables a sistemas de control en red y en entornos de tiempo real. El objetivo de este trabajo es estudiar la reconfiguración de un sistema distribuido en tiempo real considerando posibles escenarios con falla con el propósito de cuantificar el uso de un mecanismo de acuerdo entre los agentes del sistema para decidir los cambios necesarios que disminuyan el efecto de las fallas, con el estricto cumplimiento de las restricciones de tiempo real. Se propone en este trabajo la interacción entre elementos inteligentes (Desforges and Archimède, 2006) que están dotados de distintas capacidades de comunicación y negociación para conseguir la reconfiguración distribuida.

La metodología utilizada para estudiar el impacto de la reconfiguración de un sistema distribuido en tiempo real por medio de agentes consistió en tipificar un falla en los sensores del sistema, para cada falla particular se define una plan de reconfiguración definido fuera de línea, el acuerdo entre agentes determina la ejecución del plan de reconfiguración estipulado. Por medio del cálculo de la integral del valor absoluto del error entre la señal de referencia y la salida del sistema se evalúa el efecto de los cambios realizados. En la figura 1 se muestra la evolución del proceso de reconfiguración planteado.

Para implementar esta estrategia de reconfiguración se utilizó como caso de estudio el control de vuelo de un prototipo de helicóptero al cual le fue incluido un sistema distribuido en el lazo de control. Se considera que este caso de estudio funciona bajo restricciones de tiempo por lo que existe la presencia de re-

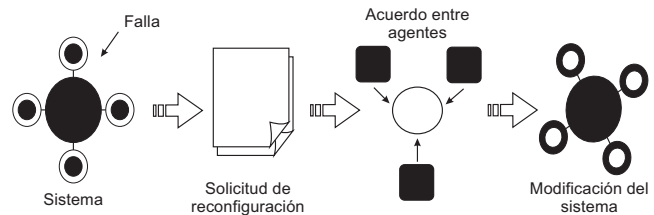


Figura 1: Reconfiguración por medio de agentes

tardos en la transmisión y recepción de datos que modifican la dinámica del sistema. Lo anterior, en consecuencia, modifica la estructura de los estados del sistema, por lo tanto el sistema de control debe tomar en cuenta las transiciones como los nuevos estadios en términos de pérdida de información y de retardos de tiempo. Esto lleva al modelado por escenarios tomando en cuenta la globalidad de las transiciones y las condiciones por elementos.

El trabajo está organizado de la siguiente forma: En la sección 2 se plantean las consideraciones necesarias para la reconfiguración y se muestra un análisis de la planificación del sistema para lograr transiciones viables al realizarla, se menciona la técnica de voto empleada como mecanismo auxiliar de reconfiguración. En la sección 3 se plantea la forma en que un sistema distribuido es considerado como un sistema multiagente y de qué manera se lleva a cabo el acuerdo para la reconfiguración. La sección 4 muestra el modelo dinámico y de control del prototipo de helicóptero utilizado como caso de estudio. En la sección 5 se muestran algunos resultados de simular el sistema bajo escenarios sin falla, con falla y falla con reconfiguración. Algunas conclusiones son presentadas al final.

## 2. Consideraciones para Realizar la Reconfiguración

En esta propuesta se estudia el impacto que provoca la reconfiguración de los periodos de muestreo en un sistema distribuido en tiempo real. Se utiliza la redundancia de Hardware y los algoritmos de voto como un método para tolerar de fallas.

### 2.1. Tiempos de Muestreo en un Sistema de Control en Red

Actualmente la tendencia de los sistemas que integran al cómputo, a la comunicación y al control usan arquitecturas de control con una red de comunicación común, esto mejora la eficiencia, flexibilidad y confiabilidad de los sistemas distribuidos. No obstante, esta arquitectura conlleva diferentes tipos de retardo entre sensores-controladores-actuadores consecuencia de compartir el medio de comunicación y el proceso propio de comunicación. (Cervin et al., 2003) discuten la posibilidad de analizar el desempeño de controladores bajo los efectos de modificaciones en los intervalos de muestreo y retardos de comunicación.

El diseño de la arquitectura de los sistemas de control en red (SCR) son analizados en (Lian et al., 2001, 2002) con la finalidad de determinar un rango aceptable de los periodos de muestreo dentro del cual el sistema se desempeña aceptablemente. El análisis se basa fundamentalmente en la gráfica de

desempeño derivada de comparar el rendimiento del control y los periodos de muestreo. En ésta, se contrasta el desempeño para tres tipos de sistemas de control: control continuo, control digital y control en red y se estipula la existencia de puntos de degradación de desempeño. Se considera el control continuo como el ideal donde se mantiene un buen desempeño del sistema en todo momento ya que el rendimiento no está en función del periodo de muestreo. Para el caso del control digital y el control en red se establecen puntos de degradación de desempeño en los cuales la curva de rendimiento cambia de aceptable a no aceptable debido al tamaño del periodo de muestreo que se está considerando. Para el control digital existe el punto con periodo de muestreo en que el desempeño del control digital pasa de ser no aceptable a aceptable y en el caso del control en red existen dos puntos y con periodos y dentro de los cuales el control en red se desempeña de manera óptima, fuera de estos puntos los periodos pequeños resultan en tráfico de red pesado que conduce a la posibilidad de retardo y/o el aumento de la pérdida de datos, mientras que periodos de muestreo largos degradan al sistema.

Como se ha mencionado, el propósito de este trabajo es evaluar el impacto que provoca la reconfigurar un sistema de control en red. Estas estrategias consisten en modificar los parámetros de planificación, utilizar algoritmos de voto, disminuir la tasa de muestreo de los sensores con falla o eliminar cualquier lectura errónea. Alguna de estas estrategias será elegida por medio de un acuerdo entre agentes tomando en cuenta los retardos de tiempo asociados para cada evento específico. La relación de estabilidad global para considerar transiciones viables se logra mediante el análisis respectivo aunque sale de objetivo de este trabajo. En este artículo se presentan los resultados de diversas modificaciones de los retardos en términos de planificación, lo que conlleva a una pérdida de plazos por tráfico que modifican el desempeño del caso de estudio aunque el controlador haya sido diseñado en los rangos de retardos de tiempo adecuados como se presenta en (Méndez-Monroy and Benítez-Pérez, 2009).

La parte novedosa consiste en decidir las acciones de reconfiguración con base en un acuerdo tomado entre agentes por medio del intercambio de información y mostrar que al asegurar la planificación local de tareas de cada agente se garantiza la planificación global del sistema. Es importante mencionar que adicionalmente se busca la planificación del medio de comunicación común dado que al llevarse a cabo el acuerdo se incrementa la tasa de transmisión de datos. Esta planificación de la red se realiza por medio de un agente planificador que asigna una ventana de tiempo de transmisión a los demás agentes que requieren transmitir, de esta manera se evitan colisiones y sobrecarga en la red.

## 2.2. Planificación

Cada uno de los agentes del sistema de control distribuido realizan un conjunto de tareas  $\tau_i = \{\tau_1, \tau_2, \dots, \tau_n\}$  en el que la tarea  $\tau_k$  realiza alguna acción del lazo de control retroalimentado, las demás tareas son tareas adicionales del sistema y son relevantes en cuanto al tiempo de procesamiento que requieren.

Las tareas periódicas que se ejecutan en cada agente son planificadas por medio de un algoritmo por prioridades. En particular, si cada agente realiza  $n$  tareas periódicas con periodo  $p_i$  y tiempo de ejecución  $c_i$  con  $i = 1, 2, \dots, n$ ; la planificación de las tareas que se lleva a cabo cada agente se garantiza (Liu and Layland, 1973) si se cumple :

$$U < n(2^{\frac{1}{n}} - 1) \quad (1)$$

donde la utilización del procesador (agente) está definida como  $U = \sum_{i=1}^n \frac{c_i}{p_i}$ , así la planificación local de cada agente estará en función de los periodos de las tareas y sus tiempos de ejecución.

En este trabajo la política de planificación utilizada es el algoritmo Rate Monotonic que es un algoritmo de prioridades fijas. Para cualquier conjunto arbitrario de tareas periódicas, el mínimo valor máximo de utilización del procesador para el cual el conjunto de tareas es planificable estará dado por 1. Esta cota decrece mientras aumenta, de tal forma que para valores grandes de la mínima cota máxima converge a  $\ln 2 \approx 0.69$ . Entonces, la utilización del procesador debe ser menor al 69 % para asegurar la planificación de cualquier conjunto arbitrario de tareas.

El agente planificador de red utiliza una tarea periódica para asignar ancho de banda a cada agente que requiere transmitir por medio de una ventana de tiempo en la cual se utiliza el canal de comunicación. El agente planificador genera un identificador  $id$  que se escribe en la memoria distribuida la cual es visible por todos los agentes, si en el momento en que un agente está listo para transmitir el valor del  $id$  en la memoria distribuida es igual a su propio identificador local el agente transmite, de lo contrario deja pasar un periodo hasta una nueva solicitud de transmisión, momento en el que se consulta de nuevo la memoria distribuida donde un nuevo  $id$  está escrito.

