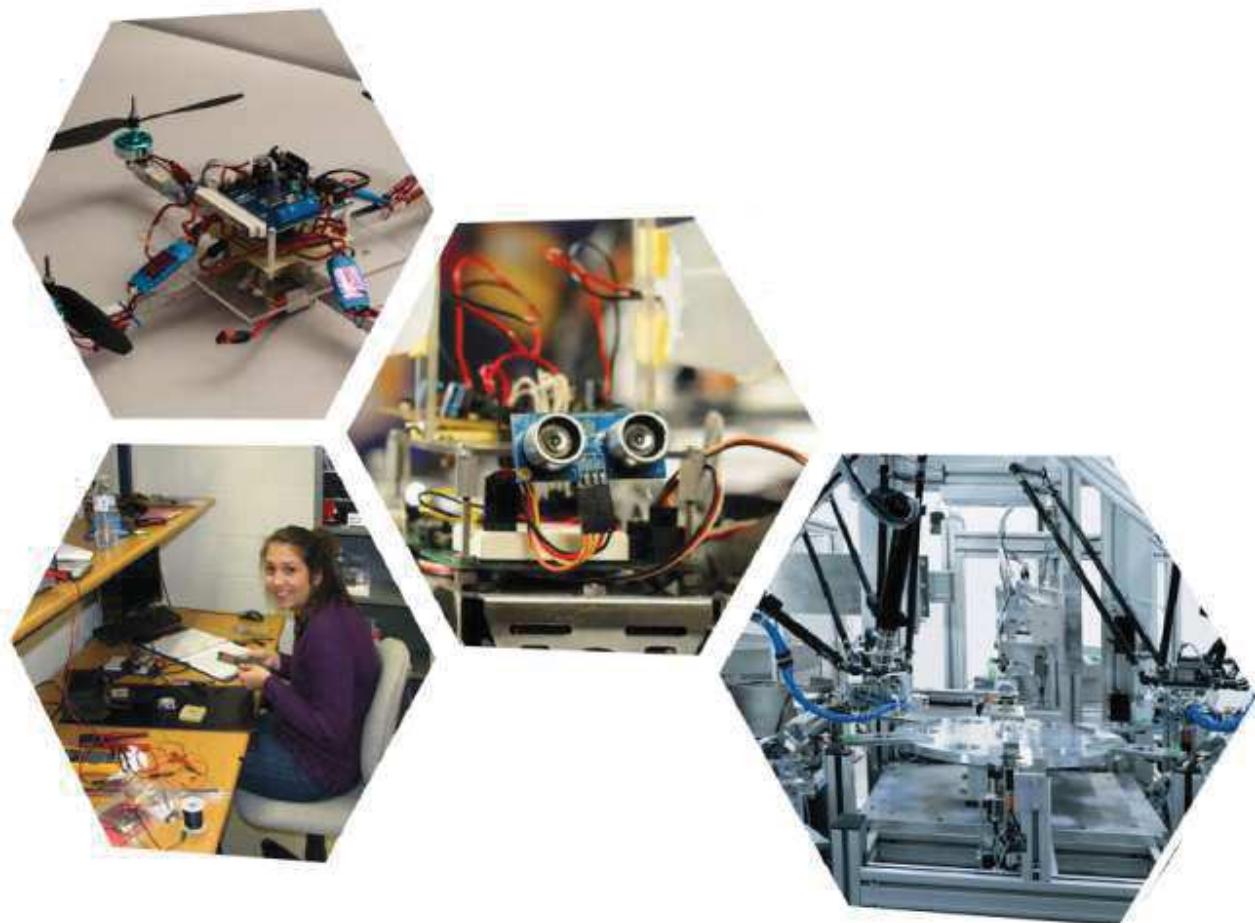


INGENIERÍA MECATRÓNICA

Proyectos y Aplicaciones

José Emilio Vargas Soto
Juan Manuel Ramos Arreguín
Jesús Carlos Pedraza Ortega



ASOCIACION MEXICANA DE
MECATRONICA A.C.

INGENIERÍA MECATRÓNICA.

Proyectos y Aplicaciones

José Emilio Vargas Soto, Juan Manuel Ramos Arreguín
y Jesús Carlos Pedraza Ortega

ISBN: 978-607-9394-00-4

® Asociación Mexicana de Mecatrónica A.C.
Primera edición.
México, Octubre 2014

Mensaje del editor

Una de las prioridades que tiene la Asociación Mexicana de Mecatrónica A.C., es difundir los trabajos e investigaciones que sobre el área de Ingeniería Mecatrónica se desarrollan en México, logrando con ello un acercamiento a los conocimientos y las técnicas que favorecen el desarrollo de personas con habilidades propias para diseñar, construir, controlar y documentar el desarrollo de sistemas con alto valor tecnológico.

El material que se presenta en este libro es una colección de trabajos realizados recientemente por diversos grupos de investigación y desarrollo en diversas Instituciones educativas mexicanas. Los trabajos seleccionados son una muestra del quehacer científico y tecnológico que se encuentra en desarrollo, así como las tendencias en nuevas áreas del conocimiento.

La misión de la Asociación Mexicana de Mecatrónica se ve favorecida con el contenido de ésta obra, misma que esperamos sea de utilidad para aquellas personas interesadas en aprender sobre la ingeniería de proyectos mecatrónicos. Cada capítulo en sí mismo es un tema de estudio que merecerá un análisis particular de los conceptos y técnicas que presentan los autores.

José Emilio Vargas Soto
Editor honorario

Índice

Capítulo 1

Diseño Asistido por Computadora para la Implementación de una Estación Repetidora en el ITS LP (XE2PTS).

1

Ramón Díaz de León Zapata, Itzcoatl Balderas Rivera, Oscar Muñoz Cruz, Ismael Lara Velázquez y Francisco Javier González Contreras

Universidad Autónoma de San Luis Potosí, Instituto Tecnológico de San Luis Potosí.

Capítulo 2

Implementación de un Nodo IRLP Multiplataforma.

6

Díaz de León Zapata Ramón, Balderas Rivera Itzcoatl, Lara Velázquez Ismael, Olivo Sotomayor Jesús Ricardo y Hernández Marín Maribel.

Universidad Autónoma de San Luis Potosí, Instituto Tecnológico de San Luis Potosí.

Capítulo 3

Diseño Mecatrónico en FPGA del Control de Posición de un Elemento Servoneumático.

12

Arroyo Chávez Víctor Ventura, Ramírez Rosales Salvador, Francisco Hernández-Hernández, José-Emilio Vargas-Soto y Ramos Arreguín Juan Manuel.

Universidad Tecnológica de San Juan del Río, Universidad Autónoma de Querétaro.

Capítulo 4

Construcción y Control de un Péndulo Invertido sobre un Carro (PISC) con Control Esclavo de Corriente por Computadora.

16

Espíndola- López Eduardo, Carrillo-Serrano Roberto Valentín y Rivas-Araiza Edgar Alejandro.

Universidad Autónoma de Querétaro.

Capítulo 5

Desarrollo de Aplicación Domótica con Comunicación Inalámbrica Bluetooth.

23

Sotelo Martínez Samuel, Olivo Flores Marco Antonio y Rodríguez Ortiz Juan Gabriel.

Universidad Tecnológica de San Juan del Río.

Capítulo 6

Concentrador Fresnel Utilizando un Controlador Lógico Programable.

30

Terán Valenzuela María Guadalupe, Vega Castro Jorge Luis,

Lugo Coporo Marcos Raúl y Ozuna Huerta Gustavo.

Universidad de Sonora.

Capítulo 7

Diseño del Sistema de Control para un Mezclador Volumétrico de Concreto Utilizando un Micro-controlador.

34

Lugo Coporo Marcos Raúl, Vega Castro Jorge Luis,

Gómez Palomares Jesús Leonel y Ozuna Huerta Gustavo.

Universidad de Sonora.

Capítulo 8

Concentrador Parabólico Controlado con Lógica Difusa.

39

Ozuna Gustavo, Wozniak Marek, Vega Castro Jorge Luis y

Lugo Coporo Marcos Raúl.

Universidad de Sonora, Politechnika Lodzka.

Capítulo 9

Monitoreo en Tiempo Real de las Variables Físicas de un Panel Solar a Traves de LabView.

42

Moreno Castro Luis Fernando, Cruz López Miguel Alberto, Ramiro Ramiro José,

Guerrero Rodríguez Erick Damián y Martínez Jiménez José Manuel.

Instituto Tecnológico Superior De Zacapoaxtla.

Capítulo 10	
<u>Simulador Basado en un Sistema Bípedo Simple.</u>	47
Cuatlaxahue-Formacio Sandra y Sánchez-Sánchez Pablo	
Benemérita Universidad Autónoma de Puebla.	
Capítulo 11	
<u>Análisis de la Precisión en los Desplazamientos de una Plataforma Móvil por Control de Motores DC.</u>	53
Apolinar Garnica Erik, Calderón Rocha José Carlos, González Torres Martín Eduardo, Rodríguez Aviña Juan Pablo, Caudillo Estrada Juan José y León-Rodríguez Miguel.	
Universidad Politécnica del Bicentenario, Universidad de Santiago de Chile y Centro de Investigaciones en Óptica.	
Capítulo 12	
<u>Velocidad de Respuesta de Brazo Robótico Utilizando sensor Kinect.</u>	58
Rodriguez Sánchez Eder Germain, Martinez Gracia Cristian Roberto, Moreno Castro María Viridiana, Jiménez Juárez Jorge Alberto, Banda Sierra Jonathan, Gonzalez Esquivel Arturo, Preciado Prieto Alberto y León Rodríguez Miguel.	
Universidad Politécnica del Bicentenario, Centro de Investigaciones en Óptica y Universidad de Santiago de Chile	
Capítulo 13	
<u>Implementación de Movimientos Complejos en Modelo Mecatrónico de Pruebas de Rodilla.</u>	63
Rangel D., Suárez C., Chagolla H., Olvera A. y Estévez M.	
Centro de Física Aplicada y Tecnología Avanzada, UNAM. Juriquilla	
Instituto Tecnológico de Querétaro.	
Capítulo 14	
<u>Modelado, Simulación y Control PID de un Evaporador con Refrigerante HFC-134a en Matlab.</u>	69
Aguilar Anastacio José Carlos A, Martín Salazar Pereyra y Pedro Romano Apórtela.	
Tecnológico de Estudios Superiores de Ecatepec.	
Capítulo 15	
<u>Adquisición de Datos en LabVIEW para un Sistema de Seguimiento Solar.</u>	75
Martínez Jiménez José Manuel, Vergara Betancourt Angel, Moreno Castro Luis Fernando y Ramiro Ramiro José	
Instituto Tecnológico Superior de Zacapoaxtla.	
Capítulo 16	
<u>Aprendizaje Basado en Proyectos en el Desarrollo de un Sistema Mecatrónico.</u>	81
Jiménez López Eusebio, Hernández Flores María Enedina, Bojórquez Morales Gloria Isabel, Navarro Fragoso Lilia Beatriz, Gaytán Martínez Lilia Zulema, Curiel Aguirre Fernando y Luna Sandoval Gabriel.	
Universidad La Salle Noroeste, Universidad Tecnológica del Sur de Sonora y Universidad Estatal de Sonora.	
Capítulo 17	
<u>Diseño Conceptual de un Robot Paralelo Reconfigurable.</u>	87
Sánchez Alonso Róger Ernesto, González Barbosa José Joel y Castillo Castañeda Eduardo.	
Instituto Politécnico Nacional. CICATA - Querétaro.	
Capítulo 18	
<u>Ánálisis de la Distribución de Temperatura sobre una Barra con Arduino.</u>	93
Aguilar-Loreto Omar, Muñoz-Gómez José Antonio y Fernández-Raudales Juan Manuel.	
Universidad de Guadalajara, Universidad de Guadalajara.	

