

Definición del entorno de trabajo para el Diseño de Sistemas de Control que consideran requerimientos de tiempo crítico - Análisis de desempeño de Tiempo Real

Gregory P. Bustamante R.
gregorypekyakov.bustamante@gmail.com

Jorge A. Nava A.
jorgeantonio.navaamador@gmail.com

ABSTRACT

Se presenta los aspectos necesarios para llevar adelante el diseño de Sistemas de Control considerando los requerimientos de tiempo críticos, de tal manera de ofrecer un ambiente de trabajo enfocado, a lograr mejorar el desempeño de los sistemas, en cuanto al cumplimiento de sus tiempos de respuesta, debido a que por lo general solo se consideran los requerimientos funcionales. Consiguientemente el entorno que se constituye, permitiría definir la base de Análisis de Tiempo Real, considerando herramientas “libres”, como Scicoslab/Scicos, RTAI y RTAI-Lab las cuales son usadas en los distintos laboratorios de Sistemas de Control de las universidades en el mundo, por ejemplo en el laboratorio de Mecatrónica de la Universidad SUPSI (Scuola universitaria professionale della Svizzera italiana) o en el grupo de trabajo de Computadoras y Tiempo Real de la Universidad de Cantabria.

Keywords – Requerimientos Funcionales, Tiempo Real, Sistemas de Control, Scicoslab, RTAI.

I. INTRODUCCIÓN

EL poco conocimiento de los sistemas en tiempo real en Bolivia, y el constante avance de trabajos de investigación en este campo, particularmente en el caso de sistemas de tiempo real estricto, obligan a abordar un aspecto que es necesario para los ingenieros electrónicos que se especializan en la Ingeniería de Control, especialmente de aquellos que trabajan en el desarrollo de sistemas con altas exigencias de respuesta temporal para la supervisión y control de procesos.

Los ingenieros de control desarrollan sistemas que solo consideran requisitos funcionales y dejan de lado los requisitos temporales, esto debido a la poca orientación que existe en este campo.

Esta es la motivación que ha impulsado a que la carrera de Ingeniería Electrónica incorpore dentro de sus líneas de investigación, la de “Desempeño de Sistemas de Tiempo Real”, aprobada por el Honorable Consejo de Carrera durante el primer semestre de la gestión 2014.

Esta Línea de Investigación tiene el propósito de impulsar el trabajo en Sistemas de Tiempo Real, para complementar la formación del pregrado, así como vincularnos con universidades extranjeras. Entre sus objetivos se tienen:

- Formular una metodología para el modelado y análisis de los requisitos, configuración y despliegue de sistemas distribuidos heterogéneos que permitan la gestión de estos sistemas.
- Desarrollar una metodología de modelado y análisis que basada en la descripción del sistema (plataforma, aplicaciones, carga de trabajo y requisitos temporales) permita garantizar el cumplimiento de requisitos combinados de tiempo real estricto, de distribuciones especificadas de las latencias en la ejecución de transacciones y métricas de productividad y rendimiento.
- Desarrollo de técnicas de extracción de los modelos de comportamiento en base a la instrumentación estándar disponible en la plataforma y de los modelos de despliegue existentes. Se consideran sistemas no críticos (firm real-time) en los que la observación de la ejecución del sistema sobre el entorno real proporciona información estadística significativa para la configuración del sistema.
- Desarrollo de técnicas de verificación del comportamiento del sistema y del cumplimiento de sus requisitos en base a la monitorización de su ejecución sobre el entorno real, de su modelo de despliegue y comportamiento.

Como producto de los primeros pasos que se dieron, es que se pueden apreciar los resultados y planteamientos que se exponen en este artículo, donde presentamos los aspectos necesarios para llevar adelante el diseño de

Sistemas de Control considerando los requerimientos de tiempo críticos, de tal manera de ofrecer un ambiente de trabajo enfocado, a lograr mejorar el desempeño de los sistemas, en cuanto al cumplimiento de sus tiempos de respuesta, debido a que por lo general solo se consideran los requerimientos funcionales. Consiguientemente el entorno que se constituye, permitiría definir la base de Análisis de Tiempo Real, considerando herramientas “libres”, como Scicoslab/Scicos, RTAI y RTAI-Lab

II. ANÁLISIS DE TIEMPO REAL

Hoy en día se utilizan plataformas de computadores distribuidas asociadas a entornos industriales, como redes de distribución de la energía, sistemas de transportes, y otros, de gestión de infraestructuras (aeropuertos, controles de accesos a edificios, servicios de información en museos, sistemas de alerta temprana) y de supervisión de entornos (gestión de la seguridad de zonas estratégicas, vigilancia de fronteras, etc.).