El lapso de tiempo que transcurre para completar el ciclo de control retroalimentado en el SCR está dado por:

$$T = nt_s + t_{cm}^{sc} + t_c + t_{cm}^{ca} + t_a \quad (2)$$

donde  $t_s$  es el tiempo consumido por los sensores y  $n$  es el número de sensores en el sistema,  $t_{cm}^{sc}$  es el tiempo de comunicación entre sensor y controlador,  $t_c$  es el tiempo que tarda en calcularse la ley de control,  $t_{cm}^{ca}$  es el tiempo de comunicación entre controlador y actuador,  $t_a$  es el tiempo del actuador. Cabe señalar que la lectura de los  $n$  sensores del sistema se hace de manera secuencial dado que existe un ligero desfaseamiento para el inicio de cada sensor, esto repercute en tomar  $n$  tiempos de sentido proporcionales a este desfaseamiento.

Cuando se incluye el enmascaramiento de fallas, el tiempo para completar el ciclo de control retroalimentado es:

$$T = nt_s + t_{cm}^{sc} + t_{cm}^{fst} + t_{cm}^{fsc} + t_c + t_{cm}^{ca} + t_a \quad (3)$$

donde  $t_{cm}^{fst}$  es el tiempo de comunicación entre los sensores y la unidad de enmascaramiento y  $t_{cm}^{fsc}$  es el tiempo entre la falla y el acuerdo.

El proceso de reconfiguración consiste en modificar los periodos y los consumos de tareas para contrarrestar el efecto de

algún evento que altere la condición de planificación, tal modificación requiere el cálculo en línea de nuevos parámetros de tareas. Se analiza la planificación del conjunto de tareas por medio de la obtención de un conjunto factible de periodos que aseguran la restricción (1) bajo un algoritmo de planificación de prioridades fijas como el *Rate Monotonic*. La obtención de este conjunto de periodos depende de un conjunto adicional de periodos que sirve como cota máxima el cual puede tomarse de diferente manera de acuerdo al caso de estudio. No obstante que la búsqueda de este nuevo conjunto de periodos puede ser computacionalmente demandante, el tiempo de cálculo disminuye al tratarse como un problema de programación entera. La aportación del presente trabajo consiste en mostrar que efecto de la reconfiguración ayuda a disminuir los efectos no lineales del sistema cuando aparecen un número considerable de retardos de tiempo producto de escenarios con falla. La propuesta de reconfiguración en el sentido de la planificación consiste en que dado un evento que modifique la planificabilidad local de algún agente del sistema:

- Se calcule un nuevo conjunto de periodos para retomar la planificabilidad.
- Realizar el cambio en los periodos correspondientes.
- Mantener la planificabilidad global.

Debido a lo anterior, a partir del cambio de los periodos de muestreo se diseña un controlador adecuado a estos intervalos ajustándose cualquier cambio de periodos dentro de estos márgenes. Por otra parte, el ajuste en los periodos de muestreo provocará un retardo menor en el desempeño del sistema que es relevante mostrar; así mismo se induce un retardo adicional producto de la reconfiguración:

$$T_r = t_{cp}^r + t_{rp}^r \quad (4)$$

donde  $t_{cp}^r$  es el tiempo del cálculo de nuevos periodos y  $t_{rp}^r$  el tiempo necesario para cambiar los periodos.

La planificación local no se realiza por medio de un acuerdo entre agentes, sin embargo es necesario que ésta se asegure para lograr un efecto positivo de la reconfiguración. El tipo de reconfiguración se toma por medio del acuerdo.

### 2.3. Enmascaramiento de Falla

Un método utilizado en sistemas de detección e identificación de fallas consiste en el uso de los algoritmos de voto que enmascaran las fallas ocurridas principalmente en la lectura de los estados del sistema. Distintas propuestas de estos algoritmos han sido presentadas y clasificadas en dos grupos: seguros y confiables (Benítez-Pérez and García-Nocetti, 2005). Un esquema de voto utilizado actualmente es el de peso promedio (Latif-Shabgahi et al., 2003; Latif-Shabgahi, 2004), el cual usa triple redundancia modular para enmascarar fallos de hardware y software en tiempo de ejecución. Las salidas de 3 módulos independientes que operan en paralelo con las mismas entradas alimentan una unidad de voto que sirve de árbitro para operar entre ellos.

En nuestra propuesta se trabajan módulos de sensado con triple redundancia cada uno, esta técnica corresponde a un tipo de voto seguro. Cada sensor alimenta una unidad de voto con una lectura local y dos lecturas provenientes de los sensores de su mismo módulo, el resultado que genera tal unidad de voto enmascara la falla de lectura.

### 3. Sistema Multiagente y Acuerdo entre Agentes

Un sistema distribuido está formado por distintas entidades, cada una de ellas es un agente del sistema y realizan varias tareas específicas, tales tareas pueden tener una relación de dependencia con una o varias tareas de cualquier otro agente. Actualmente el uso de elementos inteligentes (*smart devices*) permite integrar funciones adicionales además de las tareas básicas de sensado o el cálculo de algoritmos, por tal razón estos dispositivos son utilizados ampliamente para acciones de control y reconfiguración (Benítez-Pérez and García-Nocetti, 2005). En el contexto de la inteligencia artificial distribuida un agente puede caracterizarse como un elemento de hardware, software o ambos. Estos últimos son altamente eficientes en implementaciones de control distribuido ya que pueden realizar varias tareas requeridas en el lazo de control y la tareas adicionales como las de supervisión (García and Cenjor, 2007; García et al., 2010).

En un SCR los sensores obtienen información de los estados del sistema y son altamente sensibles a las fallas que se producen en ellos. En esta propuesta se implementan los sensores del sistema como agentes computacionales y de hardware para dotarlos de flexibilidad y además de sensar, realice tareas de comunicación con otros agentes.

Como se refiere en (Cenjor and García, 2005) un agente puede considerarse como un sistema informático que realiza acciones autónomas, percibe el entorno y responde ante los cambios que ocurren en él, es proactivo, es decir que logra sus objetivos mediante iniciativas e interactúa con otros agentes planteando procesos de negociación para lograr un objetivo específico. La implementación que se realiza en este trabajo consta de varios procesadores que se comunican a través de una red de comunicación común, cada procesador funciona con un kernel de tiempo real. Todos los procesadores colaboran para ejecutar un ciclo de control. Como se mencionó, con base en los agentes sensores se acuerda el tipo de modificaciones que experimentará el sistema cuando un escenario de falla se presenta. Cada sensor percibe el entorno por medio de su visión local y colabora con otros agentes para tomar decisiones que afectarán el estado global. El uso de redundancia de sensores no solo responde a la necesidad de seguridad en sistemas críticos si no que sugiere el esquema multiagente para realizar una reconfiguración dinámica y distribuida. En el contexto del modelo de reconfiguración anteriormente descrito (sección 1) cada agente sensor funciona como un controlador de cambios. En la figura 2 se muestra la modificación del modelo de reconfiguración centralizado usando agentes individuales con capacidad de comunicación, acuerdo y reconfiguración. La percepción de cada agente sobre su entorno se comparte con otros agentes de tal manera que se integra un conocimiento global de los estados del sistema.



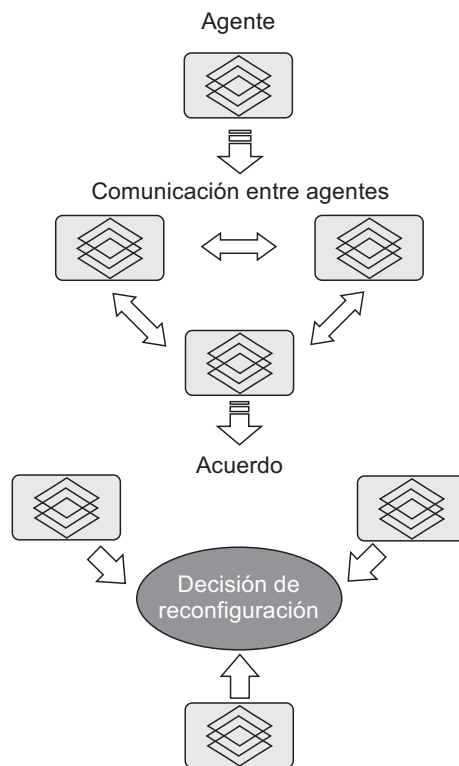


Figura 2: Reconfiguración distribuida con base en agentes

### 3.1. Estructura Multiagente

En este trabajo la redundancia de agentes sensores constituye la base del acuerdo que define las acciones de reconfiguración. Cada uno de estos agentes tiene capacidad de:

- Sensar las señales de los estados del sistema.
- Recibir una tipificación de las posibles fallas en el sistema.
- Comunicarse con otros agentes.
- Acordar un tipo de reconfiguración conforme a la falla presentada.
- Modificar uno o varios parámetros del sistema.

Cada agente hace la lectura de los estados del sistema y obtiene una tipificación de falla en caso de existir, esta información es comunicada por medio del paso de mensajes a través de la red a otros dos agentes que sensan la misma señal que él. Con estos datos se realiza el acuerdo sobre el tipo de modificación que experimentará el sistema. El resultado generado se toma con base a una mayoría calificada entre los diversos elementos.