Capítulo 19	
<u>Análisis del Comportamiento de un Panel Solar a Través de Scilab.</u>	101
Guerrero Rodríguez Erick Damián, Moreno Castro Luis Fernando, Ramiro Ramiro José y Cruz López Miguel Alberto. Instituto Tecnológico Superior de Zácapoaxtla.	
Capítulo 20	
<u>Sistema de Presencia Remota.</u>	106
Charre Ibarra Saida, López Luiz Norberto, Rodríguez García Oscar Alejandro, Palafox Rico Gerardo Javier, Aguilar Castillo Rodolfo Benjamín, Gutiérrez Figueroa Pedro José y Vergara Valora Emmanuel. Universidad de Colima.	
Capítulo 21	
<u>Procesamiento de Imágenes Aplicado al Control de Robots Manipuladores.</u>	111
Palma Asunción Adilene, Reyes Cortés Fernando, Cid Monjaraz Jaime, Vargas Treviño Aurora y Vergara Limón Sergio. Benemérita Universidad Autónoma de Puebla.	
Capítulo 22	
<u>Un Esquema 3D para la Descripción Visual de Gestos Dinámicos.</u>	117
Medrano-Aguilar José Jesús, Avilés-Arriaga Héctor Hugo, Gómez-Jáuregui David Antonio, Herrera-Rivas Hiram y Nuño-Maganda Marco Aurelio. Universidad Politécnica de Victoria, Université Paris-Sud.	
Capítulo 23	
<u>Diseño de un Sistema de Control para Generación de Energía Eléctrica en Bicicletas Horizontales.</u>	123
Méndez Zapata Elías y Cano Corona Ariana. Universidad Politécnica de Tlaxcala.	
Capítulo 24	
<u>Diseño y Análisis Estructural de un Soporte Parcial de Peso para Fisioterapia.</u>	129
Arriaga-Arriaga, Cesar Augusto, Gómez-Pavón Luz del Carmen, Muñoz-Pacheco Jesús Manuel, Félix-Beltrán Olga Guadalupe y Luis-Ramos Arnulfo. Benemérita Universidad Autónoma de Puebla.	
Capítulo 25	
<u>Control Háptico con Planificación de Trayectorias Aplicado a Novint Falcon.</u>	135
E. Zepeda-Ruelas, J. Gudiño-Lau, M. Durán-Fonseca, S. Charre-Ibarra y J. Alcalá-Rodríguez. Universidad de Colima.	
Capítulo 26	
<u>Aplicación en Telefonía Móvil para el Control de un Motor Trifásico Mediante Comunicación Bluetooth.</u>	141
Aceves Flores Julio Cesar, Flores Salazar Erick Eduardo y Aguilar Anastacio José Carlos A. Tecnológico de Estudios Superiores de Ecatepec.	
Capítulo 27	
<u>Laboratorio Virtual de Escenarios Industriales para el Adiestramiento en el Área de Automatización y Control.</u>	147
Barreiro Fabián A., Barreiro Aldo J. y Ortega-Moody Jorge Alberto Academic Division of ANT-Automation.	
Capítulo 28	
<u>Desarrollo de una Plataforma de Simulación Basada en el Robot Antropomórfico K-680.</u>	152
González-Romero Dulce Isabel, Sánchez-Sánchez Pablo y Castañeda-Camacho Josefina. Benemérita Universidad Autónoma de Puebla.	

Capítulo 29	
<u>Implementación de un Sistema Neurodifuso en la Optimización del Modelo de Control para el Ambiente de una Incubadora Avícola.</u>	158
Conde Camacho Julián, Sánchez Cuapio Iván Jesús y Atecpanecatl Gutiérrez Jonathan. Universidad Tecnológica de Tlaxcala.	
Capítulo 30	
<u>Instrumentación de Señales de Frecuencia Cardíaca para un Sistema Ambulatorio en Dispositivo.</u>	164
Sánchez Medel Luis Humberto, Juan Manuel González Sobal, Olivia Labastida Puertos y Solís Jiménez Miguel Ángel. Instituto Tecnológico Superior de Huatusco.	
Capítulo 31	
<u>Síntesis de un Sistema de Monitoreo del Consumo Eléctrico Implementado en un Dispositivo Móvil.</u>	168
Sánchez Medel Luis Humberto, Solís Jiménez Miguel Ángel y Ramírez Rebolledo Miguel Ángel. Instituto Tecnológico Superior de Huatusco.	
Capítulo 32	
<u>Simulación del Modelo VegSyst para Determinar el Crecimiento de la Planta de Jitomate.</u>	173
Mancilla Morales J. Arturo, Tornero Campante Mario, López Cruz Irineo, Ríos Silva Eduardo y Regalado López José. Colegio de Postgraduados-Campus Puebla. Universidad Autónoma de Chapingo y Benemérita Universidad Autónoma de Puebla.	
Capítulo 33	
<u>Simulador Applet de un Pándulo Simple Orientado a la Enseñanza de la Física a Nivel Secundaria.</u>	179
Altamirano Soria José Edgar, Jiménez Nava Jimena, Quijas Martínez Julio, Ramos Gutiérrez Jesús, Ulloa Calzada José Luis, José Emilio Vargas Soto y Jesús Carlos Pedraza Ortega. Universidad Autónoma de Querétaro.	
Capítulo 34	
<u>Control de un Robot con Dispositivos Móviles.</u>	185
Arana-González Lizet-Verenice, Cordero-Victoria Isaac-Jair, Avecilla-Rangel Francisco-Javier, Juan-Manuel Ramos-Arreguin y Efren Gorrostieta-Hurtado. Universidad Autónoma de Querétaro.	
Capítulo 35	
<u>Uso de los Software LinuxCNC y Eagle 6.4.0. en el Diseño y Fabricación de Circuitos Impresos por Medio de un Robot Cartesiano.</u>	193
Urbalejo Contreras Arturo, Jiménez López Eusebio, Luna Acosta Noé, Valdez Palafox Carlos Omar, José Antonio Cuadras Gerardo y Castro Bojórquez Julio Cesar. Universidad Tecnológica del Sur de Sonora, Universidad La Salle Noroeste – CIAAM-UTS-IIMM, Universidad Tecnológica del Sur de Sonora y Universidad Tecnológica de Tijuana.	
Capítulo 36	
<u>Diseño y Simulación de los Sistemas de Fundición, Instrumentación Electrónica y Diseño Mecánico para un Prototipo de Maquina Fundidora de PET.</u>	199
Guillermo Córdova Morales, Elmar Montiel Jiménez y Espinoza Hernández Andrés. Instituto Tecnológico Superior de Libres.	
Capítulo 37	

Uso del Software Solidworks en el Diseño de Sistemas Mecatrónicos.	205
Ruiz Domínguez Alejandra, Meléndez Campos Javier, Jiménez López Eusebio, Montoya Quezada Manuel Norberto, García Velásquez Luis Andrés, Lucero Velásquez Baldomero, Delfín Vázquez Juan José y Luna Sandoval Gabriel. Universidad La Salle Noroeste, Instituto Tecnológico de Sonora, Instituto Tecnológico Superior de Cajeme y Universidad Estatal de Sonora.	
Capítulo 38	
Control por Dinámica Inversa de un Manipulador Paralelo en el Espacio Articular.	211
Alvarado-Juárez Diego Armando, Villarreal-Cervantes Miguel Gabriel y Sepúlveda-Cervantes Gabriel. Centro de Innovación y Desarrollo Tecnológico en Cómputo - IPN.	
Capítulo 39	
Metodología para la Detección del Desalineamiento en Motores de Inducción Alimentado con Variador de Velocidad.	217
Mejía-Barrón A., García-Ramírez A.G., Osornio-Ríos R.A. y Romero-Troncoso R.J. Universidad Autónoma de Querétaro, Universidad de Guanajuato.	
Capítulo 40	
Aproximación Metodológica en la Enseñanza de Lenguajes de Programación Orientados a Sistemas Mecatrónicos.	223
Curiel Aguirre Fernando y Jiménez López Eusebio. Universidad Estatal de Sonora, Universidad Tecnológica del Sur de Sonora y ULSA Noroeste.	
Capítulo 41	
Controlador PID de una Celda Termoeléctrica: Sintonización por medio de un Algoritmo Evolutivo.	229
García Mejía Juan Fernando, Carlos Eduardo Torres Reyes y José Antonio García Mejía. Universidad Autónoma del Estado de México.	
Capítulo 42	
Diseño, Modelación y Simulación de un Brazo Robótico de cinco grados de libertad Utilizando LabVIEW y SolidWorks.	233
Ramírez Escobar Raúl Alonso, Martínez Soto Luis Ángel, Vázquez Cuevas Ignacio Javier, Meranza Castillón Manuel Omar, González Ochoa Sara Bertha y Jiménez López Eusebio. Instituto Tecnológico de Nogales y Universidad La Salle Noroeste – CIAAM-UTS-IIMM.	
Capítulo 43	
Desarrollo de un Simulador de un Robot Cartesiano para Dispositivos Móviles Utilizado Solidworks y Open GL ES 2.0	239
Meléndez Campos Javier, Jiménez López Eusebio, Ruiz Domínguez Alejandra, Urbalejo Contreras Arturo, Reyes Ávila Luis, Luna Acosta Noé y Vázquez Cuevas Ignacio. Universidad La Salle Noroeste, Universidad Tecnológica del Sur de Sonora, Instituto Mexicano del Transporte y Universidad Tecnológica de Nogales.	
Capítulo 44	
Integración Mecatrónica en el Desarrollo de un Robot Cartesiano Didáctico.	245
Jiménez López Eusebio, Ruiz Domínguez Alejandra, Meléndez Campos Javier, Pereo Arvayo Gerardo, Carvajal Lavandazo Ivanna, Martínez Molina Víctor Manuel, López Figueroa Francisco y Luna Sandoval Gabriel. Universidad La Salle Noroeste, Universidad Tecnológica del Sur de Sonora-ULSA Noroeste-CECATI 94 y Universidad Estatal de Sonora.	

Diseño Asistido por Computadora para la Implementación de una Estación Repetidora en el ITSLP (XE2PTS)

Díaz de León Zapata Ramón^{1,2*}, Balderas Rivera Itzcoatl², Muñoz Cruz Oscar², Lara Velázquez Ismael², González Contreras Francisco Javier¹.

¹Coordinación para la Innovación y la Aplicación de la Ciencia y la Tecnología, Universidad Autónoma de San Luis Potosí, Sierra Leona 550, Lomas 2a Sección, SLP, México.

Departamento de Ingeniería Eléctrica, Electrónica y Mecatrónica, Instituto Tecnológico de Sal Luis Potosí, Av. Tecnológico s/n Soledad de Graciano Sánchez, San Luis Potosí, C.P. 78437, México.
*ramondz@hotmail.com

Resumen

La instalación y operación de una estación repetidora del servicio de aficionados requiere el trámite de un permiso especial que, entre otros aspectos, implica la elaboración de un estudio denominado “información técnica descriptiva” en la que se debe plasmar detalles de predicciones de cobertura según la topología del terreno y el tipo de emisión de la antena, etc.

La elaboración del estudio, se basa en la norma oficial mexicana NOM-EM-086-SCT1-1994 publicada en el año de 1994 y la que no ha sido modificada desde entonces. Dicha norma sugiere procedimientos manuales pero deja abierta la posibilidad de hacer uso de tecnologías o métodos más eficientes y eficaces.

Con el avance en la computación y el impulso de proyectos como el “Space Shuttle Radar Terrain Mapping Mission” (SRTM), que además pone sus resultados a disposición gratuita de quien lo requiera, se ha creado software capaz de usar mapas cartográficos a los que se les añade o superpone de manera interactiva, las ecuaciones o funciones de la teoría electromagnética para simular el comportamiento de las transmisiones de radio y su cobertura, como el programa “Radio Mobile” de Roger Coudé.

Se desarrolla en el presente trabajo los resultados del estudio para la elaboración del informe técnico descriptivo asistido por computadora (CAD) para la instalación y operación de la estación repetidora del servicio de aficionados del Instituto Tecnológico de San Luis Potosí.

Palabras clave: Simulación, CAD, Telecomunicación, Estación de radio.

1. Introducción

La instalación y operación de una estación del servicio de aficionados requiere un estudio técnico denominado “información técnica descriptiva” cuyas secciones modulares se encuentran descritas en la Norma Oficial Mexicana NOM-EM-086-SCT1-1994 publicada en el Diario Oficial de la Federación [1]. Desde la fecha de publicación de esta norma, en el año de 1994, han ocurrido avances significativos en la manera en que se pueden obtener los datos predictivos solicitados con una sencillez, precisión y representaciones visuales muy superiores a las que se tenía alcance en aquellos años. Dado también que la norma en su sección de apéndices informativos a la letra dice: “Se hace notar que el método de predicción se recomienda por práctico, reconociéndose que pueden existir otros que arrojen resultados mejores y que en todo caso se pueden considerar para su aceptación”, se ha optado por utilizar como herramienta de apoyo para el presente estudio, el programa de cómputo denominado “Radio Mobile”.

El “Radio Mobile” es un programa Gratuito de simulación de propagación escrito por Roger Coudé - VE2DBE, que opera en el rango de frecuencias de 20 MHz a 20 GHz. Se basa en la última versión del modelo de propagación sobre Terreno Irregular ITM (Longley-Rice).

El programa permite implementar mapas de elevaciones usando los datos del “Space Shuttle Radar Terrain Mapping Mission” (SRTM), pudiendo añadirse curvas de nivel y cartas camineras, junto con

las especificaciones de las llamadas *unidades de radio* que son las estaciones transmisoras y/o receptoras y emplazarlas donde se requieran. El desempeño individual de cada unidad de radio se caracteriza por sistemas con propiedades específicas en lo que respecta a: potencia, sensibilidad, parámetros de antena etc. y en todos los enlaces de radio programados es posible examinar los perfiles del tramo y los parámetros de la señal.

