

CONTROL EN TIEMPO REAL Y SUPERVISIÓN DE PROCESOS MEDIANTE SERVIDORES OPC-UA

Francisco Blanes Noguera
pblanes@ai2.upv.es
Andrés Benlloch Faus
anbenfau@ai2.upv.es

Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de Valencia

Resumen

El control de procesos junto con la supervisión de los mismos son casos de sistemas con restricciones temporales de carácter diferente. Mientras el control requiere el cumplimiento de restricciones de forma estricta habitualmente, la supervisión permite un tratamiento más relajado de las condiciones. Esta naturaleza tan distinta de la problemática, hace que su combinación en el mismo sistema hardware sea compleja. [1]

El presente experimento aborda la problemática de controlar y supervisar el estado de una planta piloto con un elemento de posicionamiento crítico en su control. Para ello se han experimentado con diversas soluciones HW-SW valorando sus ventajas e inconvenientes.

Palabras Clave: sistema empotrado, control en tiempo real, supervisión de procesos, protocolo OPC-UA

1 INTRODUCCIÓN

El presente experimento tiene como objetivo llevar a cabo el control de un sistema de posicionamiento tridimensional crítico encargado de la recolección y posterior ubicación de piezas a lo largo de una cinta transportadora. Dicho control se deberá llevar a cabo de forma rápida y precisa, ya que la más mínima variación en la posición en cualquiera de los tres ejes de la grúa provocará una mala operación de la línea. La planta piloto que vamos a controlar durante el experimento consta de una grúa compuesta de un sistema de succión formado por un compresor y unas válvulas de vacío, este sistema es el encargado de llevar a cabo el agarre de las piezas durante el proceso de transporte, 6 encoders encargados de contabilizar los pasos de los motores así como indicar la orientación del giro y 3 motores encargados de llevar a cabo el posicionamiento de la grúa. Los motores permiten llevar a cabo movimientos en el eje vertical, el eje horizontal y rotaciones horarias y anti horarias.

En una primera instancia se planteó llevar a cabo el control del proceso de posicionamiento de la grúa a través de las entradas y salidas digitales de un PLC, sin embargo, nos encontramos con la problemática de que la frecuencia de muestreo de las entradas digitales del PLC era mucho menor que la frecuencia a la que los encoders enviaban sus pulsos, el desfase existente entre las frecuencias provocaba la pérdida de pasos de los encoders, lo que se traducía en un posicionamiento erróneo de la grúa.

Dado que la frecuencia de muestreo del módulo entradas digitales del PLC no se podía aumentar, se optó reducir la frecuencia de las pulsos que enviaba el encoder al PLC, para ello se diseñó un sistema de división de pulsos virtual mediante un sistema empotrado programable, concretamente un Arduino Mega 2560, que permitía, gracias a la frecuencia de su reloj, muestrear todos los pulsos enviados por los encoders, contabilizarlos y enviar pulsos a las entradas digitales del PLC dentro de una frecuencia de muestreo válida para que el PLC fuese capaz de detectar las señales recibidas y poder llevar a cabo un posicionamiento correcto de la grúa. Esta solución resultó válida, sin embargo, no era óptima, ya que al dividir los pulsos del encoder perdíamos precisión en el posicionamiento, lo que producía que en ocasiones fuésemos incapaces de colocar la grúa en las posiciones apropiadas.

Para solucionar esta problemática se decidió cambiar el PLC encargado de controlar el proceso de posicionamiento de la grúa y el divisor de pulsos virtual por un sistema empotrado NI myRIO capaz de muestrear correctamente los pasos de los encoders y manejar la grúa, permitiéndonos aplicar un sistema de control en tiempo real válido sobre la misma. Esta solución resultó ser la más apropiada para aplicar un control sobre el proceso, ya que permitía posicionar de forma precisa la grúa, considerando únicamente el error proveniente de los problemas mecánicos existentes.

1.1 Motivación

En anteriores experimentos, no relacionados directamente con el que actualmente abordamos, se

llevaron a cabo pruebas con los sistemas industriales utilizados, y se determinó que las entradas digitales del PLC, al no tratarse de entradas de alta velocidad, eran incapaces de detectar todos los pulsos enviados por los encoders de la grúa, lo que dificultaba el posicionamiento de la grúa, provocando fallos en el funcionamiento del proceso industrial que estábamos simulando.

