

Procedimiento de diseño para minimizar el consumo de potencia y los retrasos en WSN

Diego Martínez *, Patricia Balbastre **, Francisco Blanes **, José Simo ** y Alfons Crespo **

* Universidad Autónoma de Occidente, Cali, Colombia
e-mail: dmartinez@uao.edu.co

** Universidad Politécnica de Valencia, Valencia, España
e-mail: {patricia.pblanes, jsimo, alfons}@disca.upv.es

Resumen: Actualmente existe un gran interés por el desarrollo de aplicaciones industriales utilizando redes inalámbricas, principalmente por el aumento de la flexibilidad del sistema y la disminución de los costos de implementación. Sin embargo, los retrasos y el jitter que introduce la red de comunicaciones en las aplicaciones de control, han dado lugar a que en algunos casos no se obtenga una buena correspondencia entre los resultados experimentales y los objetivos de control propuestos, esto como consecuencia del uso de modelos imprecisos para analizar y diseñar estos sistemas, métodos de validación poco elaborados y plataformas que no soportan los modelos empleados. En este trabajo se presenta un procedimiento de diseño que permite encontrar un modo de funcionamiento óptimo del sistema, que garantiza el cumplimiento de los plazos de tiempo de las aplicaciones, y minimiza el consumo de potencia y los retrasos. Copyright © 2010 CEA.

Palabras Clave: Sistemas de tiempo real, Redes de sensores y actuadores inalámbricas, Sistemas embebidos, Monitorización y control en tiempo real.

1. INTRODUCCIÓN

Como consecuencia del incremento en el tamaño y la complejidad de las aplicaciones de monitorización, diagnóstico y control, la mayoría de estas actividades han sido distribuidas entre diferentes nodos que se integran a través de redes de comunicaciones, bien sea porque la naturaleza del sistema así lo requiere, o porque de esta forma es posible aumentar la potencialidad del sistema a unos costos relativamente menores que los presentados por soluciones centralizadas. Adicionalmente se facilitan las actividades de diagnóstico, mantenimiento y trazabilidad en los procesos.

En el caso de aplicaciones de control, denominados sistemas de control en red (*Networked Control Systems* - NCS), los lazos de control se cierran a través de una red de comunicaciones, cuyo desempeño afecta los retrasos y el jitter presentes desde el instante en que se realiza la medición hasta que se actúa sobre el sistema. Esto ha generado que en algunos casos no se obtenga una buena correspondencia entre los resultados experimentales y los objetivos de control propuestos, lo cual es consecuencia del uso de modelos imprecisos para analizar y diseñar estos sistemas, métodos de validación poco elaborados y plataformas que no soportan los modelos empleados. El uso de redes inalámbricas en estas aplicaciones ha dado origen al término redes de sensores y actuadores inalámbricas (*Wireless Sensor and Actuators Networks* - WSN), con lo cual se contribuye a aumentar la flexibilidad de estas aplicaciones, además su impacto en la reducción de costos de implementación es significativo. Actualmente se realizan diversos trabajos relacionados con el análisis del desempeño de estos sistemas, la compensación de los retrasos, y el análisis y diseño de sistemas de cómputo para su adecuado funcionamiento.

Los parámetros más relevantes en el diseño de WSN son el

tiempo de vida, la escalabilidad, el tiempo de respuesta, la sincronización de los nodos, la protección de la información, y el cumplimiento de plazos de tiempo real, (Lester, 2003), (Martínez, 2009). Dependiendo del tipo de aplicación varía la importancia de estos objetivos en el diseño. Uno de los grandes desafíos en el desarrollo de estas aplicaciones se centra en que la minimización de uno de los anteriores factores puede ocasionar el aumento de otro, con lo cual se debe lograr un compromiso entre los mismos.

En este tipo de aplicaciones las funciones de medición, control, actuación y diagnóstico de fallos, se distribuyen entre los diferentes nodos. Algunas pueden tener una asignación preestablecida, como es el caso de las funciones de medición y actuación, debido a la conexión física de ciertos nodos con el sistema; mientras que las demás funciones se asignan a un nodo siguiendo criterios relacionados con el cumplimiento de las restricciones temporales, la optimización en el consumo de potencia, la minimización de los retrasos en el sistema, entre otras. En la figura 1 se presenta de manera general una aproximación a la estructura de estas aplicaciones.

Como consecuencia de las limitaciones de recursos en los nodos y las restricciones impuestas por las aplicaciones, es necesario el desarrollo de métodos que permitan la cooperación entre los diferentes niveles del sistema para alcanzar soluciones óptimas con respecto a algún criterio. El procedimiento de diseño propuesto en este trabajo contribuye en este sentido, este integra diversos niveles de la arquitectura de los nodos, con el propósito de encontrar soluciones óptimas que garanticen el cumplimiento de restricciones temporales, y un mínimo valor de consumo de potencia y retrasos en el sistema.

El documento está organizado de la siguiente forma, en la sección 2 se presentan los trabajos relacionados, una propuesta para la arquitectura de los nodos y el análisis de planificabilidad

en WSN se presenta en la sección 3, en la sección 4 se presenta el procedimiento de diseño propuesto, los resultados de simulación de un caso de estudio son presentados en la sección 5, finalmente en 6 se presentan las conclusiones y el trabajo futuro.

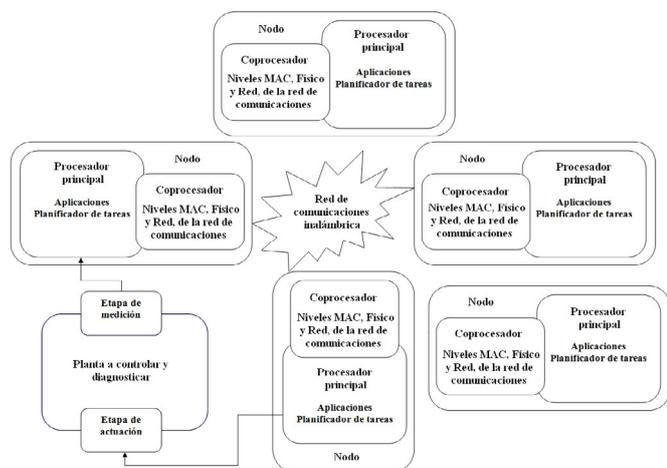


Figura 1. Esquema general del sistema de diagnóstico y control.

2. TRABAJOS RELACIONADOS

Este trabajo integra diferentes áreas como protocolos de redes de comunicaciones inalámbricas, análisis de tiempo real en la ejecución de tareas y en el envío de mensajes, escalado dinámico de voltaje, sistemas de control en red y metodologías de diseño de redes de sensores inalámbricas. En esta sección se analizan los trabajos relacionados.

2.1 Protocolos de redes de comunicaciones inalámbricas

En (Hristu-Varsakelis, 2005) y (Lee, 2008) se realiza un análisis de la utilización de Bluetooth e IEEE 802.11b como redes para control, al mismo tiempo que se presenta una arquitectura de cuatro niveles para lograr un comportamiento predecible en IEEE 802.11b, la desventaja que presentan estos protocolos respecto a otros como IEEE 802.15.4, es que su elevado consumo de potencia disminuye drásticamente el tiempo de vida de la aplicación. En (Pantazis, 2007) se presentan diferentes métodos para disminuir el consumo de potencia en diferentes niveles de la pila de comunicaciones de redes de sensores inalámbricas, entre los cuales se analiza el modo de acceso al medio TDMA (*Time Division Multiple Access*), el cual contribuye a garantizar tiempos de envío de mensajes en la red, minimizar colisiones y al ahorro de energía en los nodos.

Los resultados presentados en (Moraes, 2008) y (Cena, 2008) permiten concluir que el modo EDCA (*Enhanced Distributed Channel Access*) del 802.11e ofrece una buena alternativa para los requisitos de tiempo real en las aplicaciones industriales. Se concluye que para los escenarios presentados, el uso de los valores por defecto de EDCA permite garantizar los requisitos en el envío de información para aplicaciones industriales, donde el número de estaciones sea menor que 10 y con periodos entre mensajes por encima de 10 ms.

Actualmente existen diversos dispositivos comerciales que utilizan Zigbee (Zigbee Specification) y WirelessHART (WirelessHART), los cuales presentan un menor consumo de potencia que los desarrollos soportados en IEEE 802.11. Zigbee está soportado en el estándar IEEE 802.15.4-2003, el cual

implementa dos tipos básicos de acceso al medio, sin sincronización y con sincronización a través de *beacons*. La ventaja del modo sin sincronización es que facilita la escalabilidad y la autoconfiguración de la red, aunque no ofrece garantía en cuanto al tiempo de envío de los mensajes.

Empleando el modo de acceso con sincronización de IEEE 802.15.4, es posible garantizar tiempos máximos de envío de información haciendo uso de ranuras de tiempo garantizado (*Guaranteed Time Slot - GTS*) dentro de una *superframe*, lo cual es muy importante para el desarrollo de aplicaciones de control (Martínez, 2008). Adicionalmente existen primitivas por medio de las cuales el coordinador puede modificar el itinerario de envío de información en los GTS's. Solamente es posible asignar un máximo de siete ranuras, con un periodo mínimo de trama de 15,36 ms, lo cual puede ser suficiente en algunos casos; en (Koubâa, 2006) se presenta un método para la asignación de los GTS's. Sin embargo, este modo de operación está restringido a comunicaciones en redes estrella lo cual limita la confiabilidad y escalabilidad de la red.

Aunque la arquitectura de redes en estrella permite el uso de GTS's, el desarrollo de aplicaciones de control en ambiente industriales demanda una comunicación fiable y segura, lo cual es más fácil de suministrar con una arquitectura de red tipo *mesh*, la cual también contribuye a facilitar la escalabilidad de las aplicaciones. En cuanto a las arquitecturas de redes *mesh*, Zigbee utiliza un modo de acceso sin sincronización, aunque se han desarrollado propuestas para construir redes cluster-tree sincronizadas como la presentada en (Koubâa, 2007).