La redundancia de sensores está organizada por bloques. Para cada señal que se sensa existe un bloque de sensado formado por tres sensores cada uno, para un sistema con  $m$  señales por sensar se tienen  $B_1, B_2, \dots, B_m$  bloques de sensado con 3 sensores cada uno. Denotaremos como  $a_{ij}$  al agente  $j$  que pertenece

al bloque  $B_i$ . Cada sensor realiza la lectura de un estado del sistema  $x_{ij}$  junto con una tipificación de falla  $f_{ij}$  siempre y cuando este sea el caso. Así, cada agente de sensado  $a_{ij}$  tiene en el tiempo  $k$  un valor de la señal de los estados del sistema  $x_{ij}(k)$  y  $f_{ij}(k)$  donde  $i \in \{1, 2, \dots, m\}$ , y  $j \in \{1, 2, 3\}$ .

Distintas técnicas son utilizadas para la detección y manejo de fallas en sensores y actuadores; en este trabajo no se profundiza en la identificación de tales fallas, sin embargo se asume que el sistema está provisto de un mecanismo que permite identificar y tipificar las fallas que se presentan en la lectura de datos. Con el conocimiento previo de los efectos de estas fallas es posible elegir una estrategia que permite tomar una acción para disminuirlos.

Debido a que el sistema es de tiempo real, cada agente del sistema realiza tareas periódicas con periodo  $p$  y tiempo de ejecución o consumo  $c$ . Las tareas de los agentes sensores se subdividen en segmentos en los que se realiza el sensado, intercambio de información, acuerdo y reconfiguración (ver figura 3).

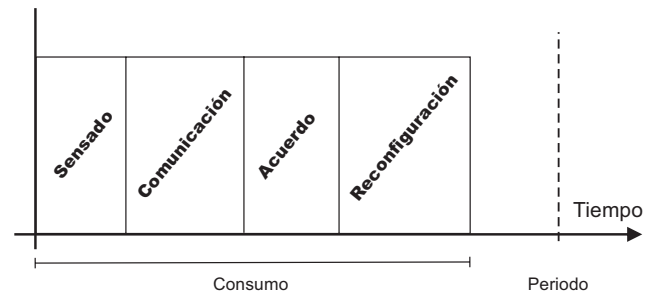


Figura 3: Acciones del agente

Con la información  $(x_{ij}, f_{ij})$  que cada agente sensor  $a_{ij}$  obtiene del medio se acuerda la reconfiguración del sistema y para hacerlo es necesario que el intercambio de información se realice antes de la finalización del plazo de la tarea, sin embargo no siempre es posible esto. Para la realización del acuerdo aumenta la demanda de la red lo que conduce al incremento del tráfico, saturación o pérdida de datos; esto se traduce en información incompleta para realizar el acuerdo y la reconfiguración. Para resolver lo anterior se propone que la información que no se disponga en algún periodo de control sea reconstruida por medio de un valor predefinido fuera de línea que sea estimado producto de la experiencia o sea un valor promedio de todos los valores sensados previamente.

### 3.2. Acuerdo

El mecanismo de acuerdo se realiza intercambiando la información de las vistas locales de cada agente, una vez consolidada la lectura de las señales de control se define un plan de reconfiguración. Previamente se mencionó que la redundancia de sensores aumenta la confiabilidad en las mediciones; no obstante, fallas en los sensores pueden provocar inestabilidad en el sistema. La lectura de las señales de control representa el estado dinámico del sistema en cada tiempo  $t$ . Bajo un escenario con falla la información se distribuye en todos los agentes sensores

para dar lugar al acuerdo y la ejecución del plan de reconfiguración evitando la suspensión del sistema, mal desempeño o una falla catastrófica.

La información de lectura y tipificación de falla  $(x_{ij}, f_{ij})$  que cada agente posee es compartida con los otros *pares* del mismo bloque de sensado al que pertenece. Cada agente  $a_{ij}$ , donde  $i = 1, 2, \dots, m$ ;  $j = 1, 2, 3$ , cuenta entonces con la tupla de datos:

$$y_{ij} = \{(x_{i1}, f_{i1}), (x_{i2}, f_{i2}), \dots, (x_{ij}, f_{ij})\}$$

Con estos valores cada agente realiza el acuerdo  $A(y_{ij})$  en colaboración con los otros agentes del mismo bloque con base en (5). Cada agente  $j$  del bloque de lectura  $i$  posee la lectura de un estado del sistema en el tiempo  $t_1$ : la señal de control  $x_{ij}$  y una posible falla  $f_{ij}$ . En el tiempo  $t_2$  el agente ha intercambiado su *información local* con los otros dos agentes de su mismo bloque  $i$ . Al paso de un tiempo  $t_3$  cada agente habrá calculado el valor del acuerdo  $A(y_{ij})$  del acuerdo (figura 4).

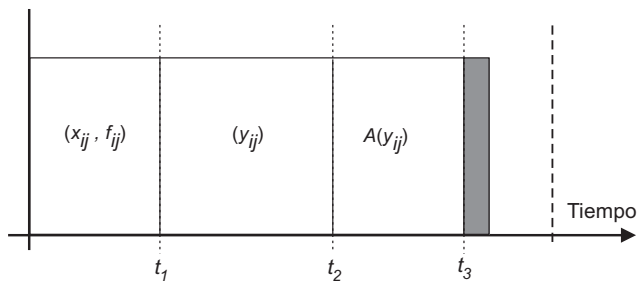


Figura 4: Información que cada agente obtiene durante el paso del tiempo

Tómese el acuerdo  $A(y_{ij})$  como una función que depende de la tupla de valores  $y_{ij}$ . Esta función devolverá el valor *acordado* del estado del sistema  $\bar{x}$  basándose en la tipificación de la falla que posee cada sensor en el tiempo  $t_k$ , dicho valor acordado será el que prevalezca y será enviado al controlador.

Dados  $m$  bloques de sensado y triple redundancia de sensores, la función de acuerdo  $A(y_{ij})$  para  $i = 1, 2, \dots, m$ ;  $j = 1, 2, 3$  está definida de la siguiente forma:

$$A(y_{ij}) = \begin{cases} \bar{x} = x_{i1} & , \quad f_{i1} = f_{i2} = f_{i3} \\ \bar{x} = \frac{(x_{i1} + x_{i2})}{2} & f_{i1} = f_{i2} \neq f_{i3} \\ \bar{x} = \frac{(x_{i1} + x_{i3})}{2} & f_{i1} = f_{i3} \neq f_{i2} \\ \bar{x} = \frac{(x_{i2} + x_{i3})}{2} & f_{i1} \neq f_{i2} = f_{i3} \\ \bar{x} = x_{i1} & f_{i1} \neq f_{i2} \neq f_{i3} \end{cases} \quad (5)$$

Esta función de acuerdo se interpreta de la siguiente manera:

- Si un agente del bloque  $i$  ha recibido de los otros dos agentes de su mismo bloque una tipificación de falla igual a la propia, el valor de la lectura  $\bar{x}$  será valor local sensado por el agente  $a_{ij}$  que hace la llamada al acuerdo.

- Si el agente recibe de los otros dos agentes una tipificación de falla igual a la suya y otra distinta, se descarta el valor que difiere y la lectura acordada  $\bar{x}$  será el promedio de los datos sensados con igual tipificación de falla.
- Si el agente recibe dos tipificaciones de falla diferentes a la suya pero iguales entre sí, se toma el promedio de los datos sensados con igual tipificación.
- Si el agente recibe dos valores de tipificación de falla distintos al suyo y aún diferentes entre sí, entonces la lectura acordada será el dato local del agente  $a_{ij}$ .

Este acuerdo lo hace cada uno de los agentes que sensan los estados del sistema para dar paso a la reconfiguración del sistema. En este trabajo nos enfocamos en las fallas que se presentan en el sensado de datos debido a que en un sistema con alto grado de seguridad y confiabilidad es importante responder a fallas en los sensores o en los datos que estos registran. Las fallas que se consideran aquí son fallas de lectura como ruido, corrimiento y retraso. Las pruebas de reconfiguración fueron realizadas induciendo una falla a un sensor de un bloque de sensado en cualquier momento de la ejecución del sistema.

#### 4. Caso de Estudio

Diversos trabajos han enfocado su interés en el control de vuelo de vehículos aéreos autónomos motivado por el gran impacto que tienen en aplicaciones militares y civiles (Romero et al., 2010). En esta propuesta, se utiliza el prototipo de helicóptero de dos hélices de la marca Quanser (Quanser, 2006) para evaluar el impacto de la reconfiguración distribuida con las características planteadas en la sección precedente. El modelo dinámico del vuelo y el modelo de control de posición del helicóptero están implementados por medio de la herramienta Matlab/Simulink™ y con el simulador de tiempo real True-Time (Ohlin et al., 2007).

En este caso de estudio se requiere de la lectura de cuatro señales correspondientes a los ángulos de cabeceo y guiñado además de sus respectivas derivadas para calcular la ley de control, con esta ley de control se modifica el voltaje de los propulsores de las hélices. Trabajos interesantes que modelan el control de vuelo de helicópteros y su implementación son presentados por (García-Sanz et al., 2006; Castillo et al., 2007; López-Martínez et al., 2007) donde se resalta el modelo dinámico de helicópteros de hasta 4 hélices y las técnicas de control no lineal utilizadas.

##### 4.1. Dinámica del Helicóptero 2-DOF Quanser

El prototipo utiliza dos motores de corriente directa que hacen girar los propulsores que controlan la elevación de la nariz del helicóptero sobre el eje horizontal o cabeceo y la desviación respecto al eje vertical también conocido como guiñado. Estos ángulos son medidos por medio de *encoders* de alta resolución.

