

Desarrollo e integración de un sistema ciber-físico aplicado al control de calidad utilizando visión artificial.

Fernando Israel Jácome Ramírez.

Máster Universitario en Ingeniería de Sistemas Automáticos y Electrónica Industrial.

Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú.

Universidad Politécnica de Cataluña.

ferisdra@hotmail.com.

Resumen

Este proyecto pretende diseñar e implementar un prototipo de sistema ciber-físico como los utilizados en la industria. Dicho prototipo se enmarcará en el ámbito académico y estará enfocado al control de calidad de engranajes mediante la utilización de visión artificial.

El proyecto se dividirá en dos partes. El diseño y construcción de un sistema físico a escala que procesa mecánicamente los engranajes y la implementación de una aplicación web, que permitirá controlar, monitorizar y visualizar los resultados del procesamiento de los engranajes del sistema físico.

Para la consecución de estos objetivos, se ha construido un prototipo conformado por un sistema de banda de transporte, sensores, actuadores y una cámara para implementar un sistema de clasificación de engranajes por visión, además del uso de dispositivos microprocesadores y microcontroladores de bajo coste.

Este proyecto se ha implementado sobre un único protocolo de comunicación TCP, conocido como Servidor - Cliente, gracias a la ayuda de un entorno de ejecución para documentos JavaScript como lo es Node.js con sus respectivas librerías socket.io, facilitando el desarrollo de aplicaciones Websockets en el cliente y en el servidor.

Se ha desarrollado un aplicativo web bajo el entorno de desarrollo HTML5 y JavaScript, que permite el control y monitorización del sistema físico. La aplicación también permite obtener información de los resultados del procesado de los engranajes que han sido guardados en un historial.

El resultado final ha sido la construcción de un sistema físico a escala y el desarrollo del software para el control

y selección de engranajes. Finalmente se ha trabajado en una guía práctica para que los alumnos puedan interactuar con el sistema ciber-físico en un entorno acotado. De este modo se pretende contribuir al aprendizaje de este concepto tecnológico desde un punto de vista eminentemente práctico.

I. Introducción

La Industria 4.0 está incorporando las tecnologías digitales a través de los conocidos “habilitadores digitales”, cuyo objetivo principal es potenciar al máximo las posibilidades que la nueva industria ofrece gracias al IoT y los sistemas ciber-físicos. El Big Data, la inteligencia artificial, el machine learning, las últimas tecnologías digitales de comunicación, algoritmos y robótica se unen para dar lugar a los sistemas ciber-físicos, que combinan sistemas físicos con software de última generación y que van a situarse en el epicentro de este cambio industrial. [1]

Los CPS (sistemas ciber-físicos) integran capacidades de computación, almacenamiento y comunicación con capacidades de seguimiento y/o control de objetos en el mundo físico. Normalmente están conectados entre sí y en algunos casos disponen de capacidad de aprender y evolucionar. [2]

Establecidas la importancia que hoy en día van adquiriendo los sistemas enfocados en el internet de las cosas dentro del ámbito industrial, este trabajo desarrolla conceptos puntuales para comprender, desarrollar e implementar dentro de un sistema a nivel académico un sistema Ciber-físico, con el fin de entender como dispositivos físicos se integran a capacidades computacionales y todo ello con dispositivos al alcance de un estudiante, como son el Raspberry pi y el Arduino.

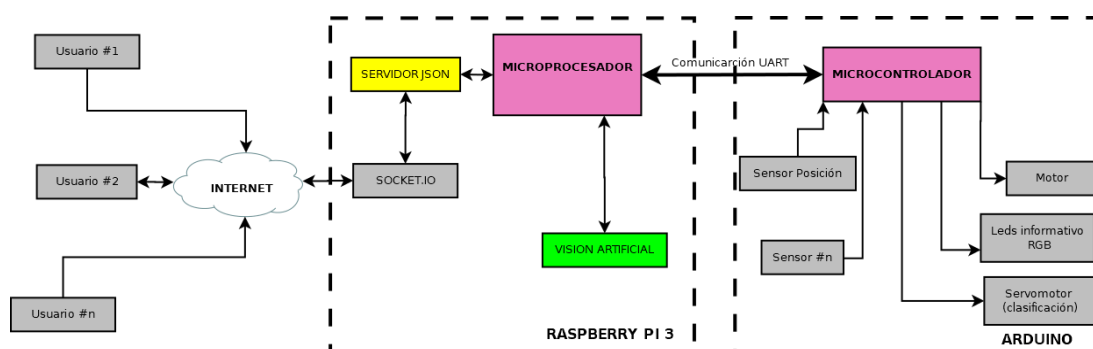


Fig. 1 Arquitectura CPS implementado

Para ello se desarrollará un prototipo denominado SECVIA (Selección de Engranajes Con Visión Artificial). SECVIA es una plataforma de aprendizaje mediante un sistema de banda transportadora a escala y enfocado al control de calidad mediante la utilización de visión artificial. En la industria y dentro del área de producción, su función principal es el control de calidad, asegurando que los productos cumplan con los requerimientos mínimos predefinidos de calidad. Este prototipo tendrá como objetivo verificar si un engranaje cuenta con los dientes correspondientes, así como datos técnicos importantes como: Diámetro exterior (De), Diámetro interior (Di), Número de dientes (z), Módulo(M) y paso (P) de los dientes.