Si se requiere, puede producir diagramas de cobertura de señales de unidades de radio múltiples o individuales, o encontrar los mejores sitios para su instalación en un mapa, para proporcionar cobertura de radio a un número especificado de unidades de radio.

También con la prestación “Route Radio Coverage” (cobertura de rutas) se puede generar el desempeño de comunicación con una unidad de radio que se desplaza por una ruta definida en un mapa, como así mismo es posible utilizar los datos de la ruta con la prestación *Mejores sitios*, a fin de proporcionar una cobertura completa de dicha ruta.

La prestación *Mejores Unidades de Radio* de un conjunto definido de estas unidades, proporciona el trazado del máximo nivel de señal y la razón Señal a Interferencia de una ubicación específica. También está disponible la cobertura combinada de Radio de un selecto número de unidades, como así mismo el trazado de la suma de las señales de múltiples unidades en cada ubicación [2].

2. Simulación del trazado de perfiles

Si bien el programa puede calcular el mejor lugar (por ejemplo dentro de la ciudad o en sus cercanías) para instalar la estación repetidora, en este caso la ubicación ya está predeterminada para ubicarse en el laboratorio de electrónica del ITS LP a una altura máxima de 30 metros sobre el nivel del piso.

Sabiendo la ubicación, se localiza ésta en el mapa según las coordenadas que pueden obtenerse directamente de los mapas de google, a través del uso de la interfaz del mismo programa “Radio Mobile” o conociendo la cuadrilla (“grid”) que a su vez puede ubicarse fácilmente con el uso de algún dispositivo de telefonía dotado con GPS o con un GPS independiente. En este caso se ubicó el punto en un mapa de google importado por el programa, como se aprecia en la figura 1.



Figura 1. Ubicación de la estación Repetidora.

Los datos de elevación del terreno se han obtenido de bases de datos del proyecto “Space Shuttle Radar Terrain Mapping Mission” (SRTM) y han sido reconstruidos en forma de imagen con la ayuda de google maps, como se aprecia en la figura 2.

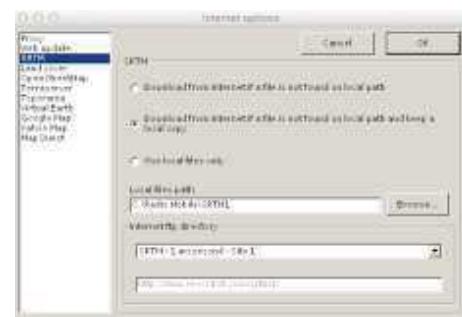


Figura 2. Captura de pantalla donde se aprecia la dirección electrónica de la descarga de archivos SRTM para el cálculo de elevaciones.

Cabe señalar que la mejor resolución para nuestro país es de 1 segundo de arco, lo cual implica una muy buena resolución, ya que para Europa se tiene una resolución de 3 segundos de arco (mucha menor resolución) y para Estados Unidos de Norte América una resolución de 1/3 de arco (una excelente resolución).

La figura 3 muestra el perfil de elevaciones con una escala de colores alrededor de la ciudad de San Luis Potosí en un radio de aproximadamente 50 kilómetros y sobre la cual posteriormente se le sobrepondrá el mapa con las calles y carreteras para una mejor referencia de las ubicaciones.



Figura 3. Perfil de elevaciones para el terreno de cobertura de la estación repetidora. Se aprecia en la gráfica la ubicación de la estación base.

Posteriormente se alimenta al programa con las características generales que tendrá la estación repetidora, según se aprecia en la figura 4, y en donde algunos de los parámetros son estándares internacionales considerados por la Unión Internacional de Telecomunicaciones (UIT).

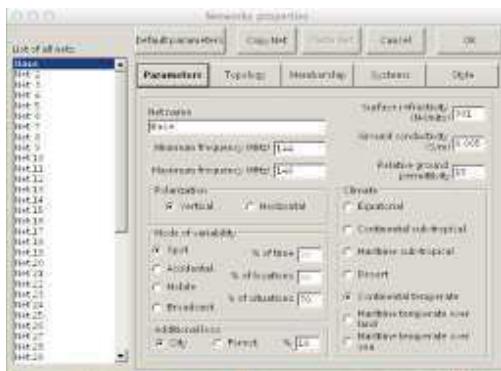


Figura 4. Parámetros de la Estación Repetidora.

El rango de frecuencia de operación se ajustó de 144 a 146 MHz, la refractividad de la superficie en 801 (sin unidades), la Conductividad de la tierra en 0.005 S/m, la permitividad relativa de la tierra en 15 (sin unidades), con una polarización vertical (por el tipo de antena a utilizarse), condiciones de temperatura continentales citadinas con una máxima pérdida de la señal por absorción de edificios y estructuras de concreto de 10%.

El patrón de radiación de la antena propuesto se aprecia en la figura 5.

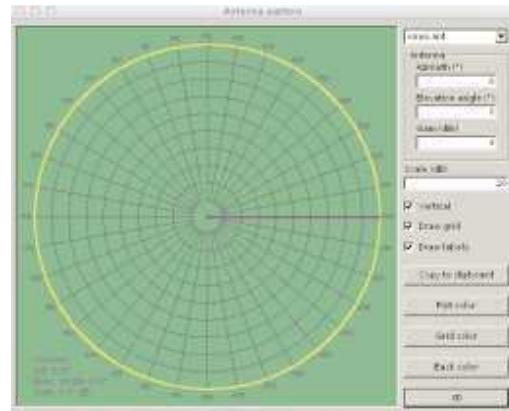


Figura 5. Patrón de radiación de la antena tipo J a ser utilizada en la estación repetidora.

Los resultados se aprecian en el mapa de perfiles del área de cubrimiento según la topología de elevación del terreno y el patrón de radiación en la figura 6. En la misma imagen se representan los rangos de potencia en una escala de colores correspondiente a las intensidades máximas y mínimas de potencia efectiva detectable en μV .

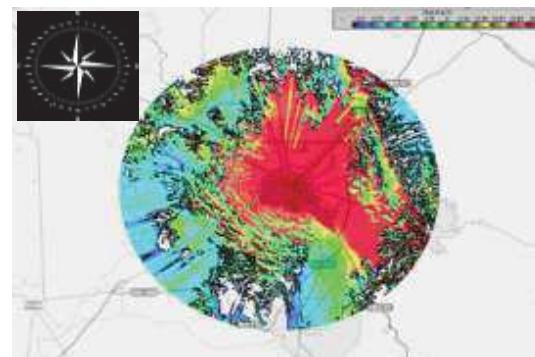


Figura 6. Mapa de perfiles del área de cubrimiento o servicio según la topología de elevación del terreno obtenido con el programa “Radio Mobile”. En el centro se aprecia la ciudad de San Luis Potosí con su respectivo anillo periférico.

3. Simulación y obtención de las predicciones

Con base en los resultados del apartado anterior, se detalla a continuación el procedimiento para la obtención de las predicciones con el uso del programa “Radio Mobile”.

Dado que el mapa contiene las alturas del terreno del área bajo análisis, se utiliza una propiedad del programa “Radio Mobile” en la que es posible sobreponer una rejilla con una resolución de 1 km cuadrado que servirá como referencia para ubicar las

alturas directamente sobre el mapa, como se aprecia en la figura 7.

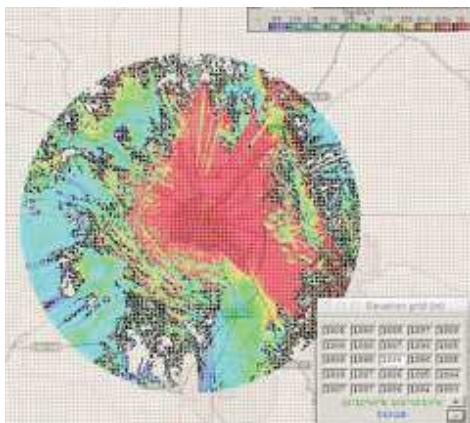


Figura 7. Mapa con enrejado en escala de 1 km cuadrado por división. Se aprecia en la esquina inferior derecha un cuadro con las elevaciones en que se desea averiguar el dato al hacer click sobre el punto de interés en la imagen.

Se han trazado 8 radiales desde el centro que representa la ubicación de la estación repetidora (líneas rojas punteadas), cada uno a 45° con el primer radial apuntando al norte geográfico. En cada radial se tomaron un total de 20 puntos para calcular el promedio, separados un máximo de 500 m según la norma NOM-EM-086-SCT1-1994 (para nuestro caso, dos lecturas por cada recuadro del mapa). Los datos tomados y el promedio solicitado se aprecian en la tabla 1.

Nº Muestreo	Radial 1 (N)	Radial 2 (NE)	Radial 3 (E)	Radial 4 (SE)	Radial 5 (S)	Radial 6 (SO)	Radial 7 (O)	Radial 8 (NO)
1	1899	1896	1898	1896	1896	1898	1896	1896
2	1893	1889	1885	1876	1888	1891	1874	1871
3	1894	1891	1887	1881	1884	1887	1877	1874
4	1873	1864	1853	1877	1824	1778	1807	1885
5	1880	1865	1860	1870	2037	2067	1960	1885
6	1832	1860	1862	1870	2057	2122	1913	1844
7	1895	1888	1886	1892	2159	2184	1914	1888
8	1890	1864	1853	1865	2101	2241	1916	1815
9	1851	1852	1850	1826	2152	2087	1926	1821
10	1894	1860	1848	1818	2353	2229	1821	1830
11	1833	1849	1848	1808	2297	2158	1949	1823
12	1893	1848	1848	1862	2200	2187	1867	1815
13	1888	1843	1847	1866	2321	2251	2018	1834
14	1880	1843	1847	1864	2318	2187	2020	1837
15	1894	1842	1845	1876	2498	2388	1960	1821
16	1830	1841	1842	1857	2296	2248	1906	1847
17	1848	1861	1853	1854	2276	2275	1806	1833
18	1883	1841	1844	1847	2540	2323	1859	1838
19	1883	1841	1845	1840	2441	2252	1860	1845
20	1879	1841	1835	1840	2361	2283	1937	1847
Promedio	1885.39	1852.1	1854.39	1886.6	2232.4	2139.4	1825.2	1814.49

Tabla 1. Datos obtenidos por cada radial y el respectivo promedio.

4. Material y equipo

Dadas las características de ubicación de la estación repetidora dentro del laboratorio de electrónica, se contará con todo el material necesario para la instalación, operación, mantenimiento preventivo y correctivo, además de equipo adicional

sofisticado para asegurar de manera permanente su óptimo funcionamiento y aprovechamiento, como son osciloscopios, espectrómetros, multímetros e instrumentación virtual a través de interfaces con programas de cómputo como LabView, Matlab, Mathematica, etc.

La línea de alimentación de la antena se realizará con cable coaxial RG58 de 50 ohms de impedancia.

La antena será omnidireccional con un diseño tipo J para la banda de 2 metros con una ganancia máxima de 6 dB. La figura 8 muestra el diseño correspondiente a la antena [3].

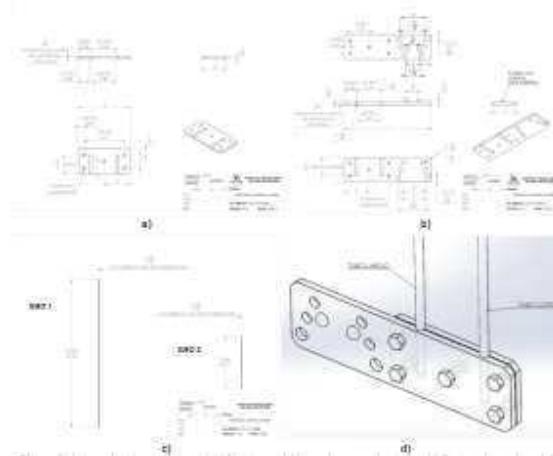


Figura 8. Antena tipo J para la banda de 2 metros. a) Abrazadera para los tubos. b) Contra abrazadera del inciso a, con sección para montaje en mástil. c) Tubos de la antena. d) detalle del acoplamiento de los tubos para formar la antena.

5. Resultados

Se construyó y puso en operación la repetidora con los resultados esperados congruentes con la simulación en el área de interés que corresponde a la mancha urbana y zona conurbada de la ciudad de San Luis Potosí, realizando un recorrido en los alrededores de la estación y la periferia de la ciudad recibiendo una excelente calidad de audio, previa calibración del sistema a través de un analizador de antena RigExpert modelo AA-520 [4], con los resultados mostrados en la gráfica de la figura 9.

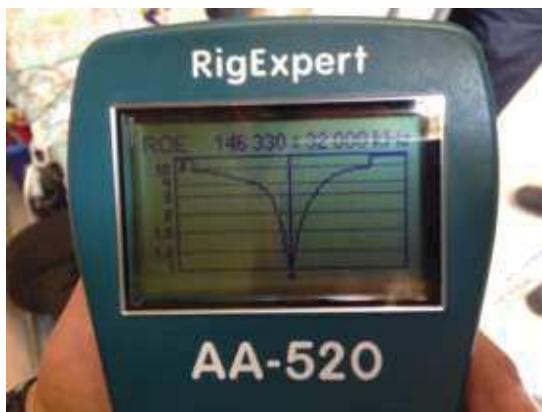


Figura 9. Resultados de la calibración de antena entonada en una frecuencia de 146 MHz.