El modelo matemático de la dinámica del prototipo del helicóptero (figura 5) es no lineal con dos grados de libertad correspondientes al ángulo de cabeceo (*pitch degree*)  $\theta$  y al ángulo de

guiñado (*yaw degree*)  $\psi$ . El ángulo de cabeceo es definido positivo cuando la nariz del helicóptero está sobre el eje horizontal y el ángulo de guiñado es positivo cuando gira en sentido de las manecillas del reloj. Existen las fuerzas  $F_p$ ,  $F_y$  que actúan perpendicularmente a los ejes con lo que el torque del eje cabeceo es aplicado a la distancia  $r_p$  y el torque de guiñado a la distancia  $r_y$ . La fuerza de gravedad  $F_g$  genera un torque en el centro que jala hacia abajo la nariz del helicóptero. El centro de masa es la distancia  $l_{cm}$  medida desde el ángulo pitch y a lo largo del cuerpo del helicóptero.

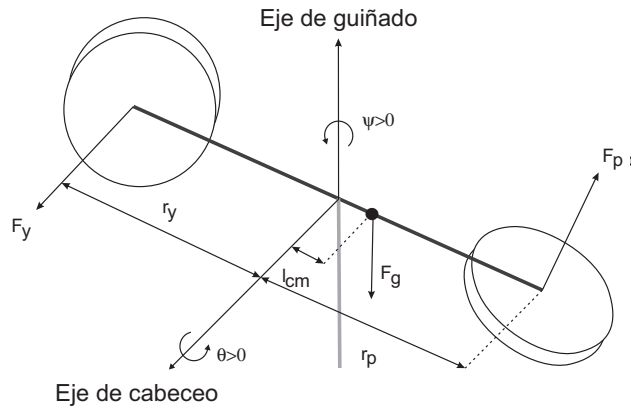


Figura 5: Dinámica del helicóptero 2-DOF

Las ecuaciones de movimiento se calculan utilizando el método de Lagrange. La cinemática del centro de masa del helicóptero es utilizada para calcular la energía cinética y potencial del sistema y por medio del método de Euler-Lagrange se derivan las ecuaciones no lineales que describen el movimiento del helicóptero. El análisis de movimiento, energía potencial y cinética, así como la obtención de las ecuaciones no lineales de movimiento del helicóptero puede consultarse en (Quanser, 2006), aquí nos enfocaremos en el modelo lineal de espacio de estados y en modelo del controlador propuesto.

#### 4.2. Modelo Lineal en Espacio de Estados y Modelo de Control

En la tabla 1 se muestran algunas especificaciones y parámetros como longitudes, masas y momentos de inercia asociados al helicóptero.

Debido a que el sistema del helicóptero es un sistema no lineal, se requiere linealizar alrededor del punto de equilibrio ( $\theta_0 = 0, \psi_0 = 0, \dot{\theta}_0 = 0, \dot{\psi}_0 = 0$ ), de esta linealización se obtienen las ecuaciones de movimiento:

$$(J_{eq,p} + m_{heli}l_{cm}^2)\ddot{\theta} = K_{pp}V_{m,p} + K_{py}V_{m,y} - B_p\dot{\theta} - m_{heli}gl_{cm} \quad (6)$$

$$(J_{eq,y} + m_{heli}l_{cm}^2)\ddot{\psi} = K_{py}V_{m,p} + K_{yy}V_{m,y} - B_y\dot{\psi} - 2m_{heli}l_{cm}^2\dot{\theta}\dot{\psi} \quad (7)$$

Al sustituir el estado  $x = [\theta, \psi, \dot{\theta}, \dot{\psi}]'$  en (6) y (7), y resolviendo para  $\dot{x}$  resulta el modelo lineal de espacio de estados:

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{B_p}{J_{eq,p} + m_{heli}l_{cm}^2} & 0 \\ 0 & 0 & 0 & -\frac{B_y}{J_{eq,y} + m_{heli}l_{cm}^2} \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{K_{pp}}{J_{eq,p} + m_{heli}l_{cm}^2} & \frac{K_{py}}{J_{eq,p} + m_{heli}l_{cm}^2} \\ -\frac{K_{yp}}{J_{eq,y} + m_{heli}l_{cm}^2} & -\frac{K_{yy}}{J_{eq,y} + m_{heli}l_{cm}^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x$$

donde  $K_{pp}$ ,  $K_{yy}$ ,  $K_{py}$ ,  $K_{yp}$  son las constantes de torque usadas para obtener los torques acoplados que actúan sobre los ejes de cabeceo y guiñado. Para el modelo de espacio de estados los vectores de entrada  $u$  y salida  $y$  son  $u = [V_{m,p}, V_{m,y}]'$  y  $y = [x_1, x_2, x_3, x_4]'$ ,  $V_{m,p}$  es el voltaje de entrada del motor del eje de cabeceo y  $V_{m,y}$  es el voltaje del motor del eje de guiñado. Nótese que por la identidad en la matriz de salida todos los estados son medibles.

El controlador implementado para regular el vuelo del helicóptero es un modelo del tipo FF+LQR+I (*Feed Forward + Linear Quadratic Regulator + Integrator*), el cual regula el eje de cabeceo del helicóptero usando alimentación hacia adelante (*Feed Forward FF*) y compensadores de velocidad proporcional (PV), así como al eje de guiñado usando control de velocidad proporcional, además incluye un integrador en el lazo de control retroalimentado. En este caso el controlador es diseñado para cada combinación de retardos de tiempo dados por la acción de planificación y la consecuente reconfiguración. Sin embargo, aún cuando no es posible establecer todas las combinaciones se toman en cuenta aquellas que no conducen a un error catastrófico al sistema. La implementación del lazo cerrado de control y del controlador FF+LQR+I se muestran en las figuras 6 y 7 respectivamente.

#### 4.3. Modelo del Sistema Distribuido Multiagente

El sistema multiagente consta de 16 procesadores con kernel de tiempo real, conectados por medio de una red CAN con protocolo CSM/AMP y tasa de transmisión de datos de 1000000 bits/seg. y sin probabilidad de pérdida de datos (figura 8). Cada bloque, así como el bloque de red pertenecen a las herramientas del software de simulación de tiempo real Truetime y son compatibles con Matlab y Simulink que y se insertaron 12 agentes sensores organizados en bloques de tres sensores cada uno para leer las señales de los estados del sistema  $x = [\theta, \psi, \dot{\theta}, \dot{\psi}]'$ . El



Tabla 1: Parámetros del modelo en Simulink/Matlab del Helicóptero 2-DOF

Símbolo	MATLAB	Dehacripción	Valor	Unidad
$B_{cq,p}$	B_eq_p	Amortiguamiento eje pitch	0.800	N/V
$B_{cq,y}$	B_eq_y	Amortiguamiento eje yaw	0.318	N/V
$m_{heli}$	m_heli	Masa total helicóptero	1.387	kg
$m_{m,p}$	m_motor_p	Masa del motor pitch	0.292	kg
$m_{m,y}$	m_motor_y	Masa motor yaw	0.128	kg
$m_{shield}$	m_shield	Masa del propulsor escudo	0.167	kg
$m_{props}$	m_props	Masa propulsores pitch, yaw, escudos y motores	0.754	kg
$m_{body,p}$	m_body_p	Masa sobre eje pitch	0.633	kg
$m_{body,y}$	m_body_y	Masa sobre el eje yaw	0.667	kg
$m_{shaft}$	m_shaft	Masa del mango del eje pitch	0.151	kg
$L_{body}$	L_body	Longitud total helicóptero	0.483	m
$l_{cm}$	l_cm	Centro de masa sobre eje pitch	0.186	cm
$L_{shaft}$	L_shaft	Longitud mango del eje yaw	0.280	m
$J_{body,p}$	J_m_p	Momento de inercia sobre el eje pitch	0.012	$kg \cdot m^2$
$J_{body,y}$	J_m_y	Momento de inercia sobre el eje yaw	0.013	$kg \cdot m^2$
$J_{shaft}$	J_shaft	Momento de inercia del mango sobre eje yaw y punto final	0.004	$kg \cdot m^2$
$J_p$	J_p	Momento de inercia motor/escudo sobre pivote pitch	0.018	$kg \cdot m^2$
$J_y$	J_y	Momento de inercia motor/escudo sobre pivote yaw	0.008	$kg \cdot m^2$
$J_{eq,p}$	J_eq_p	Momento total de inercia sobre pivote de pitch	0.038	$kg \cdot m^2$
$J_{eq,y}$	J_eq_y	Momento total de inercia sobre pivote de yaw	0.043	$kg \cdot m^2$

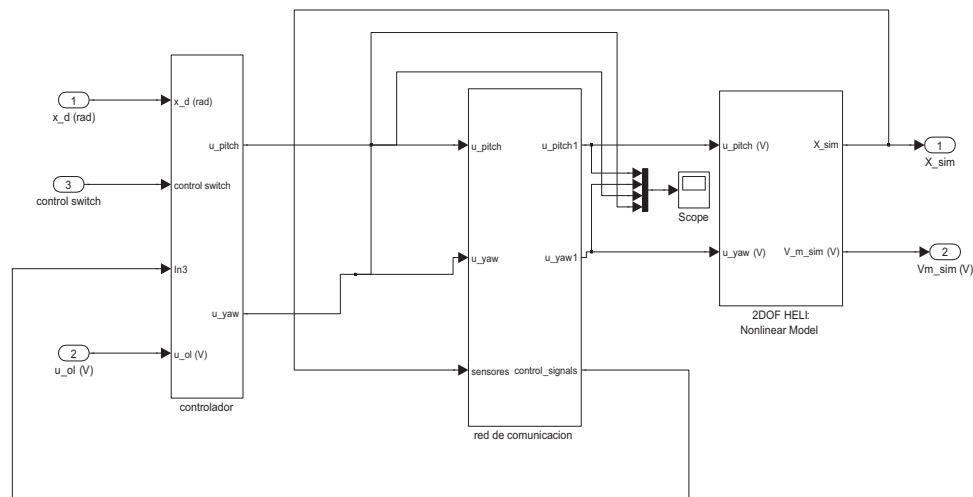


Figura 6: Modelo en Matlab/Simulink del Helicóptero 2-DOF. Simulación del lazo cerrado de control

agente controlador, extrema izquierda del modelo (figura 8), es el agente controlador que utiliza los valores provenientes de los agentes sensores para calcular las ley de control. Dos agentes actuadores, localizados a la extrema derecha abajo (figura 8), reciben los datos del controlador y accionan sobre la planta. Por último se incluye un agente planificador, extrema derecha arriba (Figura 8), que asigna periódicamente una ventana de tiempo para que los demás agentes tengan acceso a la red.