Este documento presenta el desarrollo y los resultados obtenidos a partir del prototipo SECVIA y del aplicativo web, con los que se logró simular y validar el correcto funcionamiento en tiempo real.

El contenido del presente resumen se organiza de la siguiente manera. La sección II, describe el desarrollo y la implementación del CPS SECVIA, e incluye la descripción de los subsistemas físico, electrónico y software. La Sección III, muestra los resultados de la implementación del prototipo y del aplicativo web. La sección IV, contiene la elaboración de una guía práctica que ayude a los estudiantes a comprender un CPS desde un punto de vista interactivo. Finalmente, la sección V, presenta las conclusiones y las posibles líneas futuras de continuación de este trabajo.

II. Desarrollo CPS SECVIA.

Subsistema Físico.

El prototipo SECVIA, ver figura 2, es una estructura que sustenta una cinta transportadora y una cámara de visión. Dispone de un motor DC a 12 V para otorgar el correspondiente movimiento a la cinta de transporte de los engranajes. También incluye sensores de posición que facilitarán la detección del engranaje en proceso. Cuando el engranaje haya sido detectado y ubicado mediante una cámara con tecnología de visión, capturará la imagen para luego ser procesada. Con los resultados del procesado, un servomotor de clasificación permitirá distinguir engranajes buenos y malos.

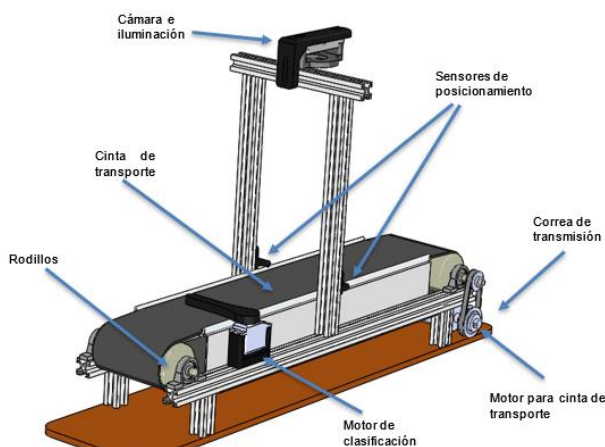


Fig. 2 Sistema físico de SECVIA

Subsistema electrónico.

El subsistema electrónico se basa en uso de un microprocesador y un microcontrolador, comunicados por un puerto serial de forma bidireccional. Concretamente se han considerado un Arduino UNO y una Raspberry Pi.

En la placa Raspberry Pi, se programan los diferentes bloques que permiten desarrollar el entorno Servidor-Cliente. También incorporará el algoritmo para el procesamiento de los engranajes, y el sistema de clasificación.

Por otra parte, la placa Arduino UNO, recibe órdenes de la placa Raspberry Pi y solo interacciona con sensores y actuadores del prototipo.

En la figura 3, se muestra el diagrama de bloques de los diferentes elementos que conforman el diseño electrónico del prototipo SECVIA.

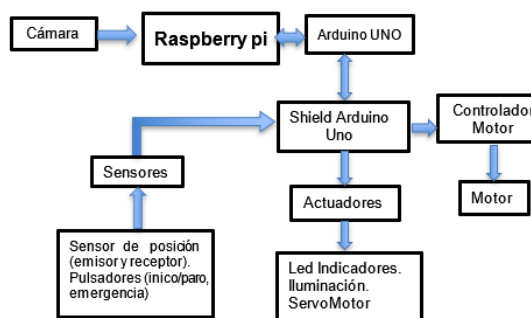


Fig.3 Interconexiones del prototipo SECVIA.

Subsistema Software.

Sistema Operativo.

Se utilizará Raspbian (parte del sistema operativo GNU/Linux), para trabajar con la placa Raspberry Pi. La instalación de este sistema operativo deberá contar con una tarjeta SD, de al menos, 8 GB.

Servidor - Cliente.

Un servidor es una aplicación que ofrece un servicio a usuarios de Internet; un cliente es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas.