WirelessHART se basa en la capa física del estándar IEEE 802.15.4-2006, pero especifica nuevos niveles de Enlace de datos, Red, Transporte y Aplicación, (Lennvall, 2008). WirelessHART emplea un modo de acceso al medio soportado en TDMA, con 100 ranuras por segundo, con lo cual se minimizan las colisiones y se pueden emplear estrategias para disminuir el consumo de potencia. Adicionalmente, WirelessHART forma redes con topologías *mesh* suministrando caminos de redundancia los cuales permiten encaminar los mensajes por diferentes rutas, con lo cual se puede bordear obstáculos físicos y evitar enlaces interrumpidos e interferencias.

2.2 Algoritmos de planificación de tareas y mensajes

Debido al efecto que generan los retrasos sobre el desempeño de NCS, estas aplicaciones tienen restricciones de tiempo real con plazos de respuesta *end-to-end*. Este tiempo es calculado desde el instante en que se realiza la medición hasta cuando se actúa sobre el sistema, y depende de los tiempos de finalización de las tareas y de envío de los mensajes.

En su formulación general, el problema de evaluar la planificabilidad de un sistema distribuido de tiempo real es *NP-hard*. Con el fin de superar esta dificultad se deben emplear restricciones y heurísticas. Un enfoque común es asignar las tareas estáticamente a los nodos que conforman el sistema y localmente utilizar un algoritmo de planificación como RM (*Rate Monotonic*) o EDF (*Earliest Deadline First*), (Spuri, 1996).

Las aplicaciones en sistemas distribuidos están caracterizadas por poseer relaciones de precedencia entre sus tareas. Si las tareas son estáticamente asignadas a procesadores

independientes, las restricciones de tiempo *end-to-end* pueden ser analizadas por una teoría que considere la relación entre los jitter (Audsley, 1993).

En la literatura aparecen varios trabajos orientados al análisis de planificabilidad *end-to-end*, en los cuales se han empleado algoritmos de planificación en los nodos tales como RM y EDF, y protocolos de acceso al medio basados en TDMA, Token y Prioridades, (Spuri, 1996), (Tindell, 1994), (Tindell, 1995). Adicionalmente, en estos trabajos se considera el uso de buffers para almacenar mensajes dentro de los nodos, y por tanto emplean métodos de planificación para ordenar el orden de salida de los mensajes desde el buffer.

En lo que respecta a planificadores de tareas en los nodos, en el contexto de la redes de sensores uno de los sistemas operativos mas empleado ha sido el TinyOs (TinyOS), el cual está orientado para uso en sistemas de cómputo con grandes restricciones, tales como microcontroladores de 8 bits con pequeña cantidad de memoria; se basa en un modelo de programación basado en componentes y guiado por eventos, en el cual los manejadores de eventos tienen mayor prioridad que las tareas, las cuales se ejecutan basadas en un planificador FCFS (*First-Come First-Served*). La principal motivación para utilizar planificadores guiados por eventos en este tipo de aplicaciones, es porque aquellas plataformas donde se han implementado están orientadas a abordar sistemas reactivos, con lo cual el uso de estos esquemas de planificación posibilitan el envío del procesador a modos de bajo consumo de potencia por instantes de tiempo más prolongados, sin embargo este tipo de planificadores no son adecuados para garantizar plazos de tiempo real, lo cual es importante en aplicaciones de control. En la serie de productos Zigbee de Chipcon y Texas Instruments se emplea un planificador de tareas basado en prioridades estáticas, y mecanismos de comunicación soportados en paso de mensajes.

Aunque la planificación por prioridades fijas es la técnica de planificación en línea más popular en sistemas de tiempo real, el uso del algoritmo EDF está tomando cada vez más atención en ambientes industriales, debido a sus beneficios en cuanto al uso de los recursos del sistema. Actualmente EDF está disponible en lenguajes de tiempo real como RTSJ. También está disponible en sistemas operativos de tiempo real como SHARK y ERIKA.

2.3 Escalado dinámico de voltaje

Además de garantizar el cumplimiento de los plazos de las tareas, la propuesta presentada en este trabajo busca disminuir el consumo de potencia, para lo cual se ha considerado el empleo de técnicas de escalado dinámico de voltaje (*Dinamic Voltage Scaling* - DVS). Estas técnicas intentan resolver el compromiso entre el cómputo requerido por la aplicación y el tiempo de vida de la misma, este último relacionado directamente con la duración de la batería.

Teniendo en cuenta que en la mayoría de los casos, el cómputo máximo necesario demandado por un conjunto de tareas es mucho mayor que el promedio que debe ser mantenido, y que para los procesadores diseñados con tecnología CMOS, que en la actualidad son la mayoría, el voltaje de funcionamiento depende de la frecuencia de conmutación, de modo que cuando se utiliza un nivel reducido de frecuencias el procesador puede funcionar con un voltaje de alimentación inferior; es posible emplear un nivel bajo de voltaje en los intervalos cuando el cómputo

máximo no es necesario. De esta manera, como la energía disipada por ciclo con circuitos CMOS se escala de forma cuadrática con el voltaje de alimentación, las técnicas de DVS pueden proporcionar un gran ahorro de energía escalando la frecuencia y el voltaje (Pillai, 2001).

En la literatura aparecen diversos algoritmos para realizar DVS garantizando el cumplimiento de restricciones de tiempo real. En (Mejia-Alvarez, 2002) se presenta una metodología basada en heurísticas para realizar DVS, la cual requiere un bajo tiempo de cómputo. En (Saewong, 2003) se muestra un método para encontrar la frecuencia de funcionamiento óptima que permite alcanzar el mínimo consumo de potencia, sin embargo éste método es muy complejo y por tanto no es adecuado su uso en línea. En (Marinoni, 2007) se propone un método que combina DVS con una estrategia de planificación de tareas basada en un modelo elástico de las mismas; para la cual, en función de diversos objetivos de funcionamiento, se altera el periodo de las tareas del sistema. En (Zhu, 2004) utilizan un feedback control scheduling para realizar el DVS del procesador y un planificador EDF para administrar el orden de ejecución de las tareas.

En (Pillai, 2001) se analizó el desempeño del algoritmo de escalado estático de voltaje, en el que se selecciona la frecuencia de operación más baja que permite al planificador de tareas alcanzar todos los plazos impuestos en el sistema. Empleando esta estrategia, la frecuencia de funcionamiento seleccionada es puesta de forma estática y no es alterada a menos que cambie el conjunto de tareas. Las ventajas que presenta este método de escalado es su fácil implementación, la carga que genera al sistema es muy baja y cumple en cada ciclo con los plazos de tiempo del conjunto de tareas. Sin embargo, al realizar los análisis siempre con el tiempo de cómputo del peor caso de ejecución de las tareas, es muy pesimista por cuanto no considera el cómputo requerido por las tareas en cada ciclo de ejecución y de esta forma no alcanza el mayor ahorro de energía del sistema. En este trabajo también se muestra como el planificador EDF permite alcanzar frecuencias de funcionamiento más bajas, cumpliendo con los plazos de tiempo del conjunto de tareas, que el alcanzado con RM.

2.4 Sistemas de control en red

Existen diversos autores que han analizado el desempeño y la estabilidad en NCS asumiendo protocolos de red con retardos constantes y variables, de igual forma se han realizado propuestas para alterar los algoritmos de control con el fin de contrarrestar dichos efectos, en (Salt, 2008) se presenta una revisión de los trabajos realizados en este área.

Si los retardos de red son constantes, se pueden emplear técnicas de análisis estabilidad para sistemas de tiempo discreto con retardo (Astrom 1997), y técnicas de diseño de controladores soportadas en el predictor de Smith o controladores clásicos. Si los retardos de red son aleatorios, estos pueden ser modelados como una distribución probabilística constante o como una cadena de Markov, y se propone el uso algoritmos de control LQG, LQR, control óptimo, entre otros.

En (Walsh, 2002) se considera un modelos de NCS donde la planta y el controlador son continuos, se desprecian los retardos de propagación y de procesamiento, y el medio de comunicación es libre de errores; con lo cual se propone un nuevo algoritmo dinámico para el acceso a la red y funciones para garantizar la

estabilidad asintótica global de un NCS. En (Zhang, 2001) consideran una planta continua y el controlador discreto, modelan un sistema de control en red con pérdida de datos, y determinan la estabilidad del NCS con altas frecuencias de pérdida de datos. En (Branicky, 2002) se considera una planta continua y un controlador discreto, analizan la relación entre el periodo de muestreo y el retardo de la red para establecer la estabilidad del sistema de control en red, y aplican un algoritmo de planificación RM para el acceso a la red. Otros trabajos sobre estabilidad de NCS se presentan en (Gregory, 1999), (Yang, 2006), (Hespanha, 2007), (Tabbara, 2007), (Hu, 2008), (Huang, 2008), (Xiong, 2009).

Sin embargo no existe una técnica de análisis de estabilidad genérica que pueda ser aplicada a cualquier NCS. Cada técnica está sujeta a la configuración de la red, el protocolo de comunicaciones empleado, las suposiciones realizadas para el sistema y el algoritmo de control utilizado.

Por otro lado, las metodologías de diseño de controladores empleadas se han enfocado de diferentes maneras, diseñando el algoritmo de control con métodos tradicionales y trasladando el problema de los retrasos al diseño del sistema de cómputo, considerando las características del sistema de cómputo y diseñando un algoritmo de control que compense su efecto, y empleando un enfoque en donde el sistema de cómputo y el algoritmo de control son co-diseñados. En la mayoría de los trabajos realizados se analiza el desempeño del sistema considerando únicamente la estabilidad del lazo cerrado.