En este sistema se utilizó redundancia de sensores para aumentar una unidad de voto que, por medio del algoritmo de peso

promedio (Latif-Shabgahi et al., 2003), consolida la lectura de la señal y con ello se asegura el error mínimo en la lectura de los datos. En cada bloque  $B_i$  de agentes sensores, alguno de los agentes es perturbado con una falla como podría ser el caso de corrimiento, muerte, ruido o falla bizantina, lo que desencadena el inicio del proceso de reconfiguración. El agente controlador que realiza el cálculo del algoritmo de control recibe los datos enviados por los agentes sensores y envía el resultado a los agentes actuadores que modifican el voltaje de los propulsores del helicóptero. El agente planificador se encarga de organizar

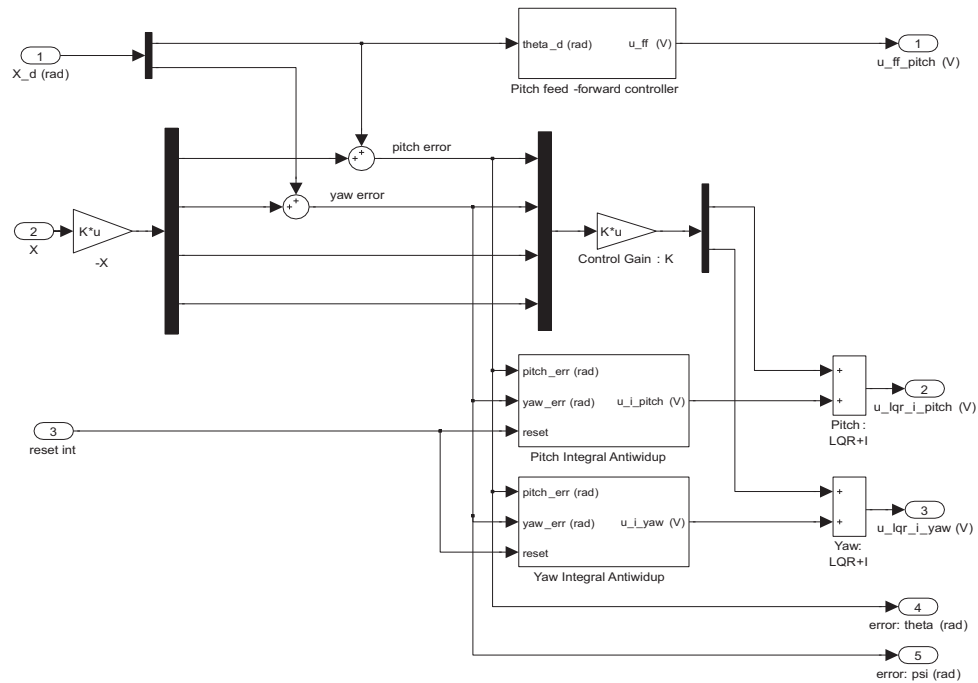


Figura 7: Modelo en Matlab/Simulink del controlador FF+LQR+I

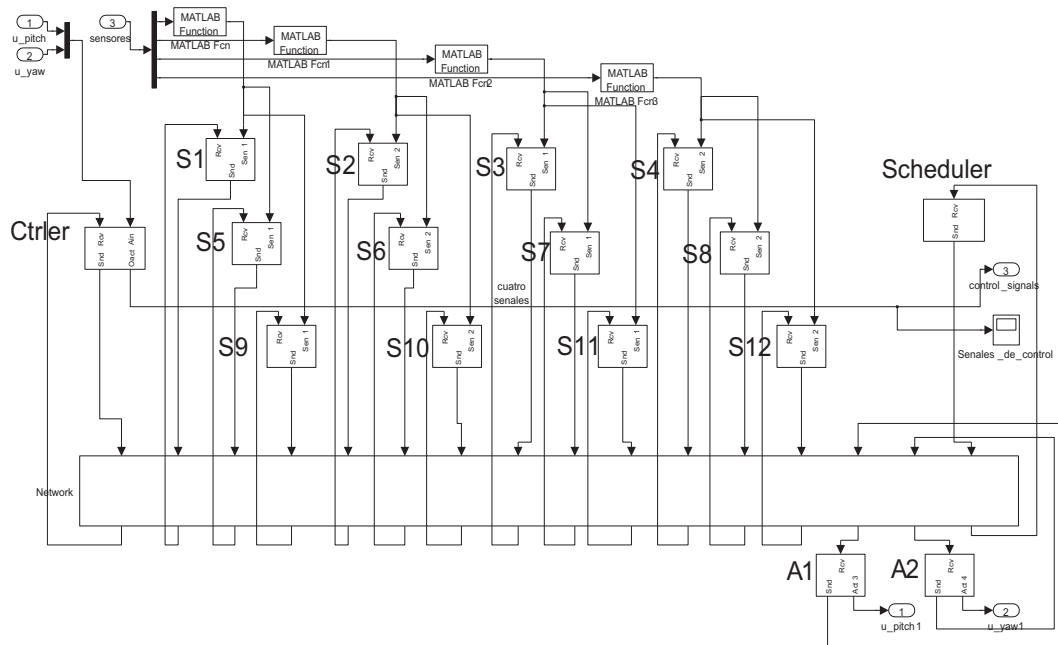


Figura 8: Modelo del sistema distribuido multiagente con redundancia de sensores

el acceso de la red cuando la demanda de comunicación entre los demás agentes aumenta, lo anterior con la finalidad de evitar sobrecarga en la red y los retardos que aparecen como consecuencia. Es de resaltar que la planificación se basa en el ámbito

del manejo de los procesos que realiza cada agente mismo que tiene un impacto en la reconfiguración y por consecuencia en el desempeño del sistema. El cumplimiento de las restricciones de utilización de cada procesador según la política de planifi-

cación empleada, en este caso *Rate Monotonic*, asegura la planificación local del sistema. Por otra parte, la modificación de periodos en cada bloque de sensado intenta evitar colisiones y saturación de la red para cada periodo de muestreo. Para esto se ha incluido un parámetro  $\rho$  que llamaremos *periodo base de muestreo* y un parámetro  $\lambda$  que llamaremos *dispersión* el cual desfasa el inicio de la ejecución de las tareas de los agentes sensores. El cambio de los periodos de las tareas que realizan los agentes del sistema distribuido se realiza por medio de la modificación de los parámetros  $\rho$ ,  $\lambda$ ; cada bloque de sensores tiene un periodo de sensado que depende de estos parámetros. Cada bloque de sensores realiza tareas periódicas con periodo de operación  $p_i$  y tiempo de cómputo  $c_i$  como se muestra en la tabla 2.

Tabla 2: Periodos de muestreo y tiempos de cómputo de las tareas de cada bloque de sensores

Bloque	$c_i$	$p_i$
1	0.001	$\rho(1+\lambda)$
2	0.001	$\rho(1-\lambda)$
3	0.001	$\rho(1-1.1\lambda)$
4	0.001	$\rho(1+1.1\lambda)$

Cabe hacer mención que cada combinación presentada en la tabla 2 corresponde a un cierto retardo de tiempo lo que conlleva a un diseño particular de la ley de control.

## 5. Simulaciones

La propuesta de reconfiguración se basa en modificar  $\rho$ ,  $\lambda$  el uso de enmascaramiento de falla y la disminución de la actividad de los sensores con falla. Para el caso de estudio se revisaron tres escenarios:

- La dinámica del sistema sin falla.
- Falla en la lectura de la señal de uno de los agentes que pertenecen a un bloque de sensado específico y no hay reconfiguración del sistema.
- Falla en la lectura de la señal de uno de los agentes del bloque y reconfiguración del sistema. Para este escenario los agentes se comunican entre sí y la actividad de la red aumenta.