Los usuarios invocan la parte cliente de la aplicación, que construye una solicitud para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte. Generalmente, un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo. [3]

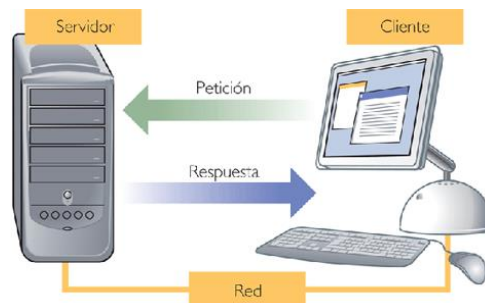


Fig. 4 Comunicación servidor - cliente [4]

Websocket.

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede ser utilizada por cualquier aplicación cliente/servidor. [5]

Para lograr el canal de comunicación bidireccional se utiliza un entorno en tiempo de ejecución multiplataforma. Este entorno se denomina Node.js. Al mismo tiempo, para la implementación de aplicaciones web en tiempo real se requirió añadir una biblioteca de JavaScript denominada socket.io.

Node.js es de código abierto para la capa del servidor, basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y está basado en el motor V8 de Google. Funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas, las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto. [6]

Socket.IO utiliza principalmente el protocolo WebSocket con sondeo como opción alternativa, a la vez que proporciona la misma interfaz. Aunque se puede usar simplemente como un contenedor para WebSocket, ofrece muchas más características, incluida la transmisión a múltiples sockets, el almacenamiento de datos asociados con cada cliente y las E / S asíncronas. [7].

Node.js + Socket.io = Real-Time IO

Cuando se trata de requisitos en tiempo real, Socket.io puede jugar un papel importante en la arquitectura de la aplicación web, proporcionando una capa de abstracción para la comunicación entre el navegador y el servidor. El modelo impulsado por eventos Node.js combinado con las capacidades en tiempo real de Socket.io ofrece una buena solución para enfrentar los desafíos de BigData en dominios donde la capacidad en tiempo real es importante. [8]

JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Está basado en un subconjunto del Lenguaje de Programación JavaScript. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.

Procesamiento de imágenes.

OPENCV (Open Source Computer Vision Library) es una biblioteca libre de visión artificial, originalmente desarrollada por Intel. Se ha venido utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos.

El sistema de transporte del prototipo SECVIA detecta la llegada de engranajes, captura la imagen y realiza el procesamiento de dicha imagen, para luego aplicar las diversas funciones matemáticas que permiten identificar y reflejar datos técnicos de un engranaje recto, datos correspondientes al diámetro externo e interno, su número de dientes, el paso y módulo.

III. Resultados

Prototipo SECVIA.

En la figura 5 se puede ver la implementación de los diferentes dispositivos hardware que integran SECVIA. Se ha construido un prototipo funcional preparado para añadir el interfaz gráfico mediante un aplicativo web.

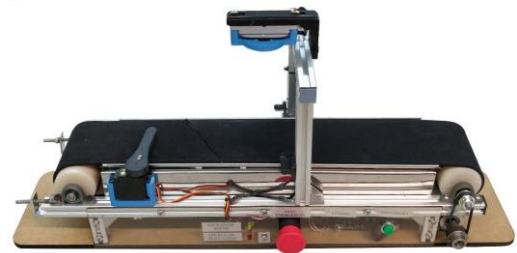


Fig. 5 Prototipo académico SECVIA

El interfaz gráfico del aplicativo web se ha desarrollado tal como se muestra en la figura 6. Está conformado por varias ventanas de navegación que hacen referencia a la información del proyecto, al control y monitorización del prototipo, historial y estadísticas de los engranajes procesados y una ventana con documentos relacionados con la implementación y configuración del sistema ciber-físico. La interacción del alumno con la aplicación web se centra en el control y monitorización, sección donde el estudiante logrará interpretar las diferentes acciones que realiza el prototipo, así como también el detalle del procesamiento de los engranajes.



Fig. 6 Interfaz gráfica del aplicativo web

En la tabla 1 se muestra el resultado del procesado realizado en dos tipos de engranajes. El ensayo uno corresponde a un engranaje de dientes completos y el ensayo dos a un engranaje con defectos en las crestas de los dientes. También se detallan las características técnicas de cada uno de ellos, así como la posición de los dientes perdidos en el caso del engranaje que presente fallas.




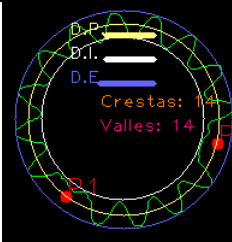
| Ensayo 1 | Ensayo 2 |
|---|--|
|  |  |
|  |  |
| <p>***Información del engrane*** Hay 0 dientes perdidos * Diámetro exterior (De): 39.17 mm * Diámetro interior (Di): 31.55 mm * Módulo(M): 2 * Paso(P): 6.28319 * Número de dientes(Z): 18</p> | <p>***Información del engrane*** Hay 2 dientes perdidos * Diámetro exterior (De): 38.0673 mm * Diámetro interior (Di): 28.67 mm * Módulo(M): 2 * Paso(P): 6.28319 * Número de dientes(Z): 16 ***** P1 No hay diente P2 No hay diente</p> |

Tabla 1. Resultado del procesamiento de un engranaje.