Esto supone la gestión de sistemas complejos de computadores distribuidos que dan soporte a múltiples y diferentes aplicaciones software en las que se combinan requisitos de comportamiento temporal de diferente naturaleza:

- Requisitos de tiempo real estrictos para la respuesta a eventos temporizados o del entorno.
- Requisitos sobre los valores medios y la dispersión de las latencias en las transacciones que se ejecutan.
- Requisitos de rendimiento, productividad y calidad de servicio.

La tecnología que actualmente domina en estos dominios implementa la plataforma en base a nodos de computadores con sistema operativo Linux, interconectados mediante redes de Ethernet. Recientemente se ha incorporado al sistema operativo Linux el parche Preempt_RTkernelpatch que permite combinar en un mismo nodo planificadores estándar de no tiempo real, basados en RoundRobin con planificadores con políticas de tiempo real de prioridades fijas.

Así mismo, junto a los servicios de comunicación estándar de Linux basados en sockets y en servicios de gestión y control de la anchura de banda que se asocian a los enlaces (tales como TrafficControlService), se han incorporado protocolos que hacen uso de los protocolos de enrutamiento de paquetes basados en prioridades, o los middleware especializados (tales como DDS) que son compatibles con requisitos de tiempo real estricto.

Para estas plataformas, el software de las aplicaciones convencionales de no tiempo real se desarrollan utilizando tecnología de despliegue basadas en componentes y particiones utilizando muy diferentes lenguajes de programación y tecnologías, pero así mismo existen lenguajes de programación (Ada 2012, RT-Java, y otras) específicas para el desarrollo de aplicaciones complejas de

tiempo real estrictas.

Los aspectos claves para la gestión de estos sistemas son la disponibilidad de modelos que permitan, caracterizar el comportamiento de las aplicaciones, su análisis para ser configuradas y su final configuración y despliegue. En este aspecto conviene destacar la reciente propuesta por la organización OMG del perfil UML denominado MARTE que estandariza el modelado del despliegue, del uso de los recursos, del comportamiento temporal y de la especificación de los requisitos temporales para aplicaciones de sistemas empujados, con la ventaja de que ofrece modelos complementarios, coherentes y homogéneos tanto para sistemas de tiempo real estrictos como para sistemas especificados mediante métricas de calidad de servicio. Así mismo, existen universidades y grupos de investigación que han desarrollado, herramientas de análisis de sistemas de tiempo real estrictos y tienen experiencia en el análisis de calidad de servicio utilizando modelos de colas, u otras metodologías, así como simuladores.

III. ANÁLISIS DE LAS HERRAMIENTAS PARA EL DISEÑO E IMPLEMENTACIÓN DE SISTEMAS DE CONTROL CONSIDERANDO EL TIEMPO REAL

El diseño de sistemas de control no es una tarea fácil y mucho menos cuando se deben considerar aspectos de tiempo, es decir, un sistema de control no solo se encarga de la obtención de datos, cálculos mediante algoritmos y verificación de estos, sino también que todas las tareas de control se realicen en el tiempo establecido obedeciendo plazos de tiempo determinado.

Por tanto, las tareas llevadas a cabo por los sistemas de control deberán ser ejecutadas en ‘tiempo real’ y permitir reaccionar a eventos externos que también suceden en “tiempo real”.

En ese sentido se presentará en los siguientes puntos aquellas herramientas que permitirán llevar a cabo con éxito tareas de control considerando tiempos críticos, las cuales de manera general se recurren a lo siguiente:

- Utilizando un lenguaje de programación de tiempo real con un hardware específico, por ejemplo un DSP, FPGA, PCIe.
- Utilizando un lenguaje de programación secuencial y/o un entorno de desarrollo gráfico dentro de un Sistema Operativo de Tiempo Real (RTOS en inglés).