Para solucionar dicha problemática, y ser capaces de llevar a cabo un control eficiente y preciso de la grúa, se decidió implementar dos posibles soluciones, la primera de ellas consiste en llevar a cabo el diseño y desarrollo de un divisor de pulsos virtual mediante un sistema empotrado programable Arduino Mega 2560. Este sistema será el encargado de llevar a cabo la cuenta de los pulsos de los tres encoders de la grúa y reducir su frecuencia para que las entradas digitales del PLC sean capaces de monitorizar los pasos del encoder.

La segunda solución, consisten en llevar a cabo la implementación del sistema de control sobre un sistema empotrado NI myRIO, el cual, gracias a sus prestaciones, nos permite llevar a cabo la monitorización de las salidas digitales de la grúa sin perder ningún pulso y a su vez llevar a cabo el manejo de la misma a través de sus entradas, permitiéndonos así implementar un sistema de control capaz de posicionar de forma precisa la grúa sin necesidad de utilizar sistemas de apoyo.

1.2 Objetivos

Los objetivos que se pretenden alcanzar al finalizar el desarrollo de este trabajo son:

1. Evaluar las características de las soluciones planteadas, desde el punto de vista de sistemas de control en tiempo real.
2. Diseñar e implementar un divisor de pulsos virtual, que permita a las entradas digitales del PLC muestrear los pulsos enviados por los encoders de la grúa y así poder posicionar de la forma más precisa posible cada uno de los tres ejes.
3. Diseñar e implementar un sistema de control sobre un sistema empotrado NI myRIO que nos permita monitorizar los encoders de la grúa y posicionarla de forma precisa. Esta implementación busca demostrar que un sistema empotrado de estas características es capaz de llevar a cabo un control en tiempo real de un sistema industrial crítico.
4. Desplegar un servidor OPC-UA sobre un sistema empotrado NI myRIO, con el objetivo de combinar en el mismo HW control en tiempo real y servicios de supervisión de alto nivel mediante protocolo OPC-UA.

5. Llevar a cabo la interconexión de diversos sistemas OPC-UA y monitorizar la grúa desde diferentes equipos.

2 EQUIPOS

Para llevar a cabo el control y posicionamiento de la grúa ha sido necesario utilizar diversos sistemas de control que permitiesen manejarla de forma precisa. Los elementos hardware utilizados en el experimento fueron:

2.1 PLC OMRON serie NJ

El PLC utilizado en los casos de uso del presente experimento es el modelo OMRON de la familia NJ1. Estos equipos son los utilizados a nivel industrial para llevar a cabo el control de procesos en tiempo real. Dichos sistemas disponen de módulos para añadir funcionalidad a la CPU, la cual se encarga de monitorizarlos y controlarlos. En el presente experimento, dado que las entradas y salidas del proceso a controlar son digitales se optó por la colocación de un módulo de entradas/salidas digitales que permitiese comunicar el PLC con el proceso y llevara a cabo el control del mismo.

2.1.1 Módulo E/S digitales MD232

El módulo de entradas/salidas digitales utilizado para llevar a cabo la comunicación entre el PLC y el proceso a controlar es el modelo MD232, dispone de un total de 16 entradas digitales y 16 salidas digitales, las cuales resultan suficientes para llevar a cabo el control del proceso de posicionamiento de la grúa.

2.2 Arduino Mega

El Arduino Mega es una placa basada en el microcontrolador ATmega2560 de Atmel que dispone de 54 pines I/O digitales, de los cuales 15 pueden utilizarse a su vez de salidas PWM, 16 entradas analógicas, 4 UARTs un reloj de 16 Mhz de frecuencia [3].

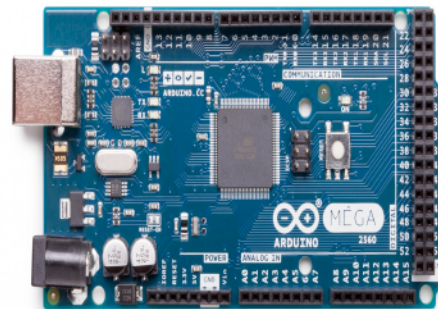


Figura 1: Arduino MEGA 2560.

Gracias a sus pines programables y a la frecuencia de reloj de 16 Mhz se convierte en un sistema válido para implementar un divisor de pulsos apropiado para el caso que nos ocupa. Se decidió utilizar un sistema programable para llevar a cabo la implementación del divisor de pulsos, y no el diseño de un sistema electrónico, para poder ajustar de forma dinámica y simple la frecuencia de división de la señal de los encoders.