En la actualidad existe un gran interés por el uso de algoritmos de control basados en eventos en NCS, los cuales contribuyen a disminuir la utilización de los procesadores y de la red. A pesar de su buen desempeño en sistemas con capacidades reducidas de computación y comunicación, la teoría de análisis y diseño de estos sistemas requiere un mayor desarrollo. En (Dormido, 2008) se presenta una revisión de los trabajos realizados respecto a sistemas de control basados en eventos.

2.5 Metodologías de diseño de redes de sensores inalámbricas

En (Meshkova, 2008) se propone una metodología de diseño de WSN (Wireless Sensor Networks) orientada al servicio, pero no se presentan herramientas para automatizar de manera parcial o totalmente el procedimiento propuesto. Esta metodología se fundamenta en la interacción entre componentes para proporcionar un servicio con una determinada calidad. Las etapas de la metodología de diseño son: recopilación de los requisitos, análisis, diseño de la solución, desarrollo de la arquitectura de software, desarrollo del código, implementación y pruebas.

Los principales parámetros que han identificado para la descripción estas aplicaciones son el costo, el tiempo de vida, el retraso, la tolerancia a fallos y los servicios. Para la descripción del nivel de red se incluye información sobre la movilidad esperada de los nodos, la tasa de fallos, el ancho de banda mínimo requerido, el número de nodos en la red, el diámetro promedio de la red, la simetría de la red y la heterogeneidad. Los parámetros de los servicios incluyen una lista de las funcionalidades previstas y los servicios necesarios, para lo cual cada servicio se caracteriza por la calidad de servicio esperada. Respecto a la especificación del hardware y software se propone la especificación de la plataforma de hardware y del sensor, el

sistema operativo requerido, la cantidad de memoria necesaria, y una lista de los módulos requeridos para el funcionamiento del software.

En (Bonivento, 2005), (Bonivento, 2006a, b) se presenta un procedimiento de diseño de WSN para aplicaciones industriales, el cual comienza con una descripción de alto nivel del algoritmo de control y un conjunto de posibles plataformas de hardware para la síntesis del mismo, para posteriormente generar de manera automática una aplicación que satisface los requisitos del sistema y optimiza el consumo de energía. Para gestionar la heterogeneidad y complejidad, se identifican tres capas de abstracción y se introducen las herramientas para la transición entre las diferentes capas y obtener la solución final.

Este enfoque de diseño a nivel de sistema, se caracteriza por una fase *top-down* donde los requisitos de la aplicación son detallados en las necesidades *end-to-end* de la red, una fase inferior donde se abstrae el rendimiento del hardware, y una fase media donde se encuentran las dos fases anteriores, en la cual se utilizan los requerimientos y el rendimiento para resolver un problema de optimización con restricciones, cuya solución determina los protocolos de acceso al medio y encaminamiento de la red.

Algunas restricciones del modelo de componentes a nivel de sistema limitan la solución propuesta, por ejemplo no se contempla la comunicación entre sensores, lo que es utilizado en aplicaciones de fusión sensorial. Además, en este procedimiento se separan el diseño de la red y el diseño del sistema embebido, lo que limita la optimización del sistema en parámetros como consumo de potencia, latencias, jitter, entre otros.

El protocolo presentado para el acceso al medio implementa dos niveles. En el nivel superior, para la comunicación entre grupos de sensores, utiliza TDMA como estrategia para ahorrar energía y disminuir las colisiones de datos; mientras que en el nivel inferior utiliza un acceso por contienda, el cual utilizan los nodos para intercambiar información al interior de cada grupo, y que no permite garantizar un tiempo máximo de envío de información. Además, para encontrar la mejor ruta el algoritmo de encaminamiento emplea una estrategia para hallar el camino mas corto, lo cual no es siempre la mejor opción en términos de ahorro de energía y disminución de retrasos en la comunicación.

En (Prasad, 2007) se presenta un método de análisis y diseño de WSN soportado en la herramienta ANDES. Este consiste en un análisis previo a la implementación del sistema, el cual se soporta en técnicas teóricas de análisis para estimar los parámetros mas importantes del rendimiento del sistema, como el tiempo de vida, la cobertura y la fiabilidad, y realiza un análisis intuitivo del cumplimiento de restricciones de tiempo real en la red soportado en la capacidad de la red, la cual relaciona la distancia que deben recorrer los mensajes con su plazo relativo. Estos análisis se soportan en un conjunto de parámetros del sistema, como el número de nodos, el rango de detección de los nodos, y el ancho de banda disponible. Durante el proceso de diseño, estas técnicas de análisis se pueden aplicar de forma iterativa para ajustar parámetros del sistema en función del rendimiento deseado y el rendimiento estimado por el análisis. Sin embargo la herramienta no cubre todas las fases del ciclo de diseño de estas aplicaciones.

3. ARQUITECTURA DE LOS NODOS Y ANÁLISIS DE PLANIFICABILIDAD

El diseño de estos sistemas está caracterizado por restricciones propias de la aplicación. En (Martínez, 2009) se presenta una clasificación de las aplicaciones en los sectores médico, industrial, medioambiental y agrícola. Con base en este estudio se propone una arquitectura para los nodos del sistema, la cual se presenta en la figura 2. Esta arquitectura se orienta a aplicaciones que demandan un ancho de banda inferior a 250 kbps, poseen un bajo requerimiento de cómputo y funcionan en áreas de pequeña extensión; además, posibilita el cumplimiento de restricciones de tiempo real *end-to-end* y facilita la implementación de estrategias de ahorro de energía.

Figura 2. Arquitectura lógica de los nodos

Los niveles que conforman la arquitectura de los nodos han sido seleccionados considerando el análisis de las diferentes temáticas presentado en el apartado 2. La función de los diferentes niveles de la arquitectura del nodo son:

- Tareas: Realizan las actividades relacionadas con la aplicación.
- Middleware: Recibe las solicitudes de las tareas a partir de lo cual realiza la configuración del modo de funcionamiento para un determinado escenario, entendido este como un conjunto de tareas con requisitos de cómputo y restricciones de tiempo conocidos. Lo cual realiza enviando órdenes al sistema operativo para que, con base en el algoritmo de escalado estático de voltaje presentado en (Pillai, 2001), configure el procesador (frecuencia y voltaje de funcionamiento) y active las tareas que ha de ejecutar el nodo durante dicho modo de funcionamiento.
- Núcleo del sistema operativo: Es un núcleo de soporte mínimo, que realiza la ejecución de las tareas con base en una política de planificación EDF, el cual es adecuado para abordar sistemas de tiempo real.
- Hardware. Está compuesto de dos procesadores, el procesador principal y el coprocesador. El procesador principal ejecuta el software y permite el uso de estrategias de DVS para disminuir el consumo de potencia, para lo cual es posible utilizar procesadores con arquitecturas como la del XScale (Varma, 2005).

El Coprocesador implementa los niveles Físico, Enlace y Red, de la red de comunicaciones. De tal forma que estas actividades no afecten el desempeño de las funciones de procesamiento, garantizando el nivel de calidad de la aplicación y la sincronización de los nodos de la red. En relación al acceso al medio se emplea un algoritmo TDMA, el cual es importante para cumplir con los plazos de tiempo en el sistema e implementar estrategias de ahorro de energía en la comunicación.

En este trabajo no se consideró el almacenamiento de mensajes en buffers, por cuanto el tamaño de los mensajes enviados en aplicaciones industriales es pequeño comparado con la cantidad de datos que es posible enviar en cada mensaje de los actuales protocolos estándar (máximo *payload* en la PDU del nivel físico de 127 bytes para Zigbee y WirelessHART). Luego esta propuesta se diferencia de las anteriores en que se supone que todos los nodos se encuentran conectados a la misma red y que cada mensaje se envía dentro de la ranura asignada a cada nodo en la red TDMA. Por tanto se considera que el retraso introducido por la red es constante e igual al máximo posible, que es el periodo con el que se repite el itinerario de ranuras en la red TDMA.

3.1 Notación y modelo del sistema

Considerando un esquema general en donde las tareas que realizan las funciones dentro del NCS, *Medición, Control y Actuación*, se ejecutan en dicha secuencia, en nodos independientes y en exclusión mutua (no se puede iniciar la ejecución de la siguiente función si la anterior no ha finalizado), se definen:

- T_{SF} , es el periodo con el que se repite el itinerario de ranuras en la red TDMA.
- D_{CGR} , es el plazo *end-to-end*, entre la medición y la actuación, impuesto por los objetivos de desempeño del sistema.
- T_S , es el periodo de muestreo empleado en la medición, el cual se define de acuerdo con la respuesta dinámica del sistema a controlar, y ha de cumplir con $D_{CGR} \leq T_S$.
- $\tau = \{Task_1, Task_2, \dots, Task_n\}$, es un sistema planificable bajo EDF, con $Task_i = (WCET_i, D_i, T_i)$; donde $WCET_i$, D_i y T_i , son los respectivos valores de tiempo de cómputo del peor caso, plazo y periodo de la tarea $Task_i$.
- $WCRT_i$, es el tiempo de respuesta del peor caso de la tarea $Task_i$.

3.2 Análisis de planificabilidad

Respecto al análisis de planificabilidad para EDF, en (Ripoll, 1996) se define:

- $H_\tau(t) = \sum_{i=1}^n WCET_i \left\lceil \frac{t + T_i - D_i}{T_i} \right\rceil$, es la cantidad de tiempo de cómputo que tiene que haber sido servida

por el procesador hasta el tiempo t para cumplir con los plazos de tiempo del sistema.

- Intervalo crítico inicial (ICI), es el intervalo de tiempo entre cero y el primer instante en el que no hay activaciones pendientes $[0, R)$.