El equipo transceptor se ajustó a una potencia de salida máxima de 20 W que fueron medidos con la ayuda de un Wattmetro marca MFJ modelo 844, como lo muestra la figura 10.



Figura 10. Potencia máxima de salida del transmisor de la estación repetidora del ITS LP.

6. Conclusiones

La simulación de las condiciones de propagación en el terreno nos permite prever condiciones especiales a tomar en cuenta en la implementación final de la repetidora, así como optimizar los recursos económicos y de inversión de tiempo en el análisis y generación de datos. Además, gracias al esfuerzo conjunto de varias áreas de la ciencia, la ingeniería y la tecnología, fue posible diseñar, construir y caracterizar en su totalidad los parámetros de operación de una estación repetidora del servicio de aficionados para el Instituto Tecnológico de San Luis Potosí.

El uso de software profesional pero gratuito, así como la colaboración de alumnos en el presente estudio, ha dado como resultado una entusiasta participación y un apropiamiento del conocimiento sin precedentes, acentuando el carácter técnico – científico de la actividad de radio aficionado, convirtiéndose en una actividad permanente en el ITS LP, así como un escaparate que sin duda ha influido a varios estudiantes para convertirse en radio aficionados.

La colaboración multidisciplinaria jugó un rol fundamental, pero es igualmente relevante enfatizar que el trabajo interinstitucional UASLP – ITS LP ha sido parte importante del éxito en estas conclusiones.

Agradecimientos

Los autores agradecen el apoyo financiero otorgado por el consorcio CEMIE-Solar a través del proyecto 32.

Se agradece también todas las facilidades otorgadas por las autoridades del Instituto Tecnológico de San Luis Potosí para la operación y mantenimiento de su estación de radio aficionados.

Referencias

- [1] Dario Oficial de la Federación, Tomo CDXCV No. 10 México, D.F., jueves 15 de diciembre de 1994.
- [2] Ian D. Brown. "Radio Mobile Handbook". Antennex Online Magazine. E. E. U. U. 2013. ISBN: 1-877992-48-8.
- [3] R. Dean Straw. "ARRL Antenna Book". The American Radio Relay League, Inc. E. E. U. U. 2011. ISBN: 978-0-87259-694-8.
- [4] Rig Expert AA-520 Antenna Analyzer User's manual. <http://www.rigexpert.com> Ucrania, 2014.

Implementación de un Nodo IRLP Multiplataforma

Díaz de León Zapata Ramón^{1,2*}, Balderas Rivera Itzcoatl², Lara Velázquez Ismael², Olivo Sotomayor Jesús Ricardo² y Hernández Marín Maribel².

¹Coordinación para la Innovación y la Aplicación de la Ciencia y la Tecnología, Universidad Autónoma de San Luis Potosí, Sierra Leona 550, Lomas 2a Sección, SLP, México.

²Departamento de Ingeniería Eléctrica, Electrónica y Mecatrónica, Instituto Tecnológico de San Luis Potosí, Av. Tecnológico s/n Soledad de Graciano Sánchez, San Luis Potosí, C.P. 78437, México.

*ramondz@hotmail.com

Resumen

El Instituto Tecnológico de San Luis Potosí cuenta con una estación de radio para el servicio de aficionados (XE2PTS), la cual operaba hasta antes del desarrollo de este proyecto únicamente en modos analógicos. Con la implementación de un sistema IRLP (Internet Radio Linking Project – Proyecto de Enlace de Radio por Internet) se tiene ahora la capacidad de conectarse a través del protocolo VoIP a distintas salas de conferencias en todo el mundo.

El sistema IRLP original fue concebido para ser implementado en una computadora personal de escritorio con sistema operativo Linux (actualmente en la distribución Debian), sin embargo hemos descubierto ventajas significativas al realizar una implementación multiplataforma ya que según el aspecto que se quiera explotar, bastará con activar el sistema con computadora de escritorio, computadora portátil estilo laptop o ultracompacta con Raspberry Pi para aplicaciones móviles.

Palabras clave: Estación de Radio, Radio Aficionado, Linux, Debian, IRLP.

1. Introducción

El concepto “radio aficionado” suele concebirse como una actividad poco seria. Nada más alejado de la realidad; en principio, para poder serlo se requiere una licencia para instalar y operar estaciones de radio del servicio de radio aficionados [1]. La obtención de la licencia implica acreditar un examen que consta de tres partes: 1. Examen de reglamentos, leyes y normas. 2. Examen de conocimientos sobre electrónica y 3. Examen de telegrafía. En promedio, un aspirante a radio

aficionado requiere un periodo de estudio y preparación entre 6 y 12 meses.

Toda vez que el aspirante demuestra poseer los conocimientos para instalar y operar una estación de radio del servicio de aficionados, recibe un indicativo de llamada único que lo identifica y que debe usar cada vez que se haga presente a través de un comunicado de radio.

Visionarios que en su momento dotaron al Instituto Tecnológico de San Luis Potosí con una estación de radio aficionados le han dejado un legado con enormes potencialidades, ya que entre muchas otras actividades, permite llevar a la práctica los conceptos teóricos estudiados en materias como telecomunicaciones, interfaces para automatización y control, instrumentación, redes basadas en VoIP etc.

Las estaciones de radio aficionados pueden operar en tres modos principalmente: analógico, digital e híbrido (combinación de interfaces analógicas con digitales) y éste último caso es el que corresponde al sistema IRLP, ya que para poder acceder al sistema de VoIP en el que se transmite la información de audio, es imprescindible el uso de un transceptor analógico (radio capaz de transmitir y recibir señales de audio en frecuencia modulada analógica) y éste a su vez, con ayuda de una interfaz de hardware y software transmite y recibe simultáneamente la misma información por internet, rompiendo limitantes de alcance o cobertura implícitas en la banda de transmisión analógica correspondiente, pudiendo llegar a sistemas similares alrededor del mundo.

Se asume además, por parte del creador del sistema IRLP, David Cameron (VE7LTD) que la utilización de un nodo IRLP será parte permanente de un repetidor para obtener el máximo provecho y

llegar al mayor número de radio aficionados. De aceptar la premisa anterior, resulta congruente suponer que el sistema IRLP no tendría por qué ser portátil y, más aún no tendría por qué requerir un sistema de cómputo de altas prestaciones. Así pues, se ha construido e implementado un sistema IRLP con una computadora personal de escritorio con las especificaciones mínimas necesarias tomadas de una computadora que para cualquiera otra aplicación resultaría obsoleta, pero que para el sistema IRLP es más que conveniente ya que se quedará de manera permanente y sin necesidad de supervisión constante como parte del sistema de la estación de radio del ITSLP, la XE2PTS.

Como también es de suponerse, se idearon aplicaciones adicionales a la propuesta original y se observó como ventaja que el sistema fuera portátil, de modo que se pudiera llevar con facilidad el sistema IRLP a cualquiera otra localidad, fuera de las instalaciones de la estación fija para realizar pruebas de campo o realizar experimentos con otras redes e inclusive en otras ciudades ya que durante un par de días al año los radioaficionados prestan un servicio social muy interesante y con mucho entusiasmo a los **Boy Scouts** en un evento denominado “Jamboree del aire”, donde los participantes realizan contacto vía radiofrecuencia en los distintos modos (analógico, digital e híbrido) con similares en todo el mundo.

Sería complejo y hasta riesgoso para el equipo de escritorio un traslado y reinstalación cada vez que esto se requiriera, por lo que se optó por una alternativa con el uso de una computadora portátil o “laptop”, lo cual introdujo algunos retos adicionales a resolver, como el uso de una tarjeta de red inalámbrica y la adaptación de un puerto paralelo, pensando en que la utilidad adicional al sistema IRLP con una computadora de características y prestaciones recientes permitiera su uso también para todas las actividades cotidianas del usuario promedio de estos equipos.

Finalmente, se puede tener la operación de un nodo IRLP con una computadora minimalista que ofrezca no sólo capacidades portátiles sino ahora incluso móviles para lo cual se exige un mínimo consumo de batería y tamaño ultra compacto para que sea conveniente su operación mientras se transporta por una persona. Tal capacidad se logró con la implementación del nodo a través de una microcomputadora denominada Raspberry Pi.

Con lo anterior, tenemos la posibilidad, (según las necesidades) de usar un mismo nodo en distintos

ámbitos y aprovecharlos todos con sus características al cien por ciento.

2. Nodo convencional (con PC)

Útil para aprovechar equipos de cómputo obsoletos comparados con aquellos que actualmente resuelven tareas más demandantes, pero que tendrían la capacidad de mantener operando un nodo IRLP y contribuyen al ahorro de recursos económicos, así como a incentivar el reciclaje de estos equipos. No requiere tener permanentemente un monitor ni teclado ni ratón ya que se puede acceder a su control y mantenimiento vía remota.

Las características mínimas del equipo de cómputo se enuncian a continuación.

- Procesador pentium II a 300 MHz
- 128 M de memoria Ram
- Tarjeta de red 10 base T
- Puerto paralelo

El equipo transceptor tampoco requiere características especiales y en este caso se utilizó el modelo FT-2600M de la marca Yaesu por ser con el que ya contaba la estación.

Siguiendo las indicaciones del manual de construcción física (cableado) y la instalación del sistema operativo linux distribución debian, del nodo IRLP [2] y del manual del radio [3], además de la adquisición de la tarjeta IRLP versión 3.0 (la versión más actual al momento del desarrollo del presente trabajo) y cuya interfaz de control con la PC es paralela, se construye y pone en operación el nodo, como se muestra en la figura 1. Cabe hacer mención que la distribución Debian utilizada para hacer funcionar el sistema IRLP es una recompilación del “kernel” del sistema operativo con código cerrado, por lo que no tenemos acceso a la programación de la tarjeta ni se tiene la posibilidad de reconfiguración del sistema, sin embargo es posible la puesta en marcha de “scripts” o pequeños programas, independientes del sistema IRLP que pueden ejecutarse directamente por el sistema operativo y que llevan a cabo tareas complementarias al sistema; ejemplo de estos “scripts” (ver listado 1) es el que permite la conexión y desconexión automática a nodos [4] que garantizan una permanente actividad del este prácticamente las 24 horas del día si así se desea, sólo basta identificar aquellos países y sus nodos que se muestren más activos en operaciones de radio en las horas adecuadas [5]; el único inconveniente relativo sería que tales actividades

pudieran corresponder a países como Rusia o Japón y sus respectivos idiomas, aún así resulta atractivo aprender unas cuantas frases que por lo menos permitan confirmar el contacto realizado.

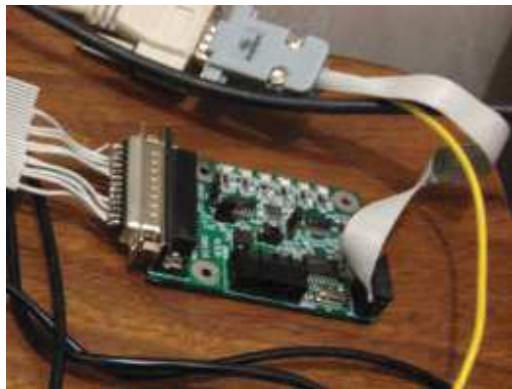


Figura 1. Nodo IRLP con interfaz para PC.

```
#--@XE2PTS Boletín Radio Club Satélite Viernes
    conecta a las 17:55 y desconecta a las 19:00
54 17 * * 5 (/home/irlp/scripts/end > /dev/null 2>&1)
55 17 * * 5 (/home/irlp/scripts/connect_to_reflector
    ref9209 &>/dev/null 2>&1)
00 19 * * 5 (/home/irlp/scripts/end > /dev/null 2>&1)
01 19 * * 5 (/home/irlp/scripts/connect_to_reflector
    ref9208 &>/dev/null 2>&1)
```

Listado 1. Ejemplo de programa o “script”

3. Nodo portátil (con “laptop”)

La construcción y puesta en marcha del nodo para PC no reviste mayores implicaciones que las asentadas en el manual correspondiente, sin embargo para ser implementado en una computadora portátil (laptop) moderna requiere algunas consideraciones adicionales.

Antes de pasar a esos detalles, se considera importante señalar que se pretende poner en marcha el nodo en una computadora portátil con las prestaciones modernas, para que el uso que se le pueda dar a ese equipo no sea exclusivo del nodo IRLP.

Con la premisa anterior, se encuentra que las computadoras contemporáneas ya no poseen de fábrica tarjeta de red para conexión por cable, y tampoco cuentan con puerto paralelo.