2.3 NI myRIO

Se trata de un sistema educativo de tiempo real basado en la arquitectura ARM Cortex-A9 dual-core con entradas/salidas programables de FPGA Xilinx. El sistema empotrado está formado por 10 salidas analógicas, 6 salidas digitales y 40 líneas de E/S digital. Además, dispone de conectividad alámbrica e inalámbrica lo que facilita su instalación en medios industriales [2].



Figura 2: NI myRIO.

En el experimento que nos ocupa el sistema empotrado se programó para llevar a cabo tanto para la adquisición de datos como para la generación de señales de control.

3 DISEÑO DE LA SOLUCIÓN

La problemática que nos ocupa en el presente experimento surgió tras intentar llevar a cabo el control de un proceso de posicionamiento preciso mediante un módulo de entradas/salidas digitales sin entradas rápidas de un PLC. Tras varias pruebas determinamos que el módulo era incapaz de detectar gran parte de los pulsos enviados por los encoders del proceso, provocando así errores en el posicionamiento de la grúa.

Dicha problemática motivó el diseño de un sistema de división de pulsos virtual con el objetivo de reducir la frecuencia de los pulsos que llegan al módulo de entradas digitales del PLC y permitir a este detectar los pasos de los encoders de posicionamiento de la grúa. El sistema de división de pulsos virtual está formado por un sistema empotrado

Arduino Mega 2560 y tres relés, a través de los cuales el sistema empotrado envía las señales de pulso de los encoders dividida a las entradas digitales del PLC.

Esta solución atenuó de forma considerable el error de posicionamiento existente cuando conectábamos directamente las salidas de los encoders a las entradas digitales del PLC, sin embargo, seguía existiendo un pequeño error en el posicionamiento de la grúa que producía que en ocasiones el funcionamiento de la grúa no fuese el correcto produciendo fallos en el proceso. Por este motivo, dado que el posicionamiento de la grúa resulta crítico para realizar correctamente las labores de ubicación de piezas, se decidió abordar una segunda solución para atenuar aún más el error de posicionamiento existente cuando utilizamos el divisor de pulsos virtual.

Esta segunda solución consiste en la utilización de un sistema empotrado myRIO de National Instruments para llevar a cabo el control de la grúa en lugar de un PLC vinculado a un divisor de pulsos virtual. Gracias a las prestaciones que proporciona dicho sistema empotrado, en esta ocasión las salidas digitales de los encoders pueden conectarse directamente a las entradas digitales del sistema de control sin miedo a perder ningún pulso de cualquiera de sus tres ejes de acción. Además, gracias a sus salidas digitales nos permite actuar directamente sobre las entradas de la grúa, permitiéndonos controlar el proceso de producción directamente desde el sistema empotrado.

3.1 Divisor de pulsos virtual con Arduino Mega 2560

La finalidad del divisor de pulsos virtual es la de reducir N veces la frecuencia de las señales que envían los tres encoders de posicionamiento de la grúa para que la frecuencia de muestreo de las entradas digitales del módulo de entradas/salidas del PLC sea capaz de detectar cambios de estados y contabilizar los pulsos del encoder.

Para ello el divisor de pulsos virtual consta de un sistema empotrado Arduino Mega 2560 programable, con una frecuencia de reloj de 16 Mhz capaz de detectar los pulsos que envía cada uno de los encoders a través de sus entradas digitales y tres relés que enviarán pulsos a las entradas digitales del PLC simulando los pulsos de los encoders.

Para ello se programaron tres contadores dentro del sistema empotrado vinculados a las señales de los encoders y a las salidas que activan los relés. De esta forma cada vez que el sistema empotrado contabilizaba un número determinado de pulsos por cada uno de los encoders de la grúa enviaba un pulso al PLC mediante la activación y posterior desactivación de los relés.

Sin embargo, este sistema presenta un inconveniente importante a tener en cuenta a la hora de posicionar

la grúa y es que para que las entradas digitales del PLC sean capaces de muestrear el paso del encoder es necesario dividir la señal del encoder en un orden de entre 8 y 10 veces, lo que provoca que se pierda precisión durante el posicionamiento de la grúa.

3.2 Control del proceso de posicionamiento de una grúa mediante NI myRIO

Debido a los errores de posicionamiento detectados cuando llevábamos a cabo el control de la grúa mediante el PLC, decidimos llevar a cabo la implementación de un sistema de control dentro de un sistema de tiempo real como es NI myRIO.