La métrica utilizada para cuantificar la respuesta de sistema bajo los tres distintos escenarios es el cálculo de la integral del cuadrado del valor absoluto del error (IAE). Dicho error corresponde a la diferencia entre la referencia proporcionada y la señal de salida del sistema (Lian et. al., 2002):

$$IAE = \int_{t_0}^{t_f} |e(t)| dt \approx \sum_{k=k_0}^{k_f} |r(kh) - y(kh)|$$

donde  $r(k)h$  es la señal de referencia,  $y(kh)$  es la señal de salida,  $t_0(k_0)$  y  $t_f(k_f)$  son los tiempos inicial y final de los tiempos de simulación.

Algunos resultados preliminares de este sistema distribuido al variar los periodos de muestreo (Menéndez and Benítez-Pérez, 2010) han mostrado que el sistema no presenta saturación de la red ni bajo muestreo para valores de  $\rho$  entre 0.010 y 0.020 segundos, pero se presenta un cierto tráfico que conlleva a una disminución en el desempeño.

### 5.1. Escenarios con falla y sin falla

De las cuatro señales que se hace la lectura  $x = [\theta, \psi, \dot{\theta}, \dot{\psi}]'$  se analiza la respuesta de la señal del ángulo de cabeceo  $\theta$ . Se simula el vuelo del prototipo durante 30 segundos y se calcula el IAE para este tiempo. La lectura de los tres sensores del bloque  $B_1$  para un escenario sin falla es la misma para cada uno de ellos (figura 9) y se obtiene un IAE para los 30 segundos de simulación de 9.64 unidades, los agentes sensores no utilizan un ancho de banda excesivo en contraste con la actividad del agente controlador (figura 10).

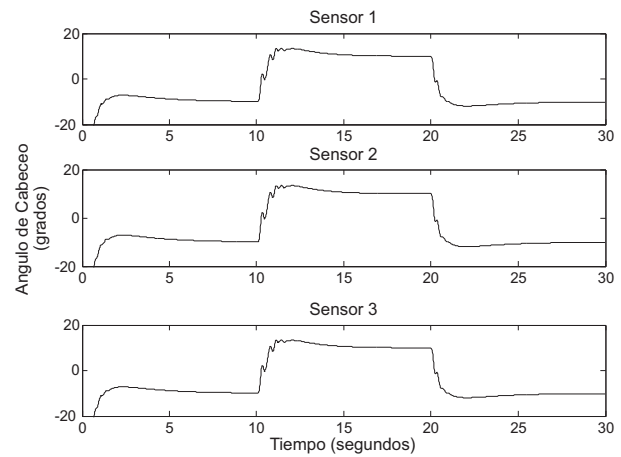


Figura 9: Lectura de los sensores de ángulo de cabeceo en escenario sin falla.

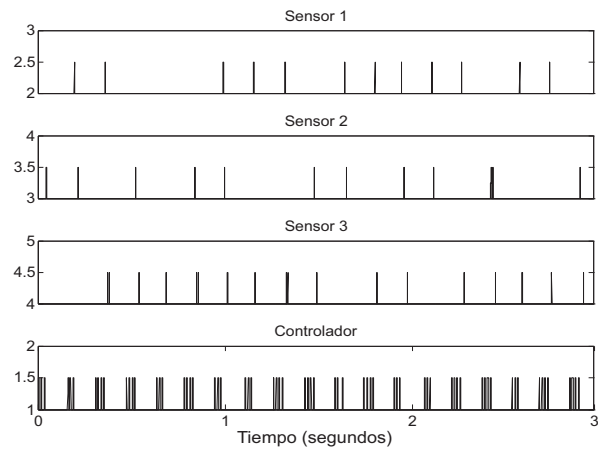


Figura 10: Transmisión de datos de los agentes del bloque y del agente controlador

Para comenzar a estudiar el impacto del esquema de reconfiguración planteado se inducen distintos tipos de falla en el sistema que afectan a uno de los agentes del bloque de sensado  $B_1$ . En particular, se induce un retraso (corrimiento) de tres segundos en la lectura de la señal del agente  $a_{13}$ , que inicia en el segundo 10 y sale en el segundo 23. Al aplicar la falla en el tercer agente se observa que la señal de control es afectada en los tres agentes y el desfase de la lectura en el agente que falla (figura 11). El valor de la IAE obtenido fue de 98.43 unidades.

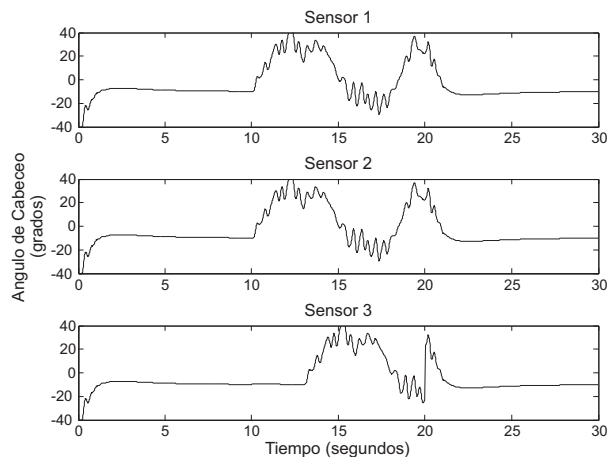


Figura 11: Lectura de los sensores del ángulo de cabeceo con falla de corrimiento inducida en un agente del bloque  $B_1$

En la tabla 3 se muestran los valores de la IAE obtenido cuando se hace variar  $\rho$  dentro del intervalo  $[0.010, 0.025]$  segundos y  $\lambda$  en el intervalo entre  $[0.0, 0.20]$ . Cuando el valor del periodo base está en el intervalo  $[0.010, 0.018]$  segundos el desempeño del sistema es estable respecto al IAE, no obstante al aumentar el periodo base, existe bajo muestreo, dando como resultado el aumento en el error (figura 12). En este sentido no existe reconfiguración solo el efecto de la modificación de los parámetros de planificación para un escenario en particular.

De la misma forma, en la tabla 4 se muestra el valor del IAE para distintos valores de  $\rho$  y  $\lambda$  dentro de los intervalos arriba mencionados cuando se presenta falla en el tercer agente del bloque  $B_1$  que toma la lectura de la señal  $\theta$ . En los casos mostrados en las tablas 3 y 4 los retardos asociados tienen un controlador correspondiente que hace al sistema estable bajo las condiciones de retardo más no del incumplimiento en el tiempo real como son los casos de alta frecuencia. En este sentido se muestra que el problema no es sólo de retardo sino de planificación por lo que es necesario tomar en cuenta esta restricción para lograr una buena reconfiguración la cual es analizada con base al IAE para todos los acuerdos. Para este caso (figura 13) no existe un acuerdo entre los agentes solo el cambio de planificación fuera de línea considerando un escenario con fallas.

### 5.2. Escenario con falla y reconfiguración

El sistema experimenta una falla de corrimiento en el tercer agente del bloque de sensado de la señal del ángulo de cabeceo

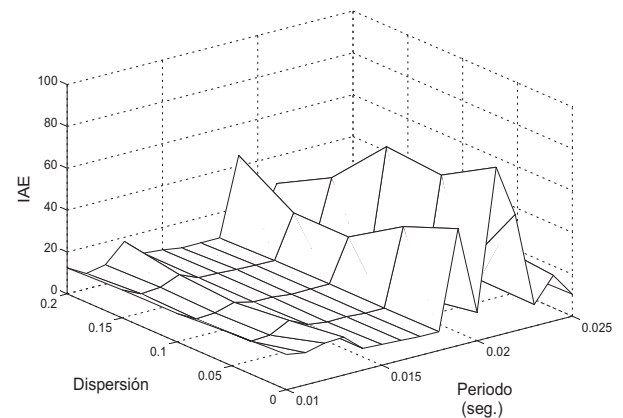


Figura 12: Relación del IAE para distintos valores del periodo base  $\rho$  y dispersión  $\lambda$  en un escenario sin falla

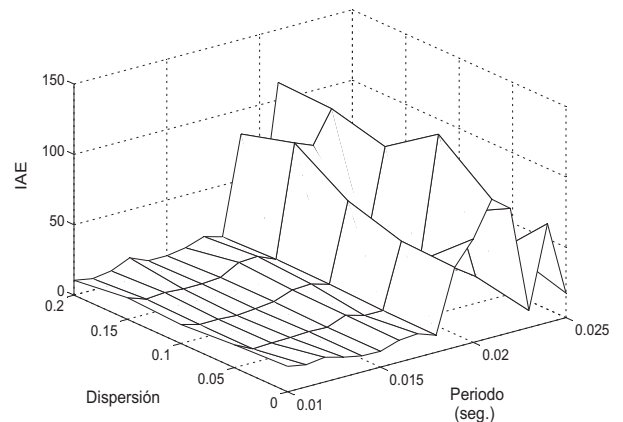


Figura 13: Relación del IAE respecto al periodo base  $\rho$  y dispersión  $\lambda$  en un escenario con falla

$\theta$  y es reconfigurado por medio de enmascaramiento, disminución de actividad o eliminación total de transmisión del agente que falla, previo acuerdo entre los agentes. En la reconfiguración por enmascaramiento el agente que falla intercambia las lecturas de la señal con la lectura local de los otros dos agentes de su bloque para alimentar la entrada de la unidad de voto y el valor devuelto es transmitido a la misma frecuencia que los otros dos agentes. Con este tipo de reconfiguración el sistema no logra estabilizarse (figura 14) y la actividad de la red aumenta por la transmisión y recepción de datos del tercer agente (figura 15). Lo anterior resulta en aumento del valor del IAE e inestabilidad del sistema.