El test de planificabilidad en un planificador EDF consiste en verificar que:

$$H_r(t) < t \quad \forall t \leq R \quad (1)$$

Luego el análisis de planificabilidad *end-to-end* para el sistema de control se garantiza cuando D_M , D_C y D_A cumplen con las expresiones presentadas en la tabla 1, donde D_M , D_C y D_A son los plazos de las tareas de *Medición*, *Control* y *Actuación* respectivamente.

Para la implementación del sistema es posible asignar a D_M , D_C y D_A un valor entre $[D_j^{\min}, D_j^{\max}]$, siendo D_j^{\min} y D_j^{\max} los valores mínimo y máximo de D_j que permiten alcanzar la planificabilidad local y *end-to-end* respectivamente, en función de su impacto en parámetros como el consumo de potencia en cada nodo, los retrasos y el jitter en el sistema de control, entre otros. Donde D^{\min} se obtiene a partir del algoritmo *Deadlinemin* presentado en (Balbastre, 2008).

Dependiendo de las características físicas del sistema, las tareas de *Medición* y *Actuación* pueden estar en el mismo nodo; de igual forma, la tarea *Control* se podrá ejecutar en cualquier nodo con el propósito de equilibrar el porcentaje de utilización entre los mismos, lo cual incide sobre el tiempo de vida en aplicaciones donde los nodos dependen de baterías. Por lo anterior, se pueden presentar tres variantes a la arquitectura general del sistema de control, para las cuales los valores presentados en la tabla 1 se deben modificar de la siguiente manera:

- *Medición* y *Control* comparten el mismo nodo. Para este caso $WCRT_M' = WCRT_M$.
- *Control* y *Actuación* comparten el mismo nodo. Para este caso $WCRT_C' = WCRT_C$.
- *Medición* y *Actuación* comparten el mismo nodo. Para este caso no se presenta ningún cambio en la formulación inicial, lo cual es consecuencia de que la red es utilizada para el envío de información entre *Medición* y *Control*, y entre *Control* y *Actuación*.

Para el cálculo de los tiempos de cómputo de las tareas para las diferentes frecuencias de operación del procesador, se considera

una arquitectura de procesador en la que al escalar la frecuencia de operación del procesador también se escalan los ciclos de lectura y escritura en memoria, logrando de esta forma simplificar el problema del escalado del tiempo de cómputo de las tareas.

Con el propósito de disminuir al mínimo el consumo de potencia, mientras no haya mas activaciones pendientes el procesador se llevará al estado de menor consumo que no implique el reinicio del sistema, y del cual salga al presentarse una interrupción.

4. PROCEDIMIENTO DE DISEÑO DE WSN

El procedimiento de diseño propuesto se presenta en la figura 3. Las consideraciones generales del procedimiento son:

- El número y ubicación de los nodos de la red es conocido.
- Debido al elevado tiempo de cómputo requerido para evaluar la planificabilidad y la configuración del sistema, estos análisis se realizan fuera de línea. En este orden de ideas, todos los escenarios de la aplicación y la configuración óptima del sistema para cada uno (modos de funcionamiento) se suponen conocidos, y se almacenan en la memoria de cada nodo para ser utilizados durante el funcionamiento del sistema en el momento en que se genere cada uno de los escenarios. Los diferentes escenarios se generan a partir de cambios de modo de las tareas, creación y destrucción de tareas, y cambios en el número y ubicación de los nodos.
- Se desprecia la latencia entre los cambios de modo de funcionamiento.
- La elección del mejor modo de funcionamiento del sistema para cada uno de los escenarios se obtiene a partir de un proceso de optimización.
- Para la activación de tareas en cada escenario de la aplicación, se supone que existe una réplica de cada una de las tareas que no tienen una asignación preestablecida en cada nodo en los que pueden ser ejecutadas en función del escenario del sistema. Esta consideración se puede establecer debido a que la cantidad de tareas en cada nodo y el tamaño de código de las mismas no es elevado y puede ser almacenado en los nodos actuales.
- Todos los nodos se encuentran dentro del mismo segmento de red y existe un enlace directo entre cada uno, y por lo tanto la comunicación entre dos nodos no tarda mas de un T_{SF} . Las tramas son transmitidas y recibidas libres de error.

Tabla 1. Parámetros de las tareas para el análisis de planificabilidad *end-to-end*.

Tarea <i>Medición</i>	Tarea <i>Control</i>	Tarea <i>Actuación</i>
Periódica, $T_M = T_S$	Esporádica, $T_{C_{Min}} = T_S$	Esporádica, $T_{A_{Min}} = T_S$
$WCET_M \leq WCRT_M \leq D_M$	$WCET_C \leq WCRT_C \leq D_C$	$WCET_A \leq WCRT_A \leq D_A$
$D_M^{\max} = D_{CGR} - (WCRT_C' + WCRT_A + T_{SF})$	$D_C^{\max} = D_{CGR} - (WCRT_M' + WCRT_A + T_{SF})$	$D_A^{\max} = D_{CGR} - (WCRT_M' + WCRT_C')$
$WCRT_M' = WCRT_M + T_{SF}$	$WCRT_C' = WCRT_C + T_{SF}$	

$$T_{SF} \leq \frac{D_{CGR} - (WCET_M + WCET_C + WCET_A)}{2}$$

- La función de costo, utilizada para medir la calidad del modo de funcionamiento para cada uno de los escenarios de la aplicación, depende de las características y restricciones de cada aplicación. En este trabajo se considera que tanto el consumo de potencia, como las latencias en el lazo de control y su variabilidad, son importantes para definir la calidad de los diferentes modos de funcionamiento del sistema.
- Los periodos y plazos de las tareas están definidos por la aplicación. Para obtener los tiempos de cómputo para cada una de las frecuencias de funcionamiento de los procesadores de los nodos, se suponen conocidos los WCET de las tareas para la frecuencia más baja, y luego se calculan los WCET para las demás frecuencias empleando un factor de escalado de frecuencia, λ , por lo tanto:

$$f_i = \lambda_i * f_{baja}, \text{ luego } WCET_{f_i} = \frac{1}{\lambda_i} WCET_{f_{baja}}.$$

Las frecuencias seleccionadas para definir los modos de funcionamiento en cada nodo deben ser elegidas de tal forma que los WCET de las tareas de *Medición* y *Control* sean del orden de magnitud o cercano al de T_{SF} , lo anterior debido a que la red actúa como filtro y por lo tanto el finalizar estas actividades muy rápido no implica que el retraso total se disminuya considerablemente, y se podría estar realizando un consumo de energía innecesario. La disminución del WCET de la tarea de actuación si afecta de forma más directa el retraso total del sistema.

Adicionalmente, es importante considerar que algunos periféricos operan con señales de reloj externas a las del procesador y por lo tanto el escalado en la frecuencia del procesador no alterará su tiempo de respuesta.

- Para verificar la planificabilidad local en cada nodo se emplea la ecuación (1).
- Para verificar la planificabilidad *end-to-end* se utiliza el algoritmo *DeadlineMin* para encontrar los $WCRT_i$, los cuales corresponde con el valor de D^{\min} , y luego se verifica el cumplimiento de las ecuaciones escritas en la tabla 1.
- Dependiendo de los parámetros definidos en la función de costo, durante la búsqueda del modo de funcionamiento óptimo puede ser necesario la utilización de herramientas de simulación para la obtención de los valores que toma la función de costo para cada modo de funcionamiento.

Para presentar el uso del procedimiento de diseño propuesto se abordará un caso de estudio; en el cual se considera una red formada por tres nodos, sobre los cuales se implementan dos aplicaciones, un NCS y un sistema de detección de fallos en línea. La figura 4 presenta un diagrama de los componentes de cada una de las aplicaciones.

Figura 3. Procedimiento de diseño de WSAN

- El tiempo requerido para los cambios de contexto es considerado como parte del tiempo de cómputo de las tareas.
- El procesador principal y el coprocesador de cada nodo no se encuentran sincronizados entre si.

A continuación se describen los pasos del procedimiento de diseño:

- Para establecer el número de nodos requeridos para implementar la solución, se considera principalmente la distribución física de las variables del sistema, lo que determina el área de cobertura de la red, y el alcance de los sistemas de comunicaciones de los nodos.
- El valor de T_{SF} está relacionado con la máxima frecuencia de muestreo presente en la aplicación, el plazo de entrega del valor de esta variable y el número máximo de saltos que pueda tener el camino entre emisor y receptor. Por ejemplo, para un lazo de control, si las funciones de medición, control y actuación se encuentran en nodos independientes, los cuales a su vez se encuentran a un salto de distancia en la red, un criterio para seleccionar T_{SF} es:

Figura 4. Caso de estudio considerado

4.1 Consideraciones generales del caso de estudio

Para simplificar el espacio de soluciones al caso de estudio, se suponen nodos homogéneos con dos frecuencias de operación del procesador, baja y alta. En este caso se ha seleccionado un factor de escalado de frecuencia, λ , de 10, por lo tanto:

$$f_{alta} = 10f_{baja}, \text{ luego } WCET_{alta} = \frac{1}{10}WCET_{baja}$$

En el cálculo de los WCET de las tareas de *Medición* para la frecuencia alta, se añade un valor correspondiente al tiempo de cómputo de los ADC, el cual no se escala.

Los parámetros de configuración de la red para el caso de estudio propuesto son:

- Tramas de igual tamaño, 144 bits.
- Velocidad de comunicación de 250 kbps.
- Duración de cada ranura en la red TDMA de 1ms.
- El periodo con el que se repite el itinerario de las ranuras es de 35ms.

4.2 Consideraciones del sistema de detección de fallos

En el caso considerado, se supone que la degradación del sistema debida a algún fallo no se produce rápidamente, por lo que esta aplicación se realizará una vez por día, además no requiere de un plazo de respuesta muy exigente.

Esta aplicación está compuesta por dos tareas. La tarea que captura las muestras necesarias para ejecutar los algoritmos de detección de fallos se denomina *D_Medidor*, es periódica y se ejecuta hasta adquirir el número de muestras necesarias por los algoritmos. La otra tarea se ejecuta una sola vez y realiza los algoritmos de detección de fallos, se denomina *Técnicas_de_diagnóstico*.