El reto consiste en adaptar el sistema de cómputo para que pueda operar con red inalámbrica

(Wi Fi) y dotarlo de un puerto paralelo. La primera parte fue resuelta editando el archivo /etc/network/interfaces eliminando cualquier referencia a redes cableadas y sin agregar referencias a Wi Fi.

Una vez que se ha logrado la conexión por radio (Wi Fi) a internet, se procede a resolver la adición de un puerto paralelo a la computadora portátil. Para ello existen múltiples opciones; en este caso particular se decidió por instalar una tarjeta externa con puerto paralelo por medio de la interfaz denominada ExpressCard que presenta la ventaja de tener una conexión directa al bus PCI Express de la computadora; sin embargo otra opción aún más general, puede ser la adición del puerto paralelo a través del puerto USB, ambas opciones se encuentran disponibles a través de la marca StarTech, modelo EC1PECPS [6], con la ventaja adicional que no se requieren controladores especiales para la versión Express Card.

A diferencia de una versión de sistema operativo distinto a Linux, no resulta necesario que el sistema reconozca la dirección del puerto paralelo como uno nativo, ya que Linux lo hace a través de una referencia al nombre del dispositivo como un archivo y no por su dirección física, de modo que prácticamente cualquier dispositivo que cumpla con el estándar IEEE 1284 funcionará.

El resto de los componentes y cables siguen siendo los mismos que los utilizados en el nodo con PC.

4. Nodo Móvil (con Raspberry Pi)

La Raspberry Pi es una placa computadora de bajo costo desarrollada en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

El diseño incluye un “System-on-a-chip” Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos Turbo para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM aunque originalmente al ser lanzado eran 256 MB. El diseño no incluye un disco duro o una unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación o carcasa. El 29 de febrero de 2012

la fundación empezó a aceptar órdenes de compra del modelo B, y el 4 de febrero de 2013 del modelo A.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora); y promueve principalmente el aprendizaje del lenguaje de programación Python, y otros lenguajes como Tiny BASIC, C y Perl [7]. La figura 2 muestra la Raspberry Pi con un resumen de sus principales periféricos.

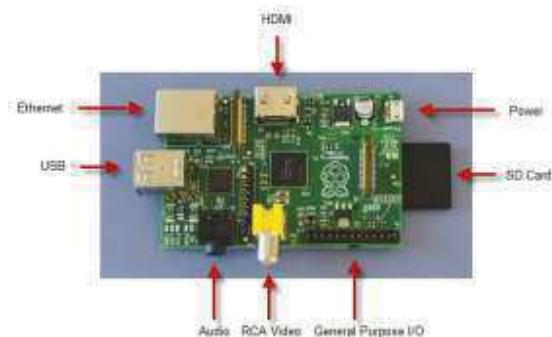


Figura 2. Placa base de la Raspberry Pi. Con flechas se indica la ubicación y nombre de los principales periféricos.

Para el caso de la construcción del nodo con Raspberry Pi, fue necesario realizar varias consideraciones preliminares. Como se menciona líneas arriba, la tarjeta no viene provista de fuente de voltaje por lo que se procedió a construir una que cumpliera con los requisitos de estabilidad en el voltaje y la capacidad suficiente de suministro de corriente. Se adquirió la tarjeta SD con un tamaño de 8 G y se le realizaron las acciones recomendadas de formateo con el programa SD Formatter para garantizar la compatibilidad; se le grabó a la tarjeta el instalador de sistema operativo denominado Raspberry NOOBS, un asistente de instalación y configuración que simplifica significativamente este proceso que debe realizarse ya desde la propia tarjeta.

Para continuar con el proceso de instalación se deben conectar a la tarjeta los periféricos que permitan interactuar con ella; específicamente un monitor a través del puerto HDMI, un teclado y la tarjeta de audio a través del par de puertos USB y la indispensable conexión a internet a través de su puerto Ethernet, que posteriormente puede ser sustituido por una tarjeta Wi Fi por USB si se desea.

Cabe mencionar que si bien la Raspberry Pi posee una salida de audio, no está provista de una entrada, de modo que resulta necesario la adición de

una tarjeta de audio con entrada y salida conectada a través del puerto USB por las características del proyecto.

Una vez terminada la instalación del sistema operativo según las instrucciones del manual correspondiente [8] se procede a realizar las modificaciones a la tarjeta IRLP para adaptarla a la interfaz de la Raspberry Pi, ya que su interfaz paralela es con conectores de poste a diferencia de la PC o Laptop que son con conector DB25 macho. También se le realiza un ajuste adicional a la propia tarjeta IRLP para que pueda operar con 3.3 V en lugar de los 5 V originales, pues la Raspberry Pi trabaja con dicho nivel de diferencia de potencial [9]. Las figuras 3, 4 y 5 muestran las adaptaciones del nodo IRLP para la Raspberry Pi.

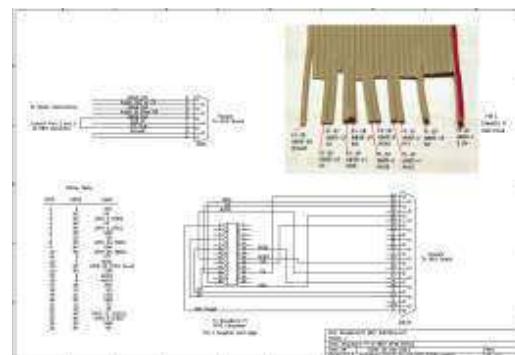


Figura 3. Adaptación del cable de la tarjeta IRLP para la interfaz paralela en la Raspberry Pi.

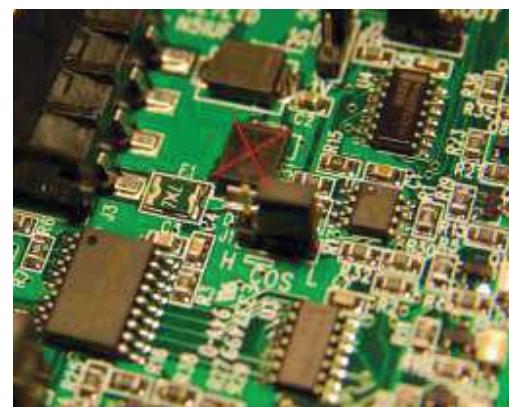


Figura 4. Remoción del diodo rectificador.

Potosí ha permitido incursionar en los modos digitales de transmisión de voz por internet (VoIP) rompiendo barreras de alcance de cobertura propias de las estaciones sin conexión a los servicios de internet.

Adicionalmente se han explotado las capacidades inherentes a los equipos de cómputo tanto los considerados obsoletos, como los más recientes y de altas prestaciones y aquellos emergentes con características particulares minimalistas para ofrecer ámbitos de aplicación extendida del uso del nodo IRLP, además de propiciar la experimentación con este sistema para seguir aportando aún más opciones útiles motivando a los alumnos, maestros y radio aficionados del ITS LP y ciertamente del mundo entero (gracias a internet) a practicar la actividad de la radio afición.

Concretamente, las contribuciones aportadas por la presente investigación fueron:

1. Dotar a la estación de radio del ITS LP (XE2PTS) con un sistema IRLP capaz de ser utilizado, según las necesidades como un nodo fijo, móvil o portátil, siendo que únicamente fue concebido de origen como un nodo fijo.
2. Se discuten los ámbitos de aplicación de cada una de las configuraciones, concluyendo que todas son útiles y se puede sacar el máximo provecho a un mismo nodo cambiando de plataforma según convenga a la aplicación que se le quiera dar en un momento determinado.
3. Extender la cobertura de alcance de los radios convencionales, agregándole al sistema de comunicaciones la capacidad de transmisión y recepción de voz por internet (VoIP).

Agradecimientos

Los autores agradecen el apoyo financiero otorgado por el consorcio CEMIE-Solar a través del proyecto 32.

Se agradece también todas las facilidades otorgadas por las autoridades del Instituto

Tecnológico de San Luis Potosí para la operación y mantenimiento de su estación de radio aficionados.

Referencias

- [1] Diario Oficial de la Federación, edición del 5 de Octubre de 1953.
- [2] Cameron David (VE7LTD) “IRLP Board Instalation Manual” 2006.
- [3] Vogl P. et al “YAESU FT-2600M Operating Manual”. VERTEX Standard Co., LTD, 2001.
- [4] www.qsl.net/xe1dvi/scripts/customcrons.html (septiembre 2014).
- [5] <http://status.irlp.net/> (septiembre 2014).
- [6] www.startech.com (septiembre 2014).
- [7] www.raspberrypi.org (septiembre 2014).
- [8] Cameron David: “PiRLP Owner’s Manual”, 2010.
- [9] <http://www.crompton.com/hamradio/irlp/PiIRLP.html>

Diseño Mecatrónico en FPGA del Control de Posición de un Elemento Servoneumático

Arroyo Chávez Víctor Ventura¹, Ramírez Rosales Salvador¹, Francisco Hernández-Hernández¹, José-Emilio Vargas-Soto² y Ramos Arreguín Juan Manuel²

¹ Universidad Tecnológica de San Juan del Río

² Universidad Autónoma de Querétaro, Facultad de Informática

jramos@mecamex.net

Resumen

El presente trabajo aborda el diseño mecatrónico en un FPGA para el control de posición de un elemento servoneumático, a partir del control de flujo de aire, así como su descripción en lenguaje VHDL para implementarse en un FPGA, que será la unidad de control que habrá de aplicarse a un robot poliarticulado que está constituido básicamente de la parte mecánica, es decir, su estructura, y la parte motriz que lo conforman dos actuadores neumáticos (2 grados de libertad). Teniendo en cuenta la alta no linealidad del aire para el control de posición de los pistones, actualmente se tienen en el mercado sistemas de control para elementos servoneumáticos, pero con un alto costo de inversión. Es por ello que se aborda esta problemática, el cual, no requiere de un presupuesto elevado, solamente de un control definido y preciso, es decir, de un control PID. Se desarrolló el controlador PID en VHDL y se implementó junto con las tarjetas electrónicas de control, así como los elementos de soporte para los motores y la válvula neumática diseñadas en SolidWorks y materializadas en impresión 3D; logrando el control de posición de los actuadores en 3 puntos (reposo, medio avance y fin de carrera).

Palabras clave: Servomecanismo, encoder óptico, controlador PID, VHDL, robot poliarticulado, impresión 3D.

1. Introducción

Actualmente existen en el mercado sistemas de control para elementos servoneumáticos con un alto costo de inversión, sin embargo, haciendo uso de herramientas como lo son el diseño de descripción de hardware empleando VHDL (Very High Speed Integrated Circuit Hardware Description Language), desarrollo de tarjetas electrónicas y con el empleo de diseños realizados en CAD y manufacturados en impresión 3D entre otros, se podría reducir el coste

en lo referente a un sistema eficaz de control para elementos servoneumáticos en procesos de manufactura dentro de la industria.

Un detalle a destacar en el sistema de control de un elemento servoneumático, es la alta no linealidad del aire y con ello, se hace complicado poder establecer un control con algún método tradicional, pues no respondería a los parámetros requeridos en procesos delicados donde se les requiera.

A continuación se describe lo realizado en este trabajo, que involucra al brazo poliarticulado de 2 grados de libertad, los accesorios desarrollados en SolidWorks maquinados en impresión 3D y el desarrollo del controlador PID en VHDL, donde la medición de su posición se efectúa por medio de los encoders ZSD1000A. .

2. Diseño del controlador PID en VHDL

El brazo cuenta con dos grados de libertad donde se ubican los actuadores neumáticos (1 pistón y un motor neumático), uno para cada grado de libertad como se aprecia en la siguiente imagen.



Figura 1. Brazo Poliarticulado

La base del brazo constituye al primer grado de libertad, donde se ubica el motor neumático; a su vez, el pistón que se muestra en la imagen, es el que se encarga de darle movilidad al segundo grado de libertad. Éste es el brazo al cual se le implementa la descripción en VHDL del controlador PID en FPGA.

2.1 PID en VHDL

El diseño del controlador se desarrolló bajo el lenguaje VHDL para ser implementado dentro del FPGA de la tarjeta Nexys2. La descripción del controlador PID se compone de dos partes principales, la primera parte que corresponde a la adquisición de la posición del brazo, a través de las señales de un encoder para producir la señal de retroalimentación hacia el controlador y la otra parte que es la que conforma al controlador PID que se encargará de comparar la señal de retroalimentación de la posición del brazo (a través del encoder) con el valor de consigna del controlador.

Para la descripción del controlador PID, se utilizó el siguiente diagrama de bloques, donde se recibe la señal “C” del ContadorIncDec como indicador de la posición actual, que corresponde a la señal de entrada “X” del controlador, a su vez se tiene la variable SP (setpoint, valor de consigna del sistema) ambas señales se juntan en un punto de suma para después pasar a las etapas del controlador PID, donde se involucran las señales de las constantes fijas del control proporcional (Kp), derivativo (Kd) e integral (Ki); obteniendo a la salida la señal “Y”. A esto le antecede el proceso de adquisición de la señal del encoder que indica la secuencia de giro, el conteo de pulsos para determinar la posición de avance/giro del actuador para que pueda ser interpretada por el módulo del controlador PID que a su vez, le da la información al bloque PWM que es el que envía los pulsos para activar las electroválvulas de los actuadores y así poder posicionar al brazo en función de los valores establecidos en el punto de ajuste (Sp) con los valores de las constantes del controlador (Kp, Kd y Ki).