Entonces, una plataforma para el diseño e implementación de sistemas de control atendiendo tiempos críticos requerirá básicamente los siguientes componentes:

- Un ambiente para el diseño y análisis de sistemas de control asistido por computador (DSCAC) [1].
- Una computadora con características de software de Tiempo Real (RTAI).
- Un paquete de desarrollo para objetivos específicos (compilador, enlazador, librerías).

- Un generador de código integrado en el ambiente DSCAC que pueda traducir un Diagrama de Bloques (programadas para las tareas de Control) a código de programa que será embebido en el dispositivo Hardware.
- Un conjunto de controladores (drivers) que interactúan con la tarjeta de adquisición de datos en la PC.
- Una aplicación para operar la tarea generada de tiempo real (monitoreo, ajuste de parámetros, inicio/parada).

En la actualidad existen versiones de distribución “libre y de código abierto” para los componentes mencionados anteriormente, entre las que se encuentran:

- **Scilab/Scicos**, es un excelente ambiente DSCAC, RTAI optimiza el sistema operativo para trabajar en Tiempo Real.
- **RTAI-Lab**, es un conjunto de herramientas (puntos 3, 4 y 6 de la lista anterior) que permite implementar el control diseñado.
- **Comedi** (punto 5), proporciona los controladores, librerías de funciones y una API (Interfaz de Programación de Aplicaciones) para interactuar con cientos de dispositivos de adquisición de datos [2].

Mientras que por el lado comercial con exquisitas características se encuentra el paquete de software MATLAB/Simulink que incorpora su propio elemento de generación de código, el Toolbox Realtime-Workshop (RTW).

Bajo ese análisis el usuario podrá realizar todas las etapas de Diseño de Sistemas de Control (ver Fig. 1) dentro un solo ambiente de Trabajo.



Figura 1: Etapas para el Diseño de Sistemas de Control considerando aspectos de Tiempo Real.

La mayoría de las etapas mostradas en la figura 1 son conocidas por los usuarios que diseñan Algoritmos de Control a diferencia que en este trabajo se presenta la incorporación de aspectos de Tiempo Real.

IV. AMBIENTE DE TRABAJO

Aquellas herramientas como, Linux - RTAI, RTAI-Lab, COMEDI y Scilab/Scicos que serán descritas a continuación conforman una ambiente de trabajo para el diseño e implementación de sistemas de control con tiempos críticos (ver Fig. 2). Una descripción detallada mostrando las etapas que se debieran seguir se encuentra en la Guía “Tutorial de RTAI” [1].

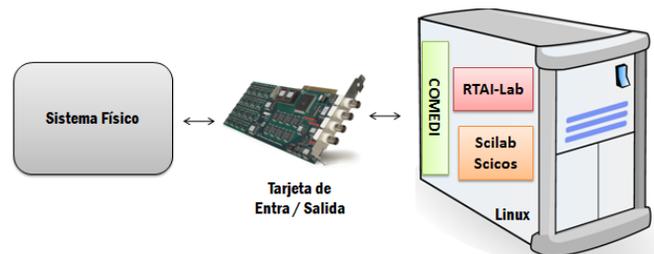


Figura 2: Ambiente de Trabajo para el Diseño de Sistemas de Control en Tiempo Real.

A. Sciloslab/Scicos

Scicoslab/Scicos [3] es una excelente opción como herramienta en cuanto a modelado, simulación, análisis y diseño de sistemas que involucran flujo de señales. Dicha herramienta ofrece casi todas las funcionalidades que proporciona MATLAB/Simulink, a diferencia de que Scicoslab es libre.

Adicionalmente Scicoslab cuenta con una serie de capacidades realizadas por investigadores en propósitos computacionales [4] y para el diseño de sistemas de control avanzado [5].

Por otro lado, Scicoslab/Scicos es usado en una amplia variedad de aplicaciones industriales e investigación, algunos de los proyectos que tuvieron su impacto fueron: La identificación de las fuentes de contaminación en ríos [6] y en la mejora de máquinas de fricción [7]. También ha sido ampliamente utilizado con fines educativos en los países industrialmente desarrollados [8], y en los países en desarrollo [9], [10] por caracterizarse como software altamente científico y “libre”.