Al disponer dicho sistema de entradas y salidas digitales somos capaces de monitorizar todas las señales que envía la grúa sin necesidad de utilizar para ello divisores de pulsos, así como manejar esta mediante las salidas digitales del sistema empotrado. Para llevar a cabo el control del proceso de posicionamiento de la grúa se implementó una máquina de estados que simulase el comportamiento del graficet implementado en el PLC y así replicar el mismo sistema de control en ambos casos y poder realizar comparaciones entre ellos a posteriori.

4 EVALUACIÓN Y CASOS DE USO

Para testear que los sistemas que se ha implementado cumplía con los objetivos establecidos, se tomó la decisión de diseñar una serie de casos de uso que se describen con detalle en las siguientes subsecciones. Durante los diferentes casos de uso se utilizó un PLC Omron serie NJ, un sistema empotrado Arduino Mega, un sistema emportado NI myRIO y una maqueta Fishertechnik modelo "Vacuum Gripper Robot 24V" como la de la Figura 3.

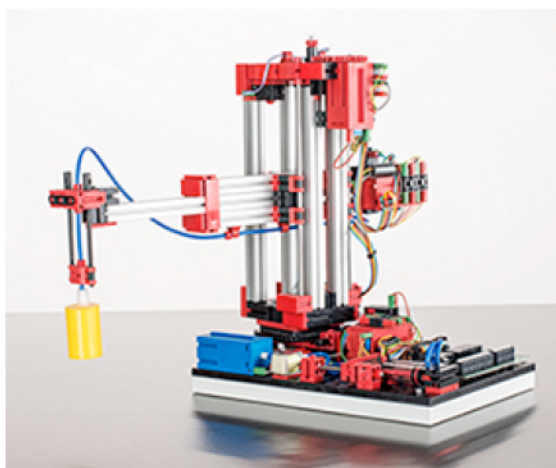


Figura 3: Maqueta Fishertechnik

4.1 Control de un proceso de posicionamiento crítico mediante PLC

El objetivo que se busca conseguir en este caso de uso es demostrar que el módulo de entradas/salidas digitales del PLC es incapaz de muestrear de forma correcta los pulsos enviados por los 3 encoders (Rotación, Avance/Retroceso y Subida/Bajada) de los que dispone la grúa para su posicionamiento.

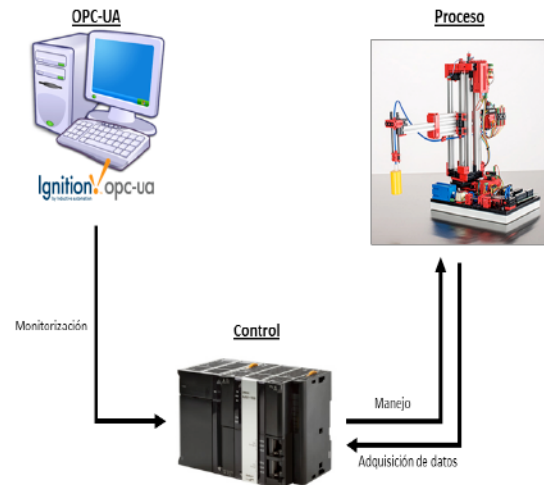


Figura 4: Esquema de la solución control mediante PLC.

1. Configuramos las entradas y salidas digitales del módulo MD232 en el PLC.
2. Conectamos las entradas y salidas digitales del PLC a las entradas y salidas digitales del proceso a controlar.
3. Cargamos en el PLC el programa graficet diseñado para controlar el proceso.
4. Ponemos en marcha el proceso en repetidas ocasiones y medimos las posiciones de la grúa para detectar los errores que se producen en el posicionamiento de la misma a causa de la baja frecuencia de muestreo de las entradas digitales del PLC.

4.2 Control de un proceso de posicionamiento crítico mediante PLC y divisor de pulsos virtual

Dado que en el caso de uso anterior demostramos que únicamente con la frecuencia de muestreo de las entradas y salidas digitales del PLC somos incapaces de detectar todos los pasos de los encoders, en este caso de uso se añade un divisor de pulsos entre el proceso y las entradas digitales del PLC para adecuar las señales enviadas a la frecuencia de muestreo del PLC, y así llegar a controlar de forma más eficiente el posicionamiento de la grúa.