IV. Guía práctica del alumno.

Se ha elaborado una guía práctica para que el alumno pueda seguir la aplicación y comprender como se ha desarrollado SECVIA. En la primera parte de la guía, el estudiante comandará desde el aplicativo web cada uno de los actuadores del prototipo. El funcionamiento del CPS SEVCIA será de modo manual.



Fig. 7 Laboratorio # 1, modo funcionamiento manual

En la segunda parte de la guía, el estudiante interactuará con el sensor de posición, estructura formada por un diodo led emisor y un receptor.

El valor recibido por este último elemento es conocido como umbral de activación. El alumno deberá realizar el proceso de calibración del umbral que permite detectar los engranajes correctamente.

V. Conclusiones

Se ha construido un sistema ciber-físico completamente funcional, con la capacidad de controlar, monitorizar, procesar y clasificar engranajes rectos en tiempo real.

Se han integrado sensores, actuadores, sistemas mecánicos de transmisión y transporte en el diseño mecánico y se han utilizado un microcontrolador y un microprocesador para el diseño electrónico.

Se ha podido verificar el correcto funcionamiento de los dispositivos hardware y programas software, así como la implementación de los diferentes algoritmos para la creación del entorno CPS y el procesamiento de engranajes rectos.

Se ha diseñado una guía práctica enfocada a la comprensión de un sistema de control manual del CPS, y a la integración de un método de calibración del sensor de posición del CPS. El objetivo de esta guía es que el estudiante sea capaz de implementar funciones y procedimientos adicionales en el prototipo, con la posibilidad de corroborar y comprender principios y conceptos de los CPS.

Este estudio sobre CPS abre una ventana de líneas de trabajos futuros. Entre ellos podemos destacar los siguientes:

Ampliar y mejorar la guía práctica del alumno con aplicaciones nuevas que permitan al estudiante interactuar más fácilmente con el sistema y de esta forma poder mejorar el proceso de aprendizaje.

Implementar una base de datos y un servidor en la nube, herramientas de gran importancia para una industria, debido a que permite organizar los datos recogidos del sistema físico y tener la flexibilidad para acceder fácilmente a recursos en el momento que sean solicitados.

Añadir algoritmos de visión y procesamiento de otro tipo de objetos que no sean engranajes rectos, con la finalidad de contar con un sistema de clasificación múltiple.

Insertar algoritmos con metodologías Machine Learning, logrando obtener un sistema complejo, sistema que podría ser capaz de evolucionar y perfeccionar su comportamiento de forma autónoma.

Integrar dispositivos de control, actuadores y sensores adicionales, con el propósito de disponer de un sistema de mayor complejidad, más cercano al ámbito industrial.

Utilizar un dispositivo programable alternativo al Raspberry pi y Arduino. Se podría pensar en el uso de una FPGA. Este dispositivo es capaz de integrar la parte hardware y software, logrando simplificar el diseño electrónico, mejorar el tiempo de respuesta en la red y contar con el ahorro en recursos.

Referencias.

- [1] E. Gomez, «smartLIGHTING,» 17 Mayo 2017. [En línea]. Available: <https://smart-lighting.es/iot-ciberfisicos-industria/>.
- [2] D. S. D. María Antolín Fernández, Abril 2016. [En línea]. Available: www.coiicv.org.
- [3] NEO (Networking and Emerging Optimization), «Networking and Emerging Optimization,» [En línea]. Available: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/Indice.html>.
- [4] culturacion, "Cuál es la utilidad de las aplicaciones cliente/servidor," [Online]. Available: <http://culturacion.com/cual-es-la-utilidad-de-las-aplicaciones-clienteservidor/>.
- [5] node js,Ryan Lienhart Dahl, «Ryan Lienhart Dahl,» Mayo 2009. [En línea]. Available: <https://es.wikipedia.org/wiki/Node.js>.
- [6] C. R. León, «Websockets,» Junio 2013. [En línea]. Available: <http://nereida.deioc.ull.es/~stw/perlexamples/node26.html>.
- [7] hipertextual, «Socket.io: Comunicación bidireccional en tiempo real para JavaScript,» 2014. [En línea]. Available: <https://hipertextual.com/archivo/2014/08/socketio-javascript/>.
- [8] JFernandez, «Node.js + Socket.io = Real-Time IO.,» Diciembre 2012. [En línea]. Available: <https://blogs.igalia.com/jfernandez/2012/12/19/node-js-socket-io-real-time-io/>.