Se supone que de acuerdo a la configuración física del sistema, la tarea *D_Medidor* estará pre-assignada al nodo 1, mientras que la tarea *Técnicas_de_diagnóstico* puede ser ejecutada en cualquiera de los tres nodos.

Los tiempos de cómputo supuestos para *D_Medidor* y *Técnicas_de_diagnóstico* se presentan en la tabla 2.

Tabla 2

Tarea	WCET (ms), frecuencia baja	WCET (ms), frecuencia alta
<i>D_Medidor</i>	2	0.3
<i>Técnicas_de_diagnóstico</i>	1000	100

Con base en lo anterior, el diagnóstico se realiza cada 24 h. En

este caso se supone que cuando se activa la tarea *D_Medidor* realiza 2048 medidas de las variables relacionadas con los fallos a detectar, con un periodo de 4 ms, el cual se supone idóneo para realizar el análisis por parte de los algoritmos de detección de fallos. Una vez esta tarea finaliza las mediciones, las almacena en la memoria e indica al middleware para que active la tarea *Técnicas_de_diagnóstico*, la cual procesa las muestras y genera un reporte del estado de la planta. Es decir que estos dos componentes se ejecutan en exclusión mutua.

Los periodos y plazos de estas tareas son, $T_{D_MEDIDOR} = 4ms$ y $D_{D_MEDIDOR} = 4ms$; La tarea *Técnicas_de_diagnóstico* tiene un periodo de $24h = 86.400.000ms$ y un plazo $D_{DIAGNÓSTICO} = 5000ms$. Este último valor solamente se asigna para realizar los análisis debido a que esta aplicación no tiene restricciones estrictas de tiempo real, aunque se espera que una vez se active finalice en un periodo corto de tiempo.

4.3 Consideraciones del NCS

En la mayoría de los trabajos realizados en NCS se analiza el desempeño del sistema considerando únicamente la estabilidad del lazo cerrado, y aunque éste es un requisito básico, en muchos casos prácticos los criterios de desempeño deseados en los sistemas de control se especifican en términos de la respuesta transitoria del sistema frente a una entrada específica. Por tal razón en este trabajo se presenta un modelo lineal del sistema en lazo cerrado que permite analizar el efecto que genera el retraso sobre la respuesta transitoria del sistema. Respecto del NCS se tienen las siguientes suposiciones:

- Sistema de control SISO (*Single Input – Single Output*).
- El retraso presente entre los instantes en que se realiza la medición y la actuación es menor o igual a T_s .
- La tarea de *Medición* es activada de forma periódica cada T_s . Mientras que las tareas de *Control* y *Actuación* se activan por eventos. Por consiguiente el sistema está sincronizado y es posible llevar el retraso de realimentación al lazo directo considerando un único retraso en el sistema, figura 5.

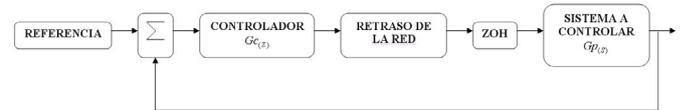


Figura 5. Representación del sistema sincronizado.

El retraso se puede modelar como $e^{-\tau_r s}$, con $\tau_r = (1-m)T_s$ y $0 \leq m \leq 1$. Donde τ_r representa el retraso equivalente a la suma de los retrasos entre medidor – controlador y controlador – actuador.

Empleando la transformada Z modificada es posible encontrar la función de transferencia en lazo cerrado del sistema entre dos periodos de muestreo consecutivos, de lo cual se obtiene:

$$H_{(Z,m)} = \frac{G_{C(Z)} G_{P'(Z,m)}}{1 + G_{C(Z)} G_{P'(Z,m)}} \quad (2)$$

Donde:

$$Gp'(z,m) = (1 - z^{-1})Z \left[\frac{Gp(s)}{S} \right]$$

$Gp'(z,m)$ es la transformada Z modificada de $Gp(s)$.

Para un sistema de primer orden $Gp(s) = \frac{K}{\tau_p S + 1}$, luego,

$$Gp'(z,m) = \frac{K}{Z} \left[1 - \frac{e^{-mT_s/\tau_p} (z-1)}{z - e^{-T_s/\tau_p}} \right] \quad (3)$$

Para el caso de estudio se supone $Gp(s) = \frac{0.3}{0.86s + 1}$, utilizando (3)

se obtiene,

$$Gp'(z,m) = \frac{0.3}{Z} \left[1 - \frac{e^{-mT_s/0.86} (z-1)}{z - e^{-T_s/0.86}} \right]$$

Con el fin de reducir el tiempo de asentamiento del sistema se utilizó un algoritmo PI de la forma,

$$U_{(z)} = \frac{k_p z + (k_i T_s - k_p)}{z - 1} \mathcal{E}_{(z)}$$

Con $k_p = 30.2778$ y $k_i = 25.463$.

Se supone como desempeño aceptable para el lazo de control el presentado en la figura 6, el cual se obtiene con $T_s = 110ms$ y

$\tau_r = 80ms$, luego se supone $D_{CGR} = 80ms$.

Figura 6. Respuesta del sistema de control para una señal de entrada cuadrada, con $T_s = 110ms$ y $\tau_r = 80ms$.

El planteamiento de la aplicación de control para el caso propuesto es el siguiente:

- La aplicación de control requiere de tres tareas, $C_Medidor$, $C_Controlador$ y $C_Actuador$, las cuales se pre-asignarán cada una en un nodo diferente, $C_Medidor$ al nodo 1, $C_Controlador$ al nodo 2 y $C_Actuador$ al nodo 3.
- Los tiempos de cómputo para las tres tareas se presentan en la tabla 3.

Tabla 3

Tarea	WCET (ms), frecuencia baja	WCET (ms), frecuencia alta
$C_Medidor$	2	0.3
$C_Controlador$	8	0.8
$C_Actuador$	1	0.1

4.4 Funciones de costo

Con el propósito de encontrar el modo de funcionamiento óptimo del sistema se plantean funciones de costo. En términos generales, los objetivos planteados en la optimización están

relacionados con la calidad del diseño del sistema de cómputo (mínimo consumo de potencia) y la calidad de la respuesta generada por los algoritmos de control (mínimos retrasos y jitter en el lazo de control), cuya importancia individual depende de las particularidades del contexto en el que se desempeña la aplicación.

Para analizar cada uno de los objetivos se han planteado dos funciones de costo, $F(Co)_{S\ Cómputo}$ para evaluar el costo de un modo de funcionamiento considerando parámetros relacionados con el sistema de cómputo, y $F(Co)_{Control}$ para evaluar la calidad de parámetros que afectan las características dinámicas del sistema de control, cada una de las cuales tiene la siguiente forma:

$$F(C_o) = \sum_0^i K_i \frac{C_i(C_o)}{C_i} + \sum_0^i K_{Ci} F_C(C_i, C_i(C_o))$$

K_i es el peso asignado al parámetro i ; $C_i(C_o)$ es la solución obtenida de un determinado modo de funcionamiento Co ; C_i es la restricción de diseño aplicado al i -ésimo parámetro considerado en la función y es usada como parámetro de normalización; K_{Ci} es el factor de peso para los términos de corrección y F_C es una función de corrección.

La función de corrección no contribuye a la función de costo cuando la solución está dentro del espacio permitido de búsqueda. La expresión utilizada para la función de corrección es:

$$F_C(C_i, C_i(C_o)) = r^2 [C_i, C_i(C_o)]$$

Donde la función $r[C_i, C_i(C_o)]$ corresponde a:

$$r(C_i, C_i(C_o)) = \max \left\{ 0, \frac{[C_i(C_o) - C_i]}{C_i} \right\}$$

Con relación al sistema de cómputo, el objetivo es generar una solución óptima que aumente el tiempo de vida de la WSA, considerando que todos sus nodos dependen de baterías para su funcionamiento; lo cual está restringido principalmente por la disponibilidad de energía en cada uno de los nodos de la red. Esto se logra empleando la frecuencia de operación más baja para cada nodo, pero logrando un porcentaje de utilización similar entre los nodos. Las posibles acciones para incidir en el consumo de potencia son:

- Escalado de frecuencia y voltaje del procesador. Actúa sobre el consumo de potencia en un único nodo y escala los $WCET_i$ y los $WCRT_i$ en el respectivo nodo.
- Migración de tareas entre los nodos, lo cual se logra por la activación de la tarea respectiva en el nodo seleccionado, para lo cual se considera que en cada nodo existe una réplica de cada una de las tareas. Busca equilibrar el consumo de potencia entre los nodos de la WSA y en general afecta los $WCRT_i$ en varios nodos.

En el enfoque empleado en este trabajo, por cuanto se conoce el WCET de las tareas y las diferentes frecuencias de operación de los nodos, y puesto que el consumo de potencia está relacionado directamente con estos parámetros, se realiza un análisis indirecto del consumo de potencia de las diferentes configuraciones de operación del sistema, empleando los

siguientes índices:

- fop_{M_j} , es la media entre las frecuencias de operación de los n nodos que conforman la WSN para un modo de funcionamiento j . Se emplea para encontrar el modo de funcionamiento donde los nodos operen a la menor frecuencia posible.
- Δfop_{DEj} , es la desviación estándar de las frecuencias de operación de los nodos. Se utiliza para buscar un consumo de potencia equilibrado entre los nodos.
- ΔU_{DEj} , es la desviación estándar de los porcentajes de utilización de los nodos. Se utiliza para buscar un consumo de potencia equitativo entre los nodos. Donde el porcentaje de utilización de un nodo que ejecuta i tareas es:

$$U = \sum_{i=1}^n \frac{WCET_i}{T_i}$$

Respecto del desempeño del sistema de control se consideraron dos parámetros. Estos son:

- r_M : Valor medio de los retrasos τ_r , medidos desde el instante de activación de $C_Medidor$ y el instante de finalización de $C_Actuador$.
- Δr_{DE} : Es la desviación estándar entre los retrasos τ_r , idealmente debe tender a cero, de tal forma que el retraso pueda ser compensado más fácilmente en el algoritmo de control.