B. RTAI (Interface para Aplicaciones en Tiempo Real)

Scicoslab es una excelente herramienta para diseño de sistemas de control pero este por sí mismo no puede interactuar con el “mundo real o físico”, es decir, Scicoslab carece de funcionalidades para manejar correctamente los componentes de software y hardware necesarios para interactuar con los sistemas físicos (por ejemplo montaje de

laboratorios). Sin embargo, la integración con otra herramienta de código abierto desarrollado por el "Politécnico de Milano (DIAPM)" [11], llamado RTAI [12], permite obtener una adecuada interacción con los sistemas físicos. Es así que RTAI es un ambiente para el desarrollo de sistemas de control digital y adquisición de datos de bajo costo.

RTAI se compone de un pequeño KERNEL de tiempo real bajo el kernel estándar de Linux y trata al kernel de Linux como una tarea de menor prioridad. RTAI además proporciona una amplia selección de mecanismos de comunicación entre procesos y otros servicios de tiempo real, adicionalmente proporciona un módulo llamado LXRT para facilitar el desarrollo de aplicaciones de tiempo real en el espacio de usuario (información recogida en [13]). Esta extensión es utilizada en diferentes proyectos y aplicaciones industriales, como por ejemplo:

- **Proyecto Orocós:** tiene como objetivo desarrollar un ambiente de control abierto para robots (manipuladores, robots móviles, humanoides, etc.) [14].
- **Proyecto ALMA** (Atacama Large Millimeter Array): donde un grupo de 16 nodos de Linux RTAI desarrolla el filtrado de datos, acondicionamiento de datos, transformada rápida de Fourier y correcciones de fase con una tasa de procesamiento de alrededor de 1 G FLOPS (operaciones de punto flotante por segundo) [15].

C. RTAI-Lab

El proyecto RTAI-Lab, desarrollado en el Departamento de Tecnología Innovativa de la SUPSI (Universidad de Ciencias Aplicadas de la Suiza Italiana), proporciona un conjunto de herramientas para el desarrollo de sistemas de control y monitoreo en tiempo real en diagramas de bloques los cuales pueden ser compilados y ejecutados en el sistema operativo Linux/RTAI en tiempo real [12]. Los diagramas de bloques se pueden desarrollar, ya sea usando Scicoslab/Scicos (versión libre) o Matlab/Simulink/RTW (versión comercial).

RTAI-Lab contiene las siguientes herramientas:

- **RTAI-Lib:** es un conjunto de bloques de Scicos, incluidos en el paquete RTAI, que permite diseñar diagramas de bloques con sensores y actuadores. Ésta provee una interfaz entre RTAI y el hardware de adquisición de señales. Los diagramas de bloques que usa RTAI-Lab pueden ser compilados dentro del software Scicoslab/Scicos.
- **Xrtailab:** es una herramienta que permite el monitoreo y ajuste de tareas en tiempo real desde cualquier computadora en una red local LAN (Local Area Network), la cual forma también parte de RTAI.

En este sentido, el proyecto de código abierto RTAI-Lab tiene como objetivo proporcionar un marco de es-

tructura común para la integración de RTAI en el ambiente Scicoslab ([16], [17]). Con RTAI-Lab y Scicoslab es posible obtener un completo entorno de código abierto para el diseño de sistemas de control y su prueba en los sistemas físicos reales.

D. MATLAB/Simulink RTW

Como herramienta paralela a lo presentado anteriormente se encuentra el potente Matlab/Simulink/RTW [18] con sus librerías altamente desarrolladas para cada una de las aplicaciones en particular, sin embargo hay que señalar que las licencias para Matlab son muy caras, en particular, la licencia para RTW. Esta licencia es necesaria si el usuario desea cambiar parámetros y señales del monitor utilizando el modo externo RTW.

Dicha herramienta cuenta con una interfaz gráfica que brinda la posibilidad de crear código en lenguaje C usando el toolbox RTW. La generación de código puede ser fácilmente aplicada y cargada en los diferentes dispositivos electrónicos. La implementación que se cuenta en el laboratorio de mecatrónica de la SUPSI trabaja con NEWLXRT [19] y la integración usando librerías brindadas por COMEDI.

Una de las ventajas de esta herramienta es la integración de sensores y actuadores en el diagrama de diseño de control realizada en Simulink. Dicha integración son directamente realizadas con C-MEX y S-Functions dentro de la programación para la generación de código. Seguidamente no se realiza más tareas más que la ejecución del programa.