El diagrama del PID, también muestra en la parte superior derecha dos bloques que se encargan de procesar los pulsos del encoder (para poder mostrar la lectura de los pulsos a través de los cuatro displays de 7 segmentos con que cuenta la tarjeta Nexys2) cuando avanza el actuador y así poder tener una ayuda visual del momento en que hay actividad en los actuadores cada que se le indica una nueva posición por parte del usuario.

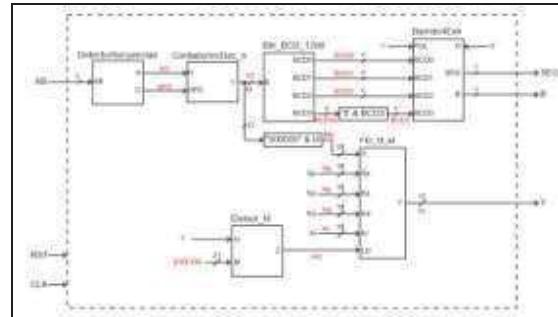


Figura 2 Diagrama del PID.

2.2 Elementos auxiliares

En cuanto a los motores de cd que se emplearon para la apertura y cierre de las válvulas reguladoras de flujo de aire que alimentan a los actuadores, se diseñaron unas guardas en SolidWorks que posteriormente se maquinaron bajo impresión 3D para tener un mayor soporte en la fijación de estos elementos.

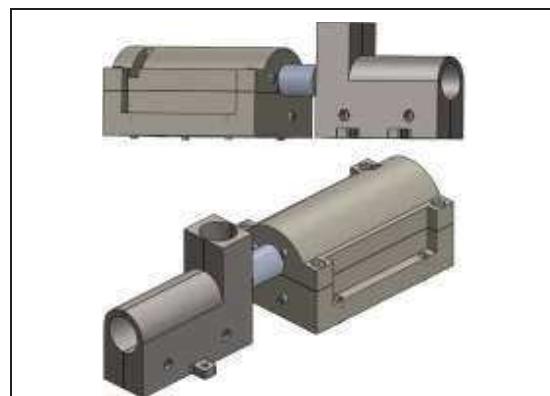


Figura 3. Diseño de accesorios en SolidWorks.



Figura 4. Diseños maquinados en impresión 3D

3. Análisis de resultados

Una vez que se probaron uno a uno los diferentes bloques que constituyen al programa de controlador PID y sabiendo el comportamientos de los sistemas de control neumático en comparación con los sistemas de control eléctrico [1] se procedió a realizar un solo archivo con la información de todos los componentes para llevar a cabo las pruebas, para lo cual, se implementó en un solo pistón para probar físicamente el controlador desarrollado, logrando posicionar al pistón en tres puntos establecidos por software a través de una señal externa de la tarjeta Nexys2, estos puntos posicionaban al pistón en la posición 1 (cuando el pistón estaba en reposo), posición 2 que corresponde cuando el pistón se encontraba a media carrera de avance y la posición 3, que es cuando el pistón se encuentra en su fin de carrera. Cabe mencionar que al inicio del ajuste, se tenía como resultado un estado oscilante al momento de que el pistón llegaba a su posición de ajuste, pero una vez ajustadas las constantes del controlador, se pudo reducir en un 80% la oscilación del pistón una vez llegada a su posición de ajuste, en las siguientes figuras se muestra la integración de todo el sistema que incluye la descripción del controlador implementado en la tarjeta Nexys2, las tarjetas electrónicas de adquisición de las señales del encoder y el control de las solenoides de las electroválvulas así como del control de los motores que regulan la apertura de la válvula de regulación de flujo de aire del pistón.



Figura 5. Elementos del sistema.

Los elementos del sistema como se muestran en la figura 5, son los siguientes:

- 1.- Tarjeta Nexys2.

- 2.- Tarjeta de puente H para control del motor.
- 3.- Tarjeta de adquisición de señales del encoder y control de electroválvulas.
- 4.- Motorreductor con guarda (con impresión 3D).
- 5.- Válvula de regulación de flujo de aire (con guarda impresa en 3D).
- 6.- Electroválvula 5/3 con centro cerrado.

Las siguientes figuras muestran la ejecución del controlador PID en las posiciones 2 y 3 para el pistón.



Figura 6. Posición del pistón a media carrera.



Figura 7. Posición del pistón a final de carrera.

4. Conclusiones

Con este trabajo se logra tener el control de un elemento servoneumático para un sólo elemento actuador de posición (un pistón). Ahora lo único que quedaría por hacer, es reproducir lo hecho hasta ahora, por el número de grados de libertad que se desee expandir al brazo en un futuro, dentro de la misma arquitectura que se tiene. Por el momento el

control del brazo se realiza por medio de 3 botones pulsadores para indicarle al brazo en qué posición se desea ubicar al brazo, si requiere modificar los valores de las constantes del controlador, se tendrá que reprogramar el FPGA.

Ahora lo que sigue con respecto a este trabajo es, implementarlo a todo el brazo y crear una interfaz con la PC aprovechando los periféricos de RS232 con que cuenta la tarjeta Nexys2, donde desde la PC se pueda ubicar al brazo y también se ingrese directamente los valores para cada una de las constantes del controlador; sustituir los encoders por

giroscopios para detectar la posición de cada eslabón (grado de libertad).

El desarrollo de este trabajo ha aprovechado equipo adquirido con apoyo PROMEP, así como con apoyo de infraestructura de CONACYT.

Referencias

- [1] Jiménez A., M Ai B. “*Robot Manipulators Trends and Development*”, Hard Cover, Croacia, 1^a edición, 2010.

Construcción y Control de un Péndulo Invertido sobre un Carro (PISC) con Control Esclavo de Corriente por Computadora

Espíndola- López Eduardo, Carrillo-Serrano Roberto Valentín y Rivas-Araiza Edgar Alejandro

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Centro Universitario, Cerro de las Campanas s/n.
C.P. 76010, Santiago de Querétaro, Qro. México

Resumen

En éste trabajo se describe la implementación de un lazo de control esclavo de corriente para controlar un prototipo de un péndulo invertido sobre un carro, que tiene como actuador principal, un motor de CD con escobillas e imán permanente. El controlador emplea tres estrategias de control: el primer controlador es para levantar al péndulo y está basado en la energía del sistema; el segundo es para atrapar al péndulo y consiste en una retroalimentación de estado. El tercer controlador es para producir la fuerza que los controladores anteriores demandan y consiste en un control proporcional-integral (PI) de corriente. Los tres controladores son programados con lenguaje C++ en una computadora portátil con interfaz serial RS-232 para la comunicación con un microcontrolador comercial. Finalmente, se muestran los resultados experimentales.

Palabras clave: péndulo invertido sobre un carro, control proporcional-integral (PI) de corriente, energía del sistema, retroalimentación de estado.

1. Introducción

El control de corriente por computadora no es muy común debido a que aún se tiene la idea de que la corriente eléctrica es una variable muy rápida para ser codificada y procesada; por lo que el presente trabajo propone un control esclavo de corriente por computadora en un sistema PISC real.

Con la idea de codificar la corriente eléctrica, en [1] se utiliza el control digital predictivo para controlar la corriente eléctrica en el inductor de un convertidor tipo Boost; el control digital le da la posibilidad de modificar directamente la corriente a través de la interrupción de un transistor, por lo cual el sistema es más flexible. Por otro lado, en [2] se codifica la referencia de voltaje en el control digital de un convertidor DC-DC, en el cual además de intervenir en el voltaje, utiliza un lazo esclavo de corriente para mejorar la eficiencia del sistema.

Los sistemas no lineales se controlan más efectivamente a partir del control digital; un ejemplo de ello es el PISC que es una herramienta para implementar y evaluar nuevas propuestas de controladores para sistemas no lineales [3]. En [4] se propone un regulador lineal cuadrático (LQR) programado en una computadora, para llevar al péndulo del punto de equilibrio estable al punto de equilibrio inestable con un control de posición del carro que hace entrar al péndulo en resonancia, y al entrar a la posición vertical conmuta a un esquema de regulación del péndulo.

El prototipo PISC representa desafíos para los ingenieros en control; para enfrentarlos en [5] se propone un controlador proporcional-integral-derivativo (PID) robusto para el PISC, en el que se menciona que el problema de la no linealidad del péndulo invertido puede ser resuelto simplemente con dos PID's clásicos de tal manera que uno controle la posición del carro y otro el ángulo del péndulo, generando entre los dos una señal de control para el PISC.

De manera similar, en [6] se propone un controlador de modo deslizante disociado (DSMC) para la estabilización del PISC en el que utilizan una técnica de optimización llamada: enjambre de partículas multi-objetivo, para optimizar los parámetros del DSMC.

En la búsqueda de un controlador estable, efectivo y robusto; en [7] se propone un controlador PI en el espacio de estados de forma fraccional para el control de un PISC; proponen un método para descomponer el polinomio característico del sistema en una forma fraccional, de tal manera que se pueda ver como un sistema de primer orden; consiguiendo así un control más confiable.

Por otro lado, en [8] se proponen un controlador de lógica difusa tipo 2 de intervalo (IT2-FLC) para controlar un PISC; el cual se basa en las reglas difusas Mamdani, con función de pertenencia triangular y el método Wu-Mendel de incertidumbre vinculada para aproximar el conjunto de tipo reducido, con el fin de minimizar las incertidumbres numéricas y lingüísticas asociadas a las entradas y salidas de un sistema controlado con lógica difusa.

El control por computadora ayuda a facilitar sistemas como el PISC; sin embargo, en [9] se menciona que existe una separación entre la fase del diseño del controlador y su implementación en la computadora de forma que, cada fase asume consideraciones sobre la otra fase, lo cual lleva a un comportamiento inesperado o incorrecto del sistema a controlar; así que propone el estudio de los retardos en el sistema de control y analiza algunas técnicas para la reducción de los mismos.

Como se puede observar, para sistemas no-lineales e inestables no siempre es suficiente con un controlador que lo estabilice; también se requiere de algún método secundario, con la intención de lograr un control más seguro; sin perder de vista que cualquier algoritmo secundario, no debe afectar en el tiempo de ejecución del controlador principal; para no caer en retardos como se menciona en [9]. En éste sentido, se propone un control esclavo de corriente para un PISC, tomando como base el modelo matemático utilizado en [10]; calculado a partir de las ecuaciones de la dinámica de Lagrange.

2. Modelo dinámico del PISC

El modelo dinámico del PISC (ver Fig. 1) se obtiene aplicando las ecuaciones de Euler-Lagrange, con las cuales, siguiendo la metodología de análisis de [10] para un sistema PISC, sin considerar los

efectos de fricción es posible llegar al modelo compacto de la ecuación (1).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \quad (1)$$

$$M(q) = \begin{bmatrix} M+m & ml\cos(\theta) \\ ml\cos(\theta) & ml^2 + I \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad (2)$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & -ml\dot{\theta}\sin(\theta) \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad (3)$$

$$G(q) = \begin{bmatrix} 0 \\ -mgl\sin(\theta) \end{bmatrix} \in \mathbb{R}^{2 \times 1} \quad (4)$$

$$q = \begin{bmatrix} x \\ \theta \end{bmatrix} \in \mathbb{R}^2 \quad (5)$$

$$\tau = \begin{bmatrix} f \\ 0 \end{bmatrix} \in \mathbb{R}^2 \quad (6)$$

Donde M y m son las masas del carro y el péndulo respectivamente, l es la longitud de un extremo del péndulo a su centro de masa, g es la aceleración de la gravedad (9.81 m/s^2), I es la inercia del péndulo, θ es la posición angular del péndulo, x es la posición del carro y f es la fuerza aplicada al carro.

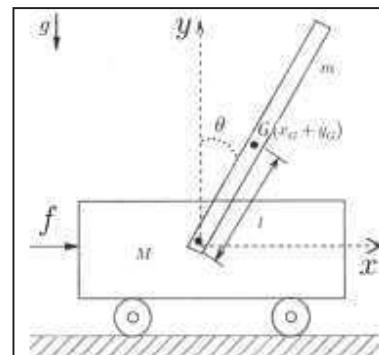


Fig. 1. Péndulo invertido sobre un carro.

En éste sentido, $M(q)$ es la matriz de inercia (simétrica y definida positiva), $C(q,\dot{q})$ es la matriz de efectos centrífugos y de Coriolis, $G(q)$ es el vector de pares gravitacionales y τ es el vector de entradas.