En la figura 16 se muestra el valor del IAE al modificar  $\rho$  en el intervalo  $[0.010, 0.025]$  segundos y  $\lambda$  en el intervalo  $[0.0, 0.20]$  bajo un escenario con falla, acuerdo entre agentes y la reconfiguración mediante voto. En general, esta acción de reconfiguración que se acuerda no impacta positivamente bajo esta falla específica, el enmascaramiento de falla no influye lo suficiente para disminuir el efecto de la falla de corrimiento inducida mientras que la red experimenta una tasa de transmisión

Tabla 3: Valores de la IAE para distintos valores del periodo base  $\rho$  y dispersión  $\lambda$  en un escenario sin falla

Dispersión	Periodo base ( $10^{-3}$ segs.)															
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0.0	16.85	15.38	17.58	18.09	10.38	9.96	9.51	9.01	8.98	55.86	13.55	60.55	55.68	10.29	21.86	10.52
0.05	15.45	9.41	9.99	14.67	10.40	9.23	9.41	8.74	8.79	45.41	8.94	65.33	17.24	10.61	24.37	9.89
0.10	13.59	8.59	9.20	11.84	9.73	9.68	8.67	8.85	8.92	28.99	9.18	67.43	14.19	10.73	24.03	11.33
0.15	13.40	9.18	9.90	13.65	9.86	9.60	9.08	8.82	8.69	29.06	9.59	41.15	20.55	10.81	22.95	10.42
0.20	12.43	7.39	9.19	17.97	10.09	9.29	7.89	7.97	7.99	45.05	8.58	26.95	15.19	10.06	23.15	7.80

Tabla 4: Valores de la IAE para distintos valores del periodo base  $\rho$  y dispersión  $\lambda$  en un escenario con falla

Dispersión	Periodo base ( $10^{-3}$ segs.)															
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0.0	18.09	14.74	17.54	14.10	11.49	11.22	15.50	16.69	12.12	56.77	73.46	98.26	88.20	12.12	70.28	17.44
0.05	16.34	9.92	10.57	10.89	11.57	11.15	16.26	12.49	12.49	58.31	40.75	127.2	68.26	19.21	34.09	28.67
0.10	13.52	10.01	10.20	9.42	10.64	12.22	16.99	17.74	13.99	69.76	36.46	101.2	33.84	17.48	29.55	13.80
0.15	15.39	10.88	12.01	11.08	11.17	12.20	19.33	20.16	14.63	93.68	88.41	111.0	55.80	16.87	23.12	16.58
0.20	10.55	8.20	10.18	15.98	10.40	10.40	12.56	16.98	13.32	82.85	14.71	112.4	43.38	13.55	24.43	13.06

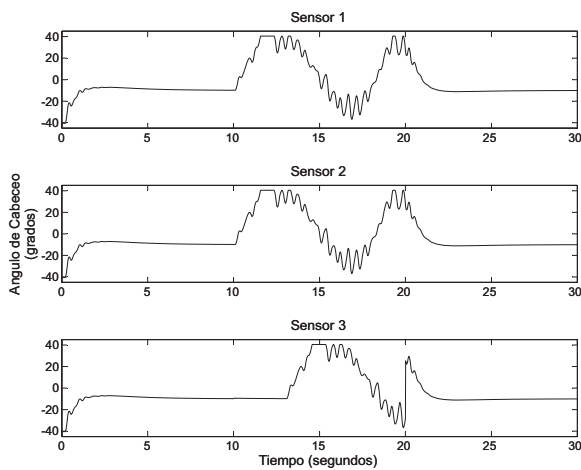


Figura 14: Lectura de los sensores del ángulo de cabeceo con falla de corrimiento y reconfiguración mediante voto

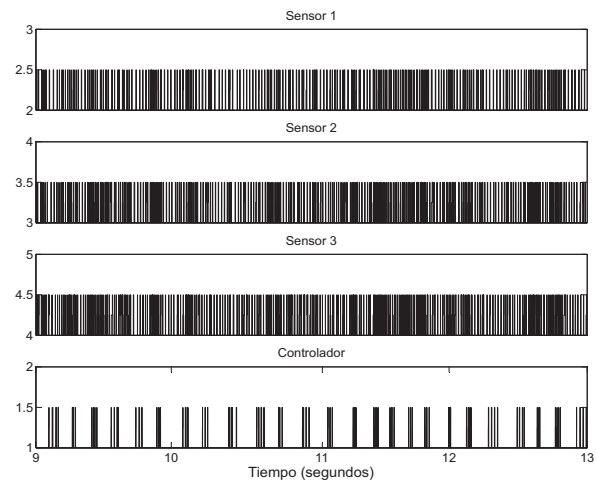


Figura 15: Transmisión de datos de los agentes del bloque de sensado y del agente controlador. La reconfiguración es por medio de voto

mayor.

El siguiente tipo de reconfiguración consistió en disminuir la tasa de transmisión de los datos del agente que falla. El agente que identifica la falla en la lectura acuerda disminuir su actividad, la tasa de transmisión disminuye a un tercio de su valor original y no modifica su periodo de muestreo (figura 17), el valor del IAE fue de 24.2 unidades. De manera similar, se implementó la suspensión total de la transmisión hacia el controlador del agente con falla, los agentes sin falla son los que transmiten asumiendo la pérdida de un agente del bloque (figura 18), bajo este escenario el valor del IAE fue de 9.3 unidades.

En ambos casos la transición debido al acuerdo entre agentes es irrelevante permitiendo un buen desempeño. Recuérdese que la transición se lleva a cabo considerando el diseño global del

controlador para ambos escenarios de retardos de tiempo exclusivamente sin considerar el tráfico de la red de cómputo.

Finalmente, se implementa una reconfiguración que modifica los periodos de muestreo y transmisión de datos de los elementos del sistema distribuido por medio del cambio de valores de  $\rho$  y  $\lambda$ . Al inicio de la ejecución los valores de  $\rho$  y  $\lambda$  se establecen en 0.010 segundos y 0.05 respectivamente, a partir del momento de inicio de la falla se modifica el periodo base a 0.017 segundos. El resultado de esta reconfiguración muestra que la inestabilidad del sistema provocado por la falla es corregida parcialmente (figura 19).

Se logra determinar una estructura de tiempo real aún cuando exista la reconfiguración dado el conocimiento de la respuesta de la planificación con base al acuerdo en línea entre agentes.



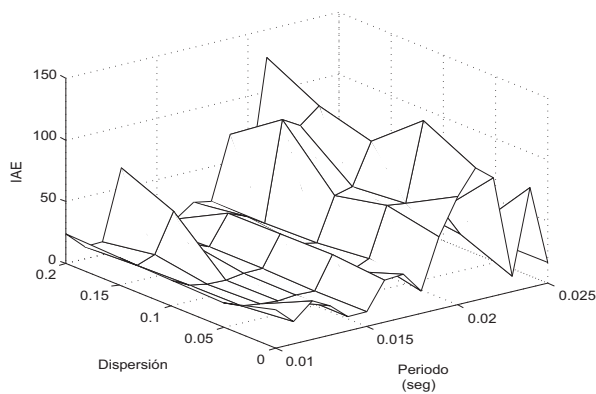


Figura 16: Relación del IAE respecto al periodo base  $\rho$  y dispersión  $\lambda$  en un escenario con falla

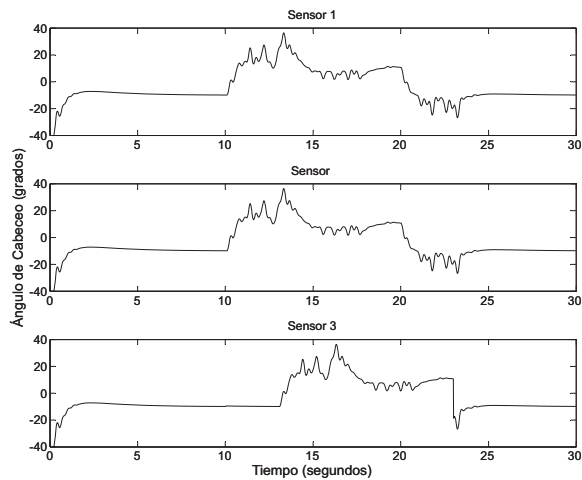


Figura 17: Lectura de los sensores del ángulo de cabeceo con falla de corrimiento y reconfiguración que disminuye la tasa de transmisión del agente con falla

Para cualquier evento fortuito que suceda en el ordenamiento se requiere un acuerdo entre los elementos independientes para evitar una situación catastrófica en el manejo de la red. Por ejemplo las regiones presentadas en las figuras 13 y 16 donde el IAE tiende a ser muy alto debido al alto uso de comunicación en la red de cómputo.