Junto con los parámetros a analizar se establecen los valores máximos de operación. Estos valores dependen de la arquitectura de síntesis de componentes que se ha seleccionado, las restricciones planteadas por la aplicación y los parámetros de cada modo de funcionamiento; actúan como restricciones para la selección de alternativas de funcionamiento, y son usados como valores de referencia para ajustar mejor las restricciones del diseño y evitar buscar una solución imposible. Estos son:

- $fop_{M_j}^{\max}$, valor promedio máximo de los fop_{M_j} .
- Δfop_{DEj}^{\max} , máxima desviación estándar entre las frecuencias de operación de los nodos.
- ΔU_{DEj}^{\max} , máxima desviación estándar entre los porcentajes de utilización de los nodos.
- r_M^{\max} , valor máximo de r_M , este valor podría ser menor que aquel con el que se realizan los análisis de planificabilidad, de tal forma que se elijan soluciones cuyo retraso *end-to-end* afecte menos la respuesta del algoritmo de control. Para el caso de estudio se asigno un valor de 80 ms, de acuerdo con el desempeño supuesto como aceptable para el lazo de control.
- Δr_{DE}^{\max} . Máxima desviación estándar entre los retrasos presentes en el sistema.

4.5 Búsqueda del modo de funcionamiento óptimo

Existen tres posibles escenarios en el caso de estudio propuesto:

- Cuando solo está en funcionamiento la aplicación de control.
- Cuando funcionan la aplicación de control y la tarea $D_Medidor$.

- Cuando funcionan la aplicación de control y la tarea $Técnicas_de_diagnóstico$.

Para la obtención de los parámetros temporales de cada modo de funcionamiento, requeridos en las funciones de costo, el caso fue simulado empleando la herramienta Truetime (Cervin, 2003). Los pesos asignados a los coeficientes de la función de costo fueron:

- Sistema de cómputo
 $K_{f_M} = 0.5$, $K_{\Delta f_{DE}} = 0.2$, $K_{\Delta U_{DE}} = 0.3$.
 $K_{cf_M} = K_{C\Delta f_{DE}} = K_{C\Delta U_{DE}} = 150$.

Se da mas importancia al valor de los fop_{M_j} (soluciones de frecuencias mas bajas). El ΔU_{DEj} tiene mayor peso que Δfop_{DEj} por estar mas relacionado con la migración de tareas entre los nodos.

- Sistema de control
 $K_{r_M} = 0.6$, $K_{\Delta r_{DE}} = 0.4$, $K_{cr} = K_{\Delta r} = 150$.

Se dio mas importancia al valor del retraso *end-to-end* (r_M), porque este parámetro afecta mas la respuesta dinámica del sistema.

Los modos de funcionamiento para los escenarios 1, 2 y 3, y el valor obtenido de las funciones de costo se presentan en las tablas 4, 5 y 6 respectivamente.

Los resultados obtenidos del análisis de las tablas 4, 5 y 6 fueron:

- Para el escenario 1, el modo de funcionamiento 3 presenta el menor valor de $F(Co)_{S_Cómputo}$, mientras que el modo de funcionamiento 8 presenta el menor valor de $F(Co)_{S_Control}$.
- Para el escenario 2, el modo de funcionamiento 2 presenta el menor valor de $F(Co)_{S_Cómputo}$, mientras que el modo de funcionamiento 8 presenta el menor valor de $F(Co)_{S_Control}$.
- Para el escenario 3, el modo de funcionamiento 19 presenta el menor valor de $F(Co)_{S_Cómputo}$, mientras que los modos de funcionamiento 8, 16 y 24 presentan el menor valor de $F(Co)_{S_Control}$.

Con base en lo anterior la optimización de ambos criterios es un problema multiobjetivo. Para analizar la forma de encontrar el modo de funcionamiento óptimo se observó el frente de Pareto, figura 7.

Figura 7. Frente de Pareto para el escenario 3.

Debido a que el frente de Pareto es convexo, es posible encontrar la solución óptima al problema multiobjetivo empleando una combinación lineal de pesos. Para lo cual se planteo:

$$F(C_o) = K_{S_Cómputo} F(C_o)_{S_Cómputo} + K_{Control} F(C_o)_{Control}$$

Donde $K_{C\acute{o}mputo}$ y $K_{Control}$ son coeficientes que indican el peso de cada tipo de parámetros en la calidad de la solución deseada, lo cual depende del tipo de aplicación.

Los valores de los coeficientes utilizados en el caso de estudio fueron:

$$K_{C\acute{o}mputo} = 0.4, K_{Control} = 0.6$$

Los resultados de la optimización son:

- Para el escenario 1, el modo de funcionamiento óptimo es el 8.
- Para el escenario 2, el modo de funcionamiento óptimo es el 2.
- Para el escenario 3, el modo de funcionamiento óptimo es el 24.

Tabla 4. Escenario 1

	Nodo	Componentes	Frecuencia de operación	$D_{Mín} [ms]$	$D_{Máx} [ms]$	$r_M [ms]$	$F(Co)_{S_C\acute{o}mputo}$	$F(Co)_{Control}$
Modo Func. 1	1	$C_Medidor,$	1	2	No es planificable <i>end – to – end</i>			
	2	$C_Controlador$	1	8				
	3	$C_Actuador$	1	1				
Modo Func. 2	1	$C_Medidor,$	10	0.3	1	52.1250000	0.6835823082	0.7909374999
	2	$C_Controlador$	1	8	8.7			
	3	$C_Actuador$	1	1	1.7			
Modo Func. 3	1	$C_Medidor$	1	2	8.2	60.6750000	0.4428184011	0.8550624999
	2	$C_Controlador$	10	0.8	7			
	3	$C_Actuador$	1	1	7.2			
Modo Func. 4	1	$C_Medidor$	10	0.3	8.2	51.9250000	0.5740133033	0.7894375000
	2	$C_Controlador$	10	0.8	8.7			
	3	$C_Actuador$	1	1	8.9			
Modo Func. 5	1	$C_Medidor$	1	2	No es planificable <i>end – to – end</i>			
	2	$C_Controlador$	1	8				
	3	$C_Actuador$	10	0.1				
Modo Func. 6	1	$C_Medidor$	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	$C_Controlador$	1	8	9.6			
	3	$C_Actuador$	10	0.1	1.7			
Modo Func. 7	1	$C_Medidor$	1	2	9.1	59.7750000	0.6139969812	0.8483124999
	2	$C_Controlador$	10	0.8	7.9			
	3	$C_Actuador$	10	0.1	7.2			
Modo Func. 8	1	$C_Medidor$	10	0.3	9.1	51.0250000	0.5240133033	0.7826874999
	2	$C_Controlador$	10	0.8	9.6			
	3	$C_Actuador$	10	0.1	8.9			

Tabla 5. Escenario 2

	Nodo	Componentes	Frecuencia de operación	$D_{Mín} [ms]$	$D_{Máx} [ms]$	$r_M [ms]$	$F(Co)_{S_C\acute{o}mputo}$	$F(Co)_{Control}$
Modo Func. 1	1	$C_Medidor$ y $D_Medidor$	1	4	No es planificable <i>end – to – end</i>			
	2	$C_Controlador$	1	8				
	3	$C_Actuador$	1	1				
Modo Func. 2	1	$C_Medidor$ y $D_Medidor$	10	0.6	1	52.1250000	0.4386745741	0.7909374999
	2	$C_Controlador$	1	8	8.4			
	3	$C_Actuador$	1	1	1.4			
Modo Func. 3	1	$C_Medidor$ y $D_Medidor$	1	4	8.2	60.6750000	0.6975970512	0.8550624999
	2	$C_Controlador$	10	0.8	5			
	3	$C_Actuador$	1	1	5.2			
Modo Func. 4	1	$C_Medidor$ y $D_Medidor$	10	0.6	8.2	51.9250000	0.5905916227	0.7894375000
	2	$C_Controlador$	10	0.8	8.4			
	3	$C_Actuador$	1	1	8.6			
Modo Func. 5	1	$C_Medidor$ y $D_Medidor$	1	4	No es planificable <i>end – to – end</i>			
	2	$C_Controlador$	1	8				
	3	$C_Actuador$	10	0.1				
Modo Func. 6	1	$C_Medidor$ y $D_Medidor$	10	0.6	1.9	51.2250000	0.5934397554	0.7841874999
	2	$C_Controlador$	1	8	9.3			
	3	$C_Actuador$	10	0.1	1.4			
Modo Func. 7	1	$C_Medidor$ y $D_Medidor$	1	4	9.1	59.7750000	0.8500000000	0.8483124999
	2	$C_Controlador$	10	0.8	5.9			
	3	$C_Actuador$	10	0.1	5.2			
Modo Func. 8	1	$C_Medidor$ y $D_Medidor$	10	0.6	9.1	51.0250000	0.5430885242	0.7826874999
	2	$C_Controlador$	10	0.8	9.3			
	3	$C_Actuador$	10	0.1	8.6			