Siguiendo con el análisis del modelo dinámico, se definen las siguientes constantes:

$$a_1 = \frac{m^2 l^2}{ml^2 + I} \quad (7)$$

$$a_2 = M + m \quad (8)$$

Combinando los dos renglones de la ecuación (1) es posible obtener la ecuación de estado del PISC como:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \in \mathbb{R}^4 \quad (9)$$

$$\dot{y} = h(y, f) = \frac{d}{dt} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} =$$

$$\begin{bmatrix} y_2 \\ \frac{-a_1 g \sin(y_3) \cos(y_3) + m l y_4^2 \sin(y_3) + f}{a_2 - a_1 \cos^2(y_3)} \\ y_4 \\ \frac{a_1 a_2 g \sin(y_3) - a_1 m l y_4^2 \sin(y_3) \cos(y_3) - a_1 f \cos(y_3)}{m l (a_2 - a_1 \cos^2(y_3))} \end{bmatrix} \in \mathbb{R}^4 \quad (10)$$

Los puntos de operación del sistema se obtienen al hacer $\dot{y} = h(y^*, f^*) = 0$, con lo cual se deduce que $y_2^* = 0$, $y_4^* = 0$, $f^* = 0$, $y_3^* = n\pi$, $n \in \mathbb{Z}$ y $y_1^* = c$, $c \in \mathbb{R}$. Entonces, el punto de operación del sistema en el cual se desea trabajar se define como:

$$y^* = 0 \in \mathbb{R}^4, \quad f^* = 0 \quad (11)$$

3. Control para levantar el péndulo

El controlador para levantar al péndulo se deduce a partir de la siguiente función de Lyapunov (ec. (12)) propuesta por [10]:

$$V(q, \dot{q}) = \frac{k_e}{2} E(q, \dot{q})^2 + \frac{k_v}{2} \dot{x}^2 + \frac{k_x}{2} [x - x^*]^2 \quad (12)$$

Donde k_e , k_v y k_x son constantes positivas. Y $E(q, \dot{q})$ es la energía total del sistema; es decir, la suma de la energía cinética más la energía potencial:

$$E = \frac{1}{2} (M + m) \dot{x}^2 + m l \dot{\theta} \cos(\theta) + \frac{1}{2} (I + m l^2) \dot{\theta}^2 + m g l (\cos(\theta) - 1) \quad (13)$$

Derivando la ecuación (12) se tiene que:

$$\dot{V} = \dot{x} \left[f \left(k_e E + \frac{k_v}{a_2 - a_1 \cos^2(\theta)} \right) + \frac{k_v \sin(\theta) (-a_1 g \cos(\theta) + m l \dot{\theta}^2)}{a_2 - a_1 \cos^2(\theta)} + k_x (x - x^*) \right] \quad (14)$$

Siguiendo el análisis de [10], se elige el segundo factor de la ecuación (14) de tal forma que \dot{V} sea:

$$\dot{V} = -k_\delta \dot{x}^2 \quad (15)$$

Donde k_δ es una constante positiva.

La ecuación (15) es posible siempre y cuando se cumpla que:

$$f = \frac{k_v \sin(\theta) (a_1 g \cos(\theta) - m l \dot{\theta}^2) - (a_2 - a_1 \cos^2(\theta)) (k_x (x - x^*) + k_\delta \dot{x})}{k_v + (a_2 - a_1 \cos^2(\theta)) k_e E} \quad (16)$$

Como se observa en la ecuación (14), la ecuación (16) solamente se puede definir de esa forma si se cumple que:

$$\left(k_e E + \frac{k_v}{a_2 - a_1 \cos^2(\theta)} \right) \neq 0 \quad (17)$$

Analizando las dos posibilidades en las cuales, la parte izquierda de la ecuación (17) es diferente de cero, se puede concluir que la única forma de cumplir con la ecuación (17) es verificando que:

$$\frac{k_v}{k_e} > 2 m g l / a_2 \quad (18)$$

Así, cumpliendo con la condición (18) y aplicando la fuerza de la ecuación (16) al carro del PISC, se asegura que la función de Lyapunov siempre va a tener pendiente negativa (ec. (15)) ó cero si la velocidad del carro es cero. Entonces, conforme pase el tiempo la función de Lyapunov tratará de llegar a cero y para que esto suceda se requiere que $\dot{x} = 0$, $\dot{\theta} = 0$, $x = x^*$ y $\theta = 0$. Por lo tanto, el controlador no lineal representado por la ecuación (16) es capaz de llevar al péndulo a su posición vertical en $\theta = 0$.

4. Control para atrapar el péndulo

El controlador para levantar el péndulo se deduce a partir de la aproximación lineal del sistema (10) en el punto de operación (11), la cual está dada como:

$$\dot{z} = Az + Bw \quad (19)$$

$$A = \frac{\partial h(y, f)}{\partial y} \Bigg|_{\begin{matrix} y^* \\ f^* \end{matrix}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-a_1 g}{a_2 - a_1} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{a_1 a_2 g}{m l (a_2 - a_1)} & 0 \end{bmatrix} \quad (20)$$

$$B = \frac{\partial h(y, f)}{\partial f} \Bigg|_{\begin{matrix} y^* \\ f^* \end{matrix}} = \begin{bmatrix} 0 \\ \frac{1}{a_2 - a_1} \\ 0 \\ \frac{-a_1}{m l (a_2 - a_1)} \end{bmatrix} \quad (21)$$

Donde:

$$z = y - y^* \quad (22)$$

$$w = f - f^* \quad (23)$$

Analizando la matriz de controlabilidad $C = [B | AB | A^2 B | A^3 B]$ del par (A, B) , se tiene que:

$$\det(C) = \frac{g^2 a_1^4}{m^4 l^4 (a_1 - a_2)^4} > 0 \quad (24)$$

Por lo tanto, el sistema es controlable y es posible asignar de manera arbitraria los valores propios de la matriz de lazo cerrado $(A - BK^T)$ a través del controlador $w = -K^T z$. Entonces el nuevo controlador para atrapar el péndulo es:

$$f = w = -K^T z = -K^T y \quad (25)$$

5. Control PI de corriente

Debido a que el actuador del prototipo PISC es un motor de CD con escobillas e imán permanente, se requiere de un control de corriente para representar de forma eficiente, la fuerza de los controladores anteriores. En este sentido, la señal de control se calcula como:

$$u_i = k_{pi}(i_d - i) + k_{ii} \int (i_d - i) dt \quad (26)$$

Donde k_{pi} y k_{ii} son constantes positivas que representan las ganancias proporcional e integral del controlador respectivamente. Y las corrientes deseada y medida de la armadura del motor son i_d e i respectivamente.

Se considera el siguiente modelo matemático del motor obtenido de [11], representado como una función de transferencia con entrada de voltaje y salida de corriente:

$$\frac{I(s)}{U_i(s)} = \frac{J_m s + b_m}{(Ls + R)(J_m s + b_m) + k_s k_m} \quad (27)$$

Donde J_m es la inercia del rotor del motor, b_m es la fricción viscosa del motor, L es la inductancia de armadura, R es la resistencia de armadura, k_m es la constante de par del motor ($T = k_m i$) y k_s es la constante de fuerza contra-electromotriz; es decir, para un motor con fuerza electromotriz e_a y posición angular ϕ , entonces: $e_a = k_s \dot{\phi}$.

Cerrando el lazo de control para una corriente deseada $I_d(s)$ considerando el controlador de la ecuación (26) y como planta la ecuación (27) (con retroalimentación unitaria), se observa que la función transferencia de lazo cerrado tiene el siguiente polinomio característico:

$$P(s) = L J_m s^3 + (J_m (R + k_{pi}) + b_m L) s^2 + (b_m (R + k_{pi}) + k_s k_m + k_{ii} J_m) s + k_{ii} b_m \quad (28)$$

Si se analiza el polinomio de la ecuación (28) con el criterio de estabilidad de Routh se observa fácilmente que todos los elementos de la primera columna en la tabla del criterio son positivos; por lo tanto, no hay cambios de signo entre esos elementos y se deduce que no hay raíces con parte real positiva en el polinomio (28), es decir, el sistema en lazo cerrado con el control PI de corriente es estable para cualquier valor de las ganancias k_{pi} y k_{ii} .

Por otro lado, se observa que el sistema de control en lazo abierto es un tipo 1 debido al término integral; por lo cual, se verifica que el error en estado estacionario es cero para cualquier entrada escalón [11].

6. Prototipo experimental

En la Fig. 2 se muestra un diagrama de bloques funcional del prototipo experimental que se construyó; este consiste en un sistema mecánico (A), del cual se obtiene la posición angular del péndulo con un codificador incremental (B) modelo S1-1000-250-I-B-D de la marca US DIGITAL, que cuenta con 1000 ppr. La posición del carro se mide con un sensor infrarrojo (C) modelo GP2Y0A21YK0F de la marca SHARP con resolución de [10-80] cm. El sistema mecánico es accionado por un motor de CD (D) modelo R11-1450 de la marca Hp de 24 Vcd y 2 A, al

cual se le mide la corriente de armadura con un sensor de corriente (E) modelo ACS712 de la marca Allegro con resolución de [-5 - +5] A.

La base del funcionamiento del prototipo es una tarjeta electrónica (F) con un microcontrolador modelo PIC16F877A de la empresa Microchip que adquiere las señales de los sensores y las envía a una computadora portátil por medio de una interfaz serial (G) RS-232. La computadora (H) calcula los algoritmos de control y regresa una variable de 8 bits (un bit de signo y 7 bits de magnitud) para que la tarjeta electrónica genere una señal de PWM (Pulse Width Modulation, I) que es utilizada por un puente H (J) modelo LMD18200 de la empresa Texas Instruments, el cual suministrar el voltaje del motor.

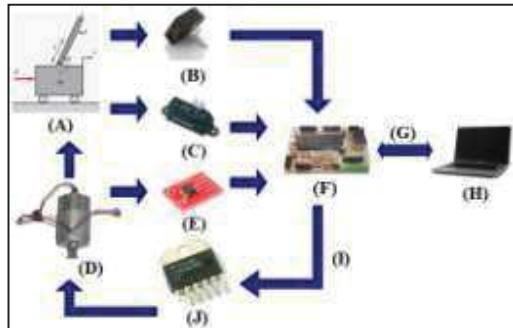


Fig. 2. Diagrama funcional del prototipo.

Los parámetros del prototipo se muestran en la Tabla 1, con los cuales se encuentran las ganancias del controlador lineal para atrapar el péndulo y se puede verificar la condición (18) para sintonizar el controlador no lineal.

Tabla 1. Parámetros del PISC.

Parámetro	Valor	Unidad
l	0.0865	m
M	0.4735	kg
m	0.01055	kg
I	0.0000263	$kg \cdot m^2$
k_m	0.0637	$\frac{N \cdot m}{A}$

Asignando los valores propios de la matriz $(A - BK^T)$ en $\lambda_1 = -15$, $\lambda_2 = -11$, $\lambda_3 = -2$ y $\lambda_4 = -1.5$ se obtiene la matriz transpuesta K^T del controlador para atrapar el péndulo como:

$$K^T = [-2.771 \ -3.669 \ -19.291 \ -2.043] \quad (29)$$

El controlador para levantar el péndulo se sintoniza de forma empírica, siempre y cuando se

cumpla la condición (18). Los valores de las ganancias se muestran en la Tabla 2.

Tabla 2. Ganancias del controlador no lineal.

Ganancia	Valor
k_v	0.1
k_x	55.5
k_δ	1.55
k_e	8.1

Para encontrar las ganancias del control PI de corriente se requieren pruebas con el prototipo real; en las cuales se ajustan las ganancias hasta que la corriente en el motor pueda seguir una corriente deseada de forma senoidal con una frecuencia y amplitud suficientemente grandes. Para una corriente deseada senoidal de 20 Hz y 0.15 A de amplitud se encontraron las ganancias de la Tabla 3.

Tabla 3. Ganancias del control PI de corriente.

Ganancia	Valor
k_{pi}	2.9
k_{ii}	1800.0

El prototipo construido se muestra en la Fig. 3.

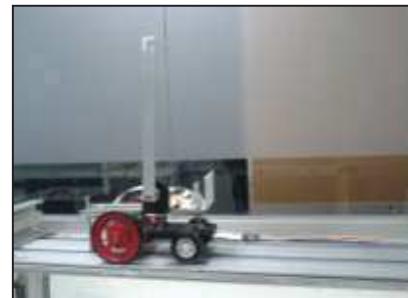


Fig. 3. Fotografía del prototipo PISC real.

7. Análisis de resultados

En la Fig. 4 se muestran los resultados experimentales obtenidos. Como se puede observar, el control para levantar al péndulo está en funcionamiento durante 2 s; en éste tiempo la posición del carro oscila y crece alrededor de 0 al igual que su velocidad.