## 6. Conclusión

Se ha mostrado la evaluación de distintos tipos de reconfiguración de un sistema distribuido incluido en un ciclo de control de un prototipo de helicóptero de dos hélices. Las simulaciones muestran el impacto de una falla en el sensado de las señales de control. Por medio de redundancia de sensores se propone un esquema multiagente que acuerda un tipo de reconfiguración cuando el sistema presenta un escenario con falla. Dicha reconfiguración puede ser elegida entre el cambio de periodos de

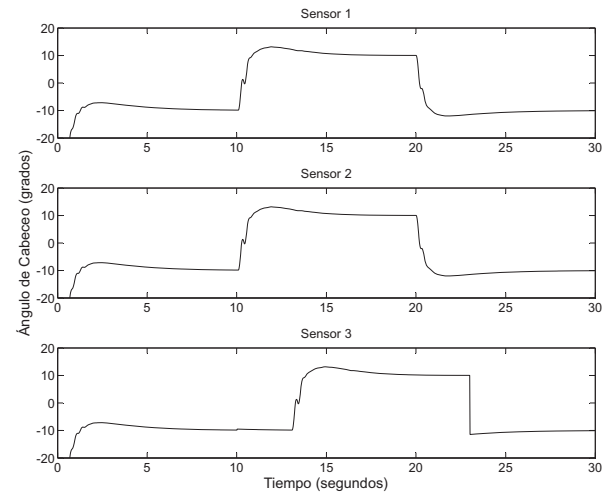


Figura 18: Lectura de los sensores del ángulo de cabeceo con falla de corrimiento y reconfiguración que elimina la transmisión de datos del agente con falla

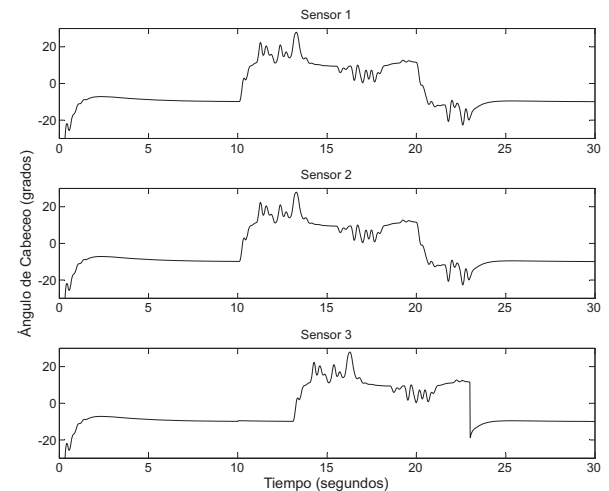


Figura 19: Lectura de los sensores del ángulo de cabeceo con falla de corrimiento y reconfiguración que modifica el periodo base de muestreo

muestreo, enmascaramiento o disminución de actividad del sensor que falla. La reconfiguración más efectiva cuando existe una falla de retraso en la lectura de los sensores es la disminución de la transmisión en el sensor con falla, de igual manera lo es la eliminación total de su actividad. Sin embargo, esto podría ser improcedente para fallas producidas en dos o más sensores dentro de un mismo módulo de sensado.

La reconfiguración por medio de enmascaramiento aumenta el tráfico de la red y bajo el escenario de falla planteado es poco útil, se mostró que para este caso aumenta el error cuadrático medio; no obstante, cabe señalar que el efecto de fallas en más agentes del mismo módulo de sensado podría disminuirse mediante voto.

La reconfiguración que cambie el periodo base de las tareas de los agentes del sistema distribuido disminuye el error

cuadrático medio, un trabajo futuro consiste en explorar la reconfiguración de dichos periodos de manera local y que impacte de manera global.

La aportación de este trabajo consiste en proponer y utilizar la redundancia de dispositivos inteligentes como base de una estructura multiagente que decida mediante un acuerdo el tipo de reconfiguración que impacte de manera global al sistema evitando así una reconfiguración centralizada.

## English Summary

### Dynamic Reconfiguration of Real-Time Distributed Systems Based on Agents

#### Abstract

Currently distributed systems are present in several applications in industry and research, such systems are restricted to time conditions. The presence of failures in these time critical systems requires decision making to avoid loss of deadlines, performance, instability or total suspension of the system. A strategy for achieving this is the dynamic reconfiguration of the structure of the system aimed at counteracting the effect of such failures. This article presents a proposal for dynamic and distributed reconfiguration considering each element of the distributed system as an agent with communication and cooperation capabilities to act joint with other agents to generate a reconfiguration action. These actions are the modification of sampling periods of sensors agents, decreased activity of the agents with fails and / or use of voting algorithms. For each type of system failure correspond a particular type of reconfiguration which is reached based on exchange of information between agents and the agreement they reached. The proposed dynamic reconfiguration scheme is implemented in the flight control of a prototype helicopter with two propellers through the inclusion in the closed loop control of a distributed system that uses a common communication media. The results of the reconfigured system response are obtained through numerical simulations carried out using Matlab and Simulink, as Real-Time simulation tool was used TrueTime.

#### Keywords:

Distributed Systems, Reconfiguration, Real-Time, Agents.

## Agradecimientos

Este trabajo ha sido realizado gracias al apoyo del Consejo Nacional de Ciencia y Tecnología (CONACYT), al Programa de Apoyo a los Proyectos de Investigación e Innovación Tecnológica PAPIIT IN 103310 y ICYTDF PICCO 10 53, así como al Programa de Posgrado en Ciencia e Ingeniería de la computación, U.N.A.M.

También deseamos expresar lo valioso que han resultado las opiniones de los revisores de la Revista Iberoamericana de Automática e Informática Industrial para consolidar este proyecto de investigación.

## Referencias

- Almeida, J., 2001. Dynamic reconfiguration of object-middleware-based distributed systems. Master's thesis, University of Twente.
- Benítez-Pérez, H., García-Nocetti, F., 2005. Reconfigurable Distributed Control. Springer Verlag.
- Benítez-Pérez, H., Ortega-Arjona, J., Cardenas-Flores, F., Quiñones-Reyes, P., Diciembre 2010. Reconfiguration control strategy using takagi-sugeno model predictive control for network systems - a magnetic levitation case study. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 224 (8), 1022–1032.
- Castillo, P., García, P., R., L., P., A., 2007. Modelado y estabilización de un helicóptero con cuatro rotores. Revista Iberoamericana de Automática e Informática Industrial 4 (1), 41–57.
- Canjor, A., García, A., 2005. Control basado en agentes mejorados con tecnología auto-id. Revista Iberoamericana de Automática e Informática Industrial 2 (3), 48–60.
- Cervin, A., Henriksson, D., Lincoln, B., Eker, J., Årzen, K.-E., Junio 2003. How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime. Control Systems, IEEE 23 (3), 16 – 30.
- Desforges, X., Archimède, B., 2006. Multi-agent framework based on smart sensors/actuators for machine tools control and monitoring. Engineering Applications of Artificial Intelligence 19 (6), 641 – 655.
- García, A., Canjor, A., 2007. Sistema heterárquico de control basado en agentes para sistemas de fabricación: La nueva tecnología proha. Revista Iberoamericana de Automática e Informática Industrial 4 (1), 83–94.
- García, C., Saltarén, R., López, J., Aracil, R., 2010. Desarrollo de una interfaz de usuario para el sistema robótico multiagente smart. Revista Iberoamericana de Automática e Informática Industrial 7 (4), 17–27.
- García-Sanz, M., Elso, J., Egaña, I., 2006. Control del ángulo de cabeceo de un helicóptero como benchmark de diseño de controladores. Revista Iberoamericana de Automática e Informática Industrial 3 (2), 111–116.
- Kramer, J., Magee, J., Abril 1985. Dynamic configuration for distributed systems. IEEE Transactions on Software Engineering 11 (4), 424–436.
- Latif-Shabgahi, G., Septiembre 2004. A novel algorithm for weighted average voting used in fault tolerant computing systems. Microprocessors and Microsystems 28 (7), 357–361.
- Latif-Shabgahi, G., Bennet, S., Bass, J., Agosto 2003. Smoothing voter: a novel voting algorithm for handling multiple errors in fault-tolerant control systems. Microprocessors and Microsystems 27 (7), 303–313.
- Lian, F., Moyne, J., Tilbury, D., Noviembre 2001. Time delay modeling and sample time selection for networked control systems. In: International Mechanical Engineering Congress and Exposition, 2001. Proceedings of ASME-DSC. Vol. XX. pp. 1–8.
- Lian, F., Moyne, J., Tilbury, D., Marzo 2002. Network design consideration for distributed control systems. Control Systems Technology, IEEE Transactions on 10 (2), 297–307.
- Liu, C., Layland, J., Enero 1973. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the Association for Computing Machinery 20 (1), 46–61.
- López-Martínez, M., Ortega, M., Vivas, C., Rubio, F., 2007. Control no lineal robusto de una maqueta de helicóptero con rotores de velocidad variable. Revista Iberoamericana de Automática e Informática Industrial 4 (3), 46–60.
- Menéndez, A., Benítez-Pérez, H., Agosto 2010. Scheduling strategy for real-time distributed systems. Journal of Applied Research and Technology 8 (2), 177–185.
- Méndez-Monroy, P. E., Benítez-Pérez, H., 2009. Supervisory fuzzy control for networked control systems for networked control systems. Innovative Computing, Information and Control, Express Letters 3 (2), 233–240.
- Ohlin, M., Henriksson, D., Cervin, A., 2007. Truetime 1.5 Reference Manual.
- Quanser, 2006. Quanser 2 DOF Helicopter. User and Control Manual.
- Romero, H., Salazar, S., Escareño, J., Lozano, R., 2010. Estabilización de un mini helicóptero de cuatro rotores basada en flujo óptico y sensores inerciales. Revista Iberoamericana de Automática e Informática Industrial 7 (2), 49–56.