Tabla 6. Escenario 3

	Nodo	Componentes	Frecuencia de operación	D_{Min} [ms]	$D_{Máx}$ [ms]	r_M [ms]	$F(Co)_s$ Cómputo	$F(Co)_{Control}$
Modo Func. 1	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	1	2	No es planificable end – to – end			
	2	<i>C_Controlador</i>	1	8				
	3	<i>C_Actuador</i>	1	1				
Modo Func. 2	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	10	0.3	1	52.1250000	0.6835817445	0.7909374999
	2	<i>C_Controlador</i>	1	8	8.7			
	3	<i>C_Actuador</i>	1	1	1.7			
Modo Func. 3	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	1	2	8.2	60.6750000	0.4428670973	0.8550624999
	2	<i>C_Controlador</i>	10	0.8	7			
	3	<i>C_Actuador</i>	1	1	7.2			
Modo Func. 4	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	10	0.3	8.2	51.9250000	0.5740087882	0.7894375000
	2	<i>C_Controlador</i>	10	0.8	8.7			
	3	<i>C_Actuador</i>	1	1	8.9			
Modo Func. 5	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	1	2	No es planificable end – to – end			
	2	<i>C_Controlador</i>	1	8				
	3	<i>C_Actuador</i>	10	0.1				
Modo Func. 6	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	<i>C_Controlador</i>	1	8	9.6			
	3	<i>C_Actuador</i>	10	0.1	1.7			
Modo Func. 7	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	1	2	9.1	59.7750000	0.6140430781	0.8483124999
	2	<i>C_Controlador</i>	10	0.8	7.9			
	3	<i>C_Actuador</i>	10	0.1	7.2			
Modo Func. 8	1	<i>C_Medidor, Técnicas_de_diagnóstico</i>	10	0.3	9.1	51.0250000	0.5240123163	0.7826874999
	2	<i>C_Controlador</i>	10	0.8	9.6			
	3	<i>C_Actuador</i>	10	0.1	8.9			
Modo Func. 9	1	<i>C_Medidor</i>	1	2	No es planificable end – to – end			
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	1	8				
	3	<i>C_Actuador</i>	1	1				
Modo Func. 10	1	<i>C_Medidor</i>	10	0.3	1	52.1250000	0.6835848327	0.7909374999
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	1	8	8.7			
	3	<i>C_Actuador</i>	1	1	1.7			
Modo Func. 11	1	<i>C_Medidor</i>	1	2	8.2	60.6750000	0.4428083399	0.8550624999
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	10	0.8	7			
	3	<i>C_Actuador</i>	1	1	7.2			
Modo Func. 12	1	<i>C_Medidor</i>	10	0.3	8.2	51.9250000	0.5740105625	0.7894375000
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	10	0.8	8.7			
	3	<i>C_Actuador</i>	1	1	8.9			
Modo Func. 13	1	<i>C_Medidor</i>	1	2	No es planificable end – to – end			
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	1	8				
	3	<i>C_Actuador</i>	10	0.1				
Modo Func. 14	1	<i>C_Medidor</i>	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	1	8	9.6			
	3	<i>C_Actuador</i>	10	0.1	1.7			
Modo Func. 15	1	<i>C_Medidor</i>	1	2	9.1	59.7750000	0.6139858071	0.8483124999
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	10	0.8	7.9			
	3	<i>C_Actuador</i>	10	0.1	7.2			
Modo Func. 16	1	<i>C_Medidor</i>	10	0.3	9.1	51.0250000	0.5240140891	0.7826874999
	2	<i>C_Controlador, Técnicas_de_diagnóstico</i>	10	0.8	9.6			
	3	<i>C_Actuador</i>	10	0.1	8.9			

Tabla 6. Escenario 3, continuación

	Nodo	Componentes	Frecuencia de operación	D_{Min} [ms]	$D_{Máx}$ [ms]	r_M [ms]	$F(Co)_S$ Cómputo	$F(Co)_{Control}$
Modo Func. 17	1	$C_{Medidor}$	1	2	No es planificable end – to – end			
	2	$C_{Controlador}$	1	8				
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	1	1				
Modo Func. 18	1	$C_{Medidor}$	10	0.3	1	52.1250000	0.6835638039	0.7909374999
	2	$C_{Controlador}$	1	8	8.7			
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	1	1	1.7			
Modo Func. 19	1	$C_{Medidor}$	1	2	8.2	60.6750000	0.4428012028	0.8550624999
	2	$C_{Controlador}$	10	0.8	7			
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	1	1	7.2			
Modo Func. 20	1	$C_{Medidor}$	10	0.3	8.2	51.9250000	0.5740488069	0.7894375000
	2	$C_{Controlador}$	10	0.8	8.7			
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	1	1	8.9			
Modo Func. 21	1	$C_{Medidor}$	1	2	No es planificable end – to – end			
	2	$C_{Controlador}$	1	8				
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	10	0.1				
Modo Func. 22	1	$C_{Medidor}$	10	0.3	1.9	51.2250000	0.8500000000	0.7841874999
	2	$C_{Controlador}$	1	8	9.6			
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	10	0.1	1.7			
Modo Func. 23	1	$C_{Medidor}$	1	2	9.1	59.7750000	0.6139936994	0.8483124999
	2	$C_{Controlador}$	10	0.8	7.9			
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	10	0.1	7.2			
Modo Func. 24	1	$C_{Medidor}$	10	0.3	9.1	51.0250000	0.5240099793	0.7826874999
	2	$C_{Controlador}$	10	0.8	9.6			
	3	$C_{Actuador},$ $Técnicas_de_diagnóstico$	10	0.1	8.9			

5. RESULTADOS DE SIMULACIÓN

En (Varma, 2005) se presenta el siguiente modelo de consumo de potencia para procesadores XScale:

$$P = P_{Estática} + P_{Dinámica} \Rightarrow P = VI_{Fugas} + \frac{1}{2} C_L V_{dd}^2 f$$

El modelo empleado para simulación, considerando el mismo factor de escalado para la frecuencia y el voltaje, es:

- Durante el tiempo de ejecución de cada tarea:

$$P = 0.01 + 0.02 \left(\frac{f_{funcionamiento}}{f_{baja}} \right)^3. \quad \text{Donde}$$

$f_{funcionamiento}$ es la frecuencia con la que está funcionando el nodo en ese momento.

- Al finalizar el periodo ICI, el procesador se llevará a un estado de bajo consumo de potencia, durante el cual $P = 0.001$.

Los resultados de simulación coincidieron con los análisis realizados en el proceso de optimización. En las figuras 8 y 9 se presenta el consumo de potencia en los nodos, la planificación de tareas en el nodo 3 y la respuesta del sistema de control, para los modos de funcionamiento 16 y 19 del escenario 3.

De las figuras 8 y 9 se puede apreciar que:

- Las dos configuraciones cumplen con los plazos end-to-end propuestos.
- En ambos casos el sistema de control cumple con los objetivos propuestos. Aunque se observa una mejor

respuesta en el modo de funcionamiento 16 por presentar un retraso mas pequeño.

- El consumo de potencia en los nodos es mayor en el modo de funcionamiento 16 que en el 19, lo que disminuye el tiempo de vida de la aplicación, ocasionando la inestabilidad del sistema de control poco antes de 7s de iniciada la simulación, como consecuencia de la terminación de la energía en el nodo 2.

Estos resultados resaltan la importancia de realizar un balance entre los parámetros más relevantes en el diseño de WSA, en este caso los retrasos en el lazo de realimentación y el consumo de potencia, pues se puede apreciar como a pesar de que en las dos configuraciones se cumple con los objetivos de control propuestos, el consumo de potencia es mucho mayor en el modo de funcionamiento 16, lo que generaría un incremento en los costos de implementación del sistema.

6. CONCLUSIONES

En este trabajo se presentó un procedimiento de diseño, que permite encontrar el modo de funcionamiento óptimo en la aplicación, considerando parámetros del sistema de cómputo y el desempeño del sistema de control, el cual se fundamenta en dos funciones de costo relacionadas con el consumo de potencia y los retrasos en los lazos de control, y emplea una combinación lineal de pesos entre las dos funciones para encontrar el modo de funcionamiento óptimo del sistema. Los parámetros empleados en las funciones de costo dependen de las características y restricciones de cada aplicación.

retrasos presentes en el sistema.

Figura 8a. Consumo de potencia en el nodo 1.

Figura 9a. Consumo de potencia en el nodo 1.

Figura 8b. Consumo de potencia en el nodo 2.

Figura 9b. Consumo de potencia en el nodo 2.

Figura 8c. Consumo de potencia en el nodo 3.

Figura 9c. Consumo de potencia en el nodo 3.

Figura 8d. Planificación de tareas en el nodo 3. E, W e I indican los estados *Execution*, *Waiting* e *Idle* de la tarea *C_Actuador*.

Figura 9d. Planificación de tareas en el nodo 3. E, W e I indican los estados *Execution*, *Waiting* e *Idle* de la tarea *C_Actuador*.

Figura 8e. Respuesta del sistema de control. El sistema de control se inestabiliza como consecuencia de la terminación de la energía en el nodo 2.

Figura 9e. Respuesta del sistema de control. Se aprecia un mayor tiempo de vida de la aplicación, debido a la disminución del consumo de potencia respecto del generado en el modo de funcionamiento 16.

Figura 8. Resultados de simulación del escenario 3, modo de funcionamiento 16.

Figura 9. Resultados de simulación del escenario 3, modo de funcionamiento 19.

Adicionalmente, se presentó un método para analizar la planificabilidad *end-to-end* en aplicaciones de NCS, para el cual la red utiliza un protocolo de control de acceso al medio TDMA con GTSs y los planificadores de tareas en los nodos son EDF.

El trabajo futuro será orientado en las siguientes direcciones:

Los resultados obtenidos corroboran la validez de las funciones y el método propuestos, y permiten apreciar la importancia de obtener configuraciones de funcionamiento que realicen un buen balance de la potencia consumida en los nodos y minimicen los

- Debido a la presencia de obstáculos y al área en la que se distribuyen estas aplicaciones, en algunos casos puede ser necesario interconectar varios segmentos de red; para lo cual es importante considerar los algoritmos de encaminamiento de mensajes en el procedimiento de diseño, cuyo desempeño afectan los

retrasos y el consumo de potencia en el sistema.