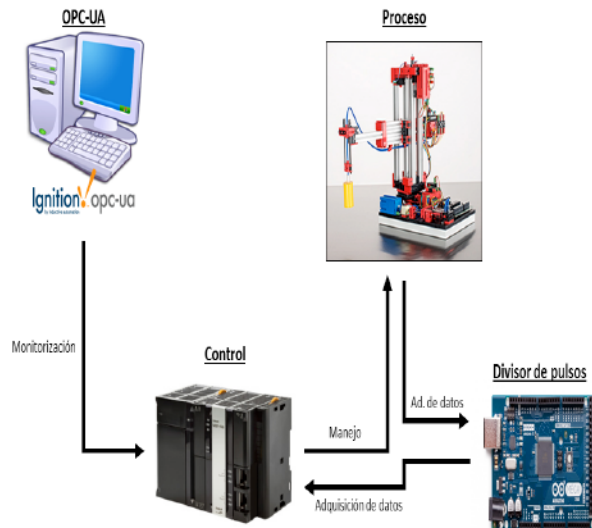


Figura 5: Esquema de la solución control mediante PLC y divisor de pulsos.

1. Configuramos las entradas y salidas digitales del módulo MD232 en el PLC.
2. Conectamos las salidas digitales del PLC a las entradas digitales del proceso.
3. Conectamos las salidas digitales del proceso al divisor de pulsos virtual implementado.
4. Conectamos las salidas del divisor de pulsos virtual a las entradas digitales del PLC
5. Cargamos en el PLC el programa grafcet diseñado para controlar el proceso.
6. Ponemos en marcha el proceso en repetidas ocasiones y medimos las posiciones de la grúa para detectar los errores que se producen en el posicionamiento de la misma.

Tras haber desplegado todo el sistema antes descrito, procedemos a comprobar si somos capaces de actuar sobre el proceso industrial sin errores de posicionamiento como si ocurría en el caso de uso anterior.

4.3 Control de un proceso de posicionamiento crítico mediante un sistema empujado NI myRIO

Dado que en el caso de uso anterior se consiguió demostrar que mediante la utilización de un divisor de pulsos éramos capaces de muestrear de forma correcta los pasos del encoder pero, sin embargo, debido a la división de pulsos perdíamos precisión en el posicionamiento de la grúa, en este caso de uso se opta por implementar el control del proceso dentro de un sistema empujado NI myRIO, que gracias a sus características permite muestrear los pasos de los encoders sin necesidad de utilizar para ello un divisor

de pulsos, y además, gracias a sus salidas digitales manejar el proceso.

1. Configuramos las entradas y salidas digitales del sistema empujado NI myRIO.
2. Implementamos una máquina de estados dentro del sistema empujado para simular el grafcet implementado en el PLC
3. Conectamos las entradas y salidas digitales del proceso a las entradas y salidas digitales de NI myRIO.
4. Ponemos en marcha el proceso en repetidas ocasiones y medimos las posiciones de la grúa para detectar los errores que se producen en el posicionamiento de la misma.

En este caso de uso, dado que la frecuencia de muestreo de NI myRIO si permite detectar todos los pasos de los encoders, y además tenemos el control del proceso en el mismo equipo, somos capaces de actuar en tiempo real sobre el proceso y llevar a cabo un posicionamiento más preciso que en los anteriores casos de uso.

4.4 Despliegue de un servidor OPC-UA sobre un sistema empujado NI myRIO

Dado que la funcionalidad del presente experimento está orientada al control de un proceso industrial, y que actualmente a nivel industrial se está abogando cada vez más por la conectividad entre equipos y dispositivos desplegados a lo largo de una fábrica, decidimos desplegar un servidor OPC-UA que permitiese a cualquier cliente o servidor conectarse al sistema empujado NI myRIO y acceder a los datos que en él se almacenan [4].

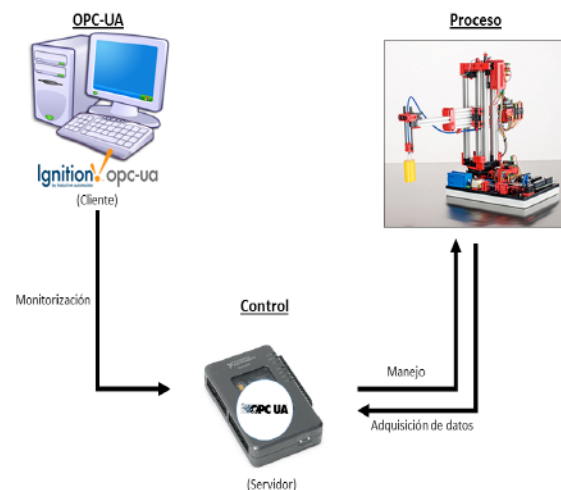


Figura 6: Esquema de la solución control mediante NI myRIO con servidor OPC-UA.