Mientras que la posición angular del péndulo comienza a crecer oscilando alrededor de π con una velocidad que aumenta bastante rápido. Por otro lado,

la corriente deseada (en rojo) se satura estrepitosamente el mismo número de veces que el voltaje del motor y que las oscilaciones de la posición del péndulo. Además se observa que la corriente medida al motor (en azul) sigue de cerca a la corriente deseada. Cuando el controlador lineal es

activado a los 2 s; la posición del carro regresa a 0 al igual que su velocidad y la velocidad del péndulo; justo al mismo tiempo en que la posición del péndulo llega a 0. Lo cual se obtiene gracias a las señales reflejadas en la corriente y el voltaje del motor.

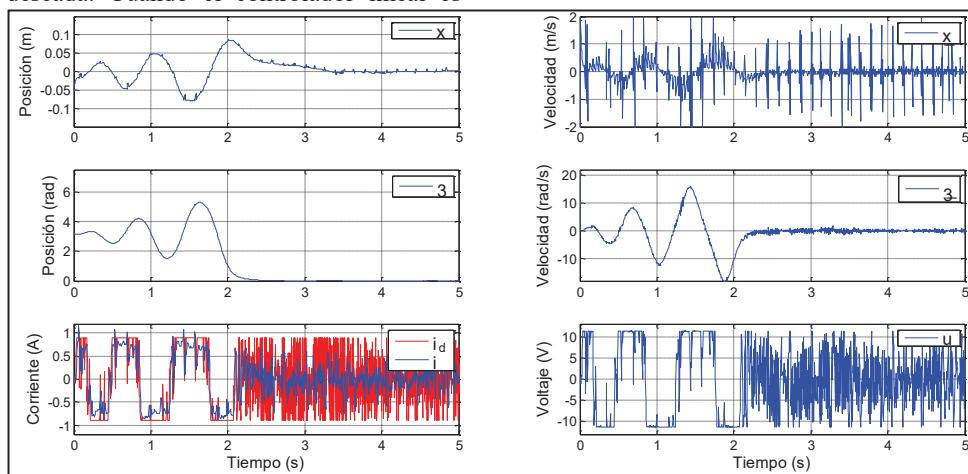


Fig. 4. Graficas de respuestas del PISC.

8. Conclusiones

La corriente en el motor es capaz de seguir simultáneamente a la corriente deseada implementando el algoritmo de control PI de corriente en una computadora, tal como se esperaba y a pesar de los cambios bruscos de alta frecuencia en la señal de control. Por otro lado, se verificó que los controladores para levantar y atrapar el péndulo funcionan a la perfección aunque se desprecien los efectos de la fricción en el modelo matemático. Se puede concluir que es posible controlar un PISC en su posición inestable, asegurando la fuerza aplicada al carro por medio de un control PI de corriente programado en una computadora.

Referencias

- [1] P. Athalye, D. Maksimovic, and R. Erickson, “Variable-frequency predictive digital current mode control”, Power Electron. Lett. IEEE, vol. 2, no. 4, pp. 113–116, 2004.
- [2] A. Kelly and K. Rinne, “Sensorless current-mode control of a digital dead-beat DC-DC converter”, in Applied Power Electronics Conference and Exposition, 2004. APEC’04. Nineteenth Annual IEEE, 2004, vol. 3, pp. 1790–1795.
- [3] K. Åström and K. Furuta, “Swinging up a pendulum by energy control”, Automatica, vol. 36, pp. 287–295, 2000.
- [4] F. Castaños, R. Carrera, and D. Coyoacan, “Levantamiento y control de un péndulo invertido con un esquema de control reconfigurable”, Inst. Ing. UNAM AP, 2004.
- [5] B. Subudhi, A. Ghosh, and T. R. Krishnan, “Robust proportional integral derivative compensation of an inverted cart pendulum system: an experimental study”, IET Control Theory Appl., vol. 6, no. 8, pp. 1142–1152, 2012.
- [6] M. J. Mahmoodabadi, S. Arabani Mostaghim, A. Bagheri, and N. Nariman zadeh, “Pareto optimal design of the decoupled sliding mode controller for an inverted pendulum system and its stability simulation via Java programming”, Math. Comput. Model., vol. 57, no. 6, pp. 1070–1082, 2013.
- [7] M. Bettayeb, C. Boussalem, R. Mansouri, and U. M. Al Saggaf, “Stabilization of an inverted pendulum car system by fractional PI state feedback”, ISA Trans., vol. 53, no. 2, pp. 16–508, 2014.
- [8] A. M. El-Nagar, M. El-Bardini, and N. M. EL-Rabaie, “Intelligent control for nonlinear inverted pendulum based on interval type 2 fuzzy PD controller”, Alex. Eng. J., vol. 53, no. 1, pp. 23–32, 2014.
- [9] P. Balbastre, M. Lluesma, and I. Ripoll, “Análisis y compensación de los retardos de

Ingeniería Mecatrónica. Proyectos y Aplicaciones. Capítulo 4, pp.16-22, Octubre 2014.
® Asociación Mexicana de Mecatrónica A.C
ISBN: 978-607-9394-00-4

- planificación en sistemas de control”, RIAII,*
vol. 3, no. 2, pp. 40–49, 2010.
- [10] I. Fantoni and R. Lozano, *Non-linear control for underactuated mechanical systems*. Great Britain: Springer,, 2002.
- [11] V. M. Hernández-Guzmán, R. Silva-Ortigoza, and R. V. Carrillo-Serrano, *Control Automático: Teoría de Diseño, Construcción de Prototipos, Modelado, Identificación y Pruebas Experimentales*. México D.F.: COLECCIÓN CIDETEC, 2013.

Desarrollo de Aplicación Domótica con Comunicación Inalámbrica Bluetooth

Sotelo Martínez Samuel¹, Olivo Flores Marco Antonio¹ y Rodríguez Ortiz Juan Gabriel².

¹ Ingeniería Mecatrónica. Universidad Tecnológica de San Juan del Río. Av. La Palma No. 125, Col. Vista Hermosa, C. P. 76806. San Juan del Río, Querétaro. Tel. (427) 129 2000.

² Ingeniería Electrónica. Instituto Tecnológico de San Juan del Río. Av. Tecnológico No. 2, Col. Centro, C. P. 76800. San Juan del Río, Querétaro. Tel. (427) 272 4118.

Resumen

La domótica es un área relativamente nueva, de elevados costos. Incorpora nuevas tecnologías en los edificios, en particular las que tienen que ver con la automatización, el ahorro energético, la seguridad y el acceso a la información, para funciones de ocio, entretenimiento y confort [1]. El diseño de la aplicación con tecnología bluetooth es una propuesta para la automatización doméstica haciendo uso de medios inalámbricos reduciendo el costo y haciendo mas accesible la domótica a personas que mas lo necesiten como lo son las personas con capacidades diferentes.

La aplicación domótica con tecnología bluetooth implica distintas etapas como: un dispositivo móvil que puede ser un teléfono móvil, un PDA (Asistente Personal Digital) o un ordenador, en todos los casos el dispositivo debe contar con un sistema operativo Android y la aplicación App Inventor la cual se programará para poder generar una señal por Bluetooth con intensidad suficiente para ser captada a cierta distancia por el receptor HC-06 que a su vez interactúa con un controlador Arduino Mega que proporcionará las señales que llevarán a cabo las acciones de control para la automatización deseada.

Palabras clave: Domótica, Automatización inalámbrica doméstica, Tecnología bluetooth.

1. Introducción

El término domótica proviene de la unión de las palabras: *domus* (del latín, casa) y *tica* (de automática, en griego) y que literalmente significa: “casa automática” [2]. La domótica se ha convertido en un área de oportunidad cada vez más aceptada por

la sociedad; así, el concepto de control inteligente aplicado a grandes edificios donde se incluyen cámaras de videovigilancia, sistemas de audio ambiental, sistemas de iluminación, redes informáticas, entre otras, obliga a renovar la idea de casa-habitación y/o edificios de oficina [3].

La tecnología domótica instalada actualmente en México está en proceso de aceptación y crecimiento y generalmente se utilizan sistemas alámbricos de control. Por tanto, el proponer sistemas alternos aplicados a la integración de diferentes tecnologías en el hogar u oficinas que generen confort, seguridad, ahorro energético y además de bajo costo, se convierten en una alternativa viable para su aceptación y consumo entre la sociedad.

El desarrollo de la tecnología inalámbrica aplicada a la domótica se engloba dentro de las siguientes tecnologías: Wi-Fi, bluetooth y ZigBee todas ellas bajo protocolo IEEE 802.15.4 para redes inalámbricas [4].

Hoy en día, la transmisión de datos a través de tecnología bluetooth y los dispositivos que la manejan son utilizados generalmente para uso personal, entretenimiento, envío de paquetes de datos en telefonía y pequeñas redes inalámbricas que facilitan la sincronización de datos entre equipos personales. A través del presente trabajo se busca verificar si la tecnología bluetooth tiene capacidad, fiabilidad y flexibilidad para realizar tareas de control en casa-habitación a fin de mejorar el confort, la seguridad y el ahorro de energía requerido en las aplicaciones domóticas.

A diferencia de otros trabajos desarrollados en el área donde se presenta un sistema domótico operado mediante Notebook (Bogado, 2012); ahora, se opta por utilizar un dispositivo móvil en donde para el diseño de la aplicación domótica con

tecnología bluetooth que se presenta se realizó un estudio previo donde se encontraron dos problemas particulares; el poco mercado de clientes producto del alto costo de las aplicaciones domóticas y el medio de transmisión utilizado en la aplicación, donde los sistemas alámbricos a pesar de ser actualmente los mas utilizados, requieren de una mayor instalación, de una modificación del lugar donde se implementará y de un incremento en el costo del producto final.

El resultado que se persigue en este proyecto es el desarrollo de un sistema operado mediante tecnología bluetooth para controlar inalámbricamente el encendido y apagado de lámparas y controlar la intensidad de iluminación generada por las mismas desde un teléfono celular o una tableta electrónica con sistema operativo Android.

Se espera tener un prototipo de menor costo, usando dispositivos y tecnologías de buena calidad pero comercialmente más accesible, como valor agregado, se pretende dirigirlo también para personas con capacidades diferentes o de la tercera edad que podrían mejorar su calidad de vida en su lugar de residencia.

2. Integración de la aplicación.

La domótica se clasifica según su medio de transmisión en los siguientes tres tipos: Cableado dedicado, red eléctrica y transmisión inalámbrica [5]. A pesar de que las aplicaciones con cableado dedicado son las mas utilizadas presentan inconvenientes en viviendas ya construidas que no disponen de una preinstalación de cable para redes domésticas, los cual puede resolverse utilizando medios inalámbricos.

2.1 Comunicación Bluetooth

Tecnología de comunicación inalámbrica presentada por SIG (Special Interest Group) en 1998, diseñada para reemplazar los cables que interconectan dispositivos de bajo consumo, corto alcance y bajo costo. Utiliza radiofrecuencia de 2,4 GHz en la banda ISM (Industrial, Cientific and Medical) [4]. El uso abierto al publico sin necesidad de licencias y requiere de un mecanismo de protección contra interferencias como técnicas de ensanchado de espectro. En la figura 1 se muestra una comparación entre las diferentes tecnologías inalámbricas existentes en el mercado.

ZigBee (WPAN)	Bluetooth (WLAN/WPAN)	Wi-Fi (WLAN)
------------------	--------------------------	-----------------

Estándar 802.15.4 250 Kbps TX: 35 mA Standby: 3 µA 32-60 KB Memory 75 m alcance PMI	Estándar 802.15.1 10 Mbps TX: 40 mA Standby: 200 µA 100+ KB Memory 100 m alcance PMI	Estándar 802.15.11 54 Mbps TX: >400 mA Standby: 20 mA 100+ KB Memory > 100 m alcance PMI
---	--	--

Fig. 1. Comparación entre comunicaciones inalámbricas.

Se puede identificar una clara ventaja de Wi Fi sobre Bluetooth, su velocidad de transmisión, en contraparte Bluetooth tiene un menor consumo, el alcance para Wi Fi es un poco mayor, sin embargo esto se puede compensar con la clase de comunicación Bluetooth que se use. Clase 1, utilizado en aplicaciones industriales. Clase 2, utilizada en dispositivos móviles y Clase 3 generalmente poco utilizada, la figura 2 muestra sus potencias y alcances:

Clase	Pot. Máx.	Pot. Min.	Alcance
1	100 mW.	1 mW.	100 m.
2	2.5 mW.	0.25 mW	10 m.
3	1 mW.	NA	1 m.

Fig. 2. Clases de comunicación Bluetooth.

2.2 Sistema Operativo Android

Es un sistema operativo basado en Linux® para dispositivos móviles como teléfonos, PDA, ordenadores etc. Posee licencia Apache®, que es una licencia libre y de código abierto, este una razón por la cual se utilizará en el diseño de la aplicación domótica con tecnología bluetooth propuesta.

Una característica adicional en el uso de Android® tiene que ver con suficiente presencia en el mercado de los dispositivos móviles y disponibilidad para la mayor parte de la sociedad. Sobre esto, en