- La consideración de todos los posibles modos de funcionamiento en los diferentes escenarios durante el proceso de optimización, permite encontrar la mejor configuración pero requiere un elevado tiempo de análisis; por lo tanto es importante el planteamiento de herramientas de cómputo para automatizar el proceso, y de heurísticas para reducir las configuraciones de funcionamiento analizadas.
- Las herramientas de diseño y simulación ofrecen una alternativa para observar el desempeño dinámico del sistema, pero como consecuencia de la gran cantidad de nodos en la red, la arquitectura de los mismos y que cada uno ejecuta más de una tarea, es difícil observar el desempeño del sistema en todas las posibles configuraciones, y por tanto se dificulta garantizar la no presencia de bloqueos y la alcanzabilidad de ciertos estados en el sistema, lo que afecta la confiabilidad del mismo. Por lo cual es importante proponer modelos formales que faciliten el análisis y diseño de estas aplicaciones.

AGRADECIMIENTOS

Este trabajo fue financiado parcialmente por el proyecto D2ARS de CYTED, código UNESCO: 120325;330417;120314;120305, y por el proyecto SIDIRELI DPI2008-06737-C02-01/02 financiado por el Ministerio de Ciencia e Innovación español y fondos europeos FEDER.

REFERENCIAS

- Astrom K. and Wittenmark B.: *Computer Control Systems. Theory and design*. Prentice Hall, 1997.
- Audsley N., Burns A., Richardson M., Tindell K., and Wellings A.J.: "Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling". *Software Engineering Journal*, September 1993.
- Balbastre P., Ripoll I., and Crespo A.: "Minimum Deadline Calculation for Periodic Real-Time Tasks in Dynamic Priority Systems". *IEEE Transactions on computers*. **Vol. 57, No. 1**, January 2008.
- Bonivento A., Sangiovanni-Vincentelli A., Graziosi F., Santucci F.: "SERAN: A Semi Random Protocol Solution for Clustered Wireless Sensor Networks", Proc. of MASS 2005. 2005.
- Bonivento A., Carloni L. and Sangiovanni-Vincentelli A.: "Platform-Based Design of Wireless Sensor Networks for Industrial Applications". *Proceedings of the conference on Design, automation and test in Europe*, pp. 1103-1107. 2006a.
- Bonivento A., Carloni L. and Sangiovanni-Vincentelli A.: "Platform based design for wireless sensor networks". *Mobile Networks and Applications* - Springer, pp.469-485. 2006b.
- Branicky M., Phillips M., and Zhang W.: "Scheduling and feedback codesign for networked control systems". In *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, USA. 2002.
- Cena I G., Bertolotti A C., C. Zunino V.: "Industrial Applications of IEEE 802.11e WLANs". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.
- Cervin A., Henriksson D., Lincoln B., Eker J., Årzén K.: "How Does Control Timing Affect Performance?, Analysis and Simulation of Timing Using Jitterbug and TrueTime". *IEEE Control Systems Magazine*, June 2003.
- Dormido S., Sánchez J., Kofman E.: "Muestreo, control y comunicación basados en eventos. *Revista Iberoamericana de Automática e Informática Industrial*. **Vol. 5, No. 1**, pp. 5-26, 2008.
- Gregory C W., Octavian B., Linda B.: "Asymptotic Behavior of Networked Control Systems". *Submitted to Control Applications Conference, International Conference on Control Applications*. Hawaii, USA. August 22-21, 1999.
- Hespanha J. P., Xu Y.: "A Survey of Recent Results in Networked Control Systems". *Proceedings of the IEEE*. **Vol. 95, No. 1**, January 2007.
- Hristu-Varsakelis D., Levine W. S. (Eds.): "*Handbook of Networked and Embedded Control Systems*". Páginas: 677-720. Birkhäuser 2005.
- Hu S., Yan W.: "Stability of Networked Control Systems Under a Multiple-Packet Transmission Policy". *IEEE Transactions on Automatic Control*. **Vol. 53**, pp. 1706-1711. August 2008.
- Huang D., Nguang S. K.: "State Feedback Control of Uncertain Networked Control Systems With Random Time Delays". *IEEE Transactions on Automatic Control*. **Vol. 53**, pp. 829-834. April 2008.
- Koubâa A., Alves M., and Tovar E.: "GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks", in *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2006)*. Rhodes Island (Greece): IEEE, 2006.
- Koubâa A., Cunha A., Alves M.: "A Time Division Beacon Scheduling Mechanism for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks". *Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference on IEEE Computer Society* 2007.
- Lee S., Park J. H., Ha K. N., Lee K. C.: "Wireless Networked Control System Using NDIS-based Four-Layer Architecture for IEEE 802.11b". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.
- Lennvall T., Svensson S., Hekland F.: "A Comparison of WirelessHART and Zigbee for Industrial Applications". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.
- Lester H. J.: "System architecture for wireless sensor networks". *PhD thesis*. University of California, Berkeley. 2003.
- Marinoni M., Buttazzo G.: "Elastic DVS Management in Processors With Discrete Voltage/Frequency Modes". *IEEE Transactions on industrial informatics*, **vol. 3**, No. 1, 2007.
- Martínez D., Blanes F., Simo J., Crespo A.: "Evaluación del Comportamiento Temporal de Sistemas Distribuidos de Control Sobre IEEE 802.15.4 y CAN". *21st Symposium on Integrated Circuits and Systems Design – Workshop on Sensor Networks and Applications*. Gramado, Brasil. Septiembre de 2008.
- Martínez D., Blanes F., Simo J., Crespo A.: "Wireless Sensors and Actuators Networks: Characterization and Cases Study for Confined Spaces Healthcare and Control Applications". *Scalable Computing: Practice and Experience Scientific International Journal for Parallel and Distributed Computing*. **Vol. 10, no.3**. pp. 291 – 305. 2009.
- Mejia-Alvarez P., Levner E., and Mosse D.: "Power-Optimized Scheduling Server for Real-Time Tasks," *Proc. IEEE Real-*

- Time and Embedded Technology and Applications Symp. (RTAS '02)*, p. 239, 2002.
- Meshkova E., Riihijarvi J., Oldewurtel F., Jardak Ch. and Mahonen P.: "Service-Oriented Design Methodology for Wireless Sensor Networks: A View through Case Studies". *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2008.
- Moraes R., Portugal P., Vasques F., Fonseca J. A.: "Limitations of the IEEE 802.11e EDCA Protocol when Supporting Real-Time Communication". *IEEE International Workshop on Factory Communication Systems*, 21-23 May 2008.
- Pantazis, N.A.; Vergados, D.D.: "A survey on power control issues in wireless sensor networks". *IEEE Communications Surveys & Tutorials*, 4th Quarter 2007. vol 9, No. 4.
- Pillai P. and Shin K.G.: "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems". *Proc. ACM Symp. Operating Systems Principles*, pp. 89-102, 2001.
- Prasad V., Yan T., Jayachandran P., Li Z., Son S., Stankovic J., Hansson J. and Abdelzaher T.: "ANDES: an ANalysis-based DEsign tool for wireless Sensor networks". *28th IEEE International Real-Time Systems Symposium, IEEE Computer Society*, 2007.
- Ripoll I., Crespo A., and Mok A.: "Improvement in feasibility testing for real-time tasks". *Journal of Real-Time Systems*, 11:19-40, 1996.
- Saewong S. and Rajkumar R.: "Practical Voltage-Scaling for Fixed-Priority RT-Systems". *Proceedings of the 9th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'03)* 2003 IEEE.
- Salt J., Casanova V., Cuenca A., Pizá R.: "Sistemas de control basados en red modelado y diseño de estructuras de control". *Revista Iberoamericana de Automática e Informática Industrial*. **Vol. 5, No. 3**, pp. 5-20, 2008.
- Spuri M.: "Holistic Analysis for Deadline Scheduled Real-Time Distributed Systems". *Tech. Rep. RR-2873*, INRIA, France, April 1996.
- Tabbara, M., Netic, D.; Teel, A.R.: "Stability of Wireless and Wireline Networked Control Systems". *Automatic Control. IEEE Transactions on Automatic Control*. **Vol. 52**, pp. 1615-1630. September 2007.
- Tindell K. and Clark J.: "Holistic Schedulability Analysis for Distributed Hard Real-Time Systems". *Microprocessors and Microprogramming* 40, 1994.
- Tindell K., Burns A., and Wellings A.J.: "Analysis of Hard Real-Time Communications". *The Journal of Real-Time Systems* 9, 1995.
- TinyOS: <http://www.tinyos.net/>
- Varma A., Debes E., Kozintsev I. and Jacob B.: "Instruction-Level Power Dissipation in the Intel XScale Embedded Microprocessor". *Proceedings of the SPIE, 17th Annual Symposium on Electronic Imaging Science & Technology*. **Vol. 5683**, pp. 1-8. 2005.
- Walsh G.C., Ye H., and Bushnell L.: "Stability analysis of networked control systems". *IEEE Transactions on Control Systems Technology*. **Vol. 10, no.5**, pp.438-446, 2002.
- WirelessHART:
http://www.hartcomm2.org/hart_protocol/wireless_hart/wirelesshart_datasheet.pdf
- Xiong J., Lam, J.: "Stabilization of Networked Control Systems With a Logic ZOH". *IEEE Transactions on Automatic Control*. **Vol. 54**, pp. 358-363. February 2009.
- Yang T.C.: "Networked control system: a brief survey". *IEE Proc.-Control Theory Appl.* **Vol. 153, No. 4**, July 2006.
- Zhang W., Branicky M., and Phillips S.: "Stability of networked control systems". *IEEE Control Systems Magazine*. Pp. 84-99. 2001.
- Zhu Y. and Mueller F.: "Feedback EDF Scheduling Exploiting Dynamic Voltage Scaling". *Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*, 2004 IEEE.
- Zigbee Specification. <http://www.Zigbee.org>