1. Implementación de un servidor OPC-UA en el sistema empotrado NI myRIO mediante los objetos OPC de los que dispone el sistema.
2. Vinculamos las variables a los Tags del servidor OPC-UA.
3. Ponemos en marcha el servidor OPC-UA dentro del sistema empotrado NI myRIO.

Después de haber desplegado todo el sistema antes enumerado, procedimos a comprobar si éramos capaces de acceder a los datos habilitados en el servidor OPC-UA a través de un cliente OPC-UA. Gracias a la capacidad de comunicarnos con el sistema empotrado a través de OPC-UA somos capaces de monitorizar y manejar el proceso controlado por NI myRIO gracias a la escritura de valores en las variables del árbol de Tags.

4.5 Interconexión de servidores OPC-UA desplegados en diferentes arquitecturas

Dado que para testear que el despliegue del servidor OPC-UA sobre el sistema empotrado NI myRIO se había realizado correctamente precisábamos de un cliente OPC-UA, decidimos llevar a cabo la prueba que aborda el actual caso de uso, que consiste en llevar a cabo la inclusión del servidor OPC-UA dentro de un servidor OPC-UA Ignition desplegado sobre un PC

1. Preparamos el servidor OPC-UA dentro del sistema empotrado NI myRIO.
2. Desplegamos el servidor OPC-UA de Ignition sobre un PC.
3. Comunicamos ambos servidores a través del protocolo de comunicaciones OPC-UA.
4. Visualizamos el árbol de Tags del servidor OPC-UA de myRIO a través del cliente OPC-UA que incorpora Ignition.
5. Realizamos operaciones de lectura y escritura sobre las variables de NI myRIO a través del cliente OPC-UA de Ignition.

Tras realizar las pruebas anteriormente enumeradas se pudo concluir que gracias al protocolo OPC-UA éramos capaces de llevar a cabo la monitorización y el manejo del proceso industrial mediante lecturas y escrituras de las variables del sistema empotrado NI myRIO desde el cliente OPC-UA desplegado en el PC.

6 CONCLUSIONES

Tras llevar a cabo los casos de uso antes expuestos, las conclusiones que obtuvimos fueron:

- El sistema de control implementado en el PLC con un módulo de entradas/salidas digitales no era suficiente para poder

controlar de forma eficiente el posicionamiento de la grúa, ya que la frecuencia de muestreo de las entradas digitales del PLC era menor que la frecuencia de las señales de los encoders, por lo tanto, éramos incapaces de detectar todos pasos, lo que provocaba errores en el posicionamiento de la grúa.

- El sistema de división de pulsos virtual resulto de utilidad, ya que al reducir la frecuencia de las señales de los encoders el PLC era capaz de contabilizar los pasos sin perdida alguna de señales, sin embargo, como consecuencia de la división de pulsos, se perdió precisión en el posicionamiento de la grúa.
- El sistema empotrado de tiempo real NI myRIO resulto ser la solución idónea para el experimento que nos ocupa, ya que gracias a sus prestaciones y a sus entradas y salidas digitales, fuimos capaces de llevar a cabo un posicionamiento preciso, sin apenas errores, de la grúa. Se determinó que el error de posicionamiento detectado en ocasión estaba provocado por errores mecánicos de la grúa, ya que al tratarse de una maqueta los engranajes patinaban provocando que la grúa se detuviese unos pasos por delante de la posición establecida.
- El control del proceso mediante NI myRIO ofrece como ventajas frente al control del proceso mediante PLC una mayor versatilidad al programar el sistema y llevar a cabo el control del proceso gracias la utilización de una FPGA, unos tiempos de respuesta más cortos que el módulo de E/S digitales estándar del PLC y un coste económico menor, ya que para conseguir llevar a cabo el control del proceso mediante el PLC se precisa de un módulo específico de E/S digitales rápidas, cuyo coste sin la CPU del PLC ya supera el coste de sistema empotrado NI myRIO.

Referencias

- [1] Burns, Wellings: **Real-Time Systems and programming Languages**
- [2] N. Instruments, **MyRio. National Instruments, 2015.**
- [3] Website: <https://www.arduino.cc/en/Main/arduinoBoardMega>
- [4] Wolfgang Mahnke, Stefan-Helmut Leitner, **OPC Unified Architecture, 1ª Edición, 2009.**