V. CONCLUSIONES Y TRABAJO FUTURO

En este documento se describe los aspectos necesarios para llevar a cabo con éxitos algoritmos de control considerando aspectos de tiempo real bajo herramientas libres. El uso de un Sistema Operativo Linux con el parche de tiempo real RTAI, que a su vez es compatible con lenguajes de programación de alto nivel (Matlab/Simulink, Scicoslab/Scicos) e interfaces E/S convencionales (FPGA, PCIe, PLC) permite la creación y validación de tareas de Control o algoritmos de Control conforme a las necesidades de diseño de una forma fácil, rápida y flexible.

Bajo este enfoque los siguientes trabajos a considerar serán aquellos que además de la necesidad de considerar tiempos críticos, serán aquellos Sistemas de Control Distribuidos donde se tendrá una red de computadoras trabajando en Tiempo Real cada una.

REFERENCIAS

- [1] S. Mannori R. Bucher y T. Netter. *RTAI-Lab tutorial: Scicoslab, Comedi, and real-time control*. 2010. URL: <http://robertobucher.dti.supsi.ch/bucherdl/RTAI-Lab-tutorial.pdf> (vid. págs. 2, 3).

- [2] COMEDI. 2014. URL: <http://www.comedi.org/> (vid. pág. 3).
- [3] Scicoslab. 2014. URL: <http://scicoslab.org/> (vid. pág. 3).
- [4] I. Pendharkar. “Rtool for scilab: A public domain tool for SISO system design”. En: *IEEE Control System Magazine*. Vol. 25. 2005, págs. 23-25 (vid. pág. 3).
- [5] F. Delebecque. “Rtool for scilab: A public domain tool for SISO system design”. En: *IEEE Control System Magazine*. 2000 (vid. pág. 3).
- [6] M. Maldiney J. Chancelier M. Cohen y F. Pacard. “Identification of pollution sources in rivers”. En: *IEEE International Symposium on Computer-Aided Control System Design*. 1996 (vid. pág. 3).
- [7] Z. Dongjie Q. Dongping y T. Qiang. “Linux/rtai and scicos in low cost high performance friction testing machine”. En: *International Conference on Electronic Measurement and Instruments*. 2007 (vid. pág. 3).
- [8] P. Tona. “Teaching process control with scilab and scicos”. American Control Conference”. En: 2006 (vid. pág. 3).
- [9] J. Ramirez y M. Romero. “Scilab as a tool to increase learning in courses of communication systems”. En: *Electronics, Robotics and Automotive Mechanics Conference*. 2008 (vid. pág. 3).
- [10] P. Pires y D. Rogers. “Free/open source software: An alternative for engineering students”. En: *Frontiers in Education*. 2002 (vid. pág. 3).
- [11] L. Dozio y P. Mantegazza. “Linux real time application interface (RTAI) in low cost high performance motion control”. En: *Motion Control Conference*. 2003 (vid. pág. 4).
- [12] RTAI. 2014. URL: <https://www.rtai.org/>. (vid. pág. 4).
- [13] Pasi Sarolahti. “Real-Time Application Interface”. En: *Research seminar on Real-Time Linux and Java*. 2001 (vid. pág. 4).
- [14] *Open Realtime Control versus Open Robot Control*. 2014. URL: <http://www.orocos.org> (vid. pág. 4).
- [15] *Atacama Large Millimeter Array*. 2014. URL: <http://legacy.nrao.edu/alma.shtml> (vid. pág. 4).
- [16] R. Bucher y S. Balemi. “Rapid control prototyping using linux real-time”. En: *Conference in Control Education*. Olulu - Finland, 2003 (vid. pág. 4).
- [17] R. Bucher y S. Balemi. “Scilab/scicos and linux rtai - an unified approach”. En: *IEEE Conference on Control Applications*. 2005 (vid. pág. 4).
- [18] *Mathworks*. 2014. URL: <http://www.mathworks.com> (vid. pág. 4).
- [19] P. Mantegazza G. Quaranta. “Using MATLAB-Simulink RTW to Build Real Time Control Applications in User Space with RTAILXRT”. En: *Real-time Linux Workshop*. Milano - Italia, 2001 (vid. pág. 4).
- [20] R. Bucher y L. Dozio. “CACSD under RTAI Linux with RTAI-LAB”. En: *OSADL Project: Real Time Linux Workshops*. España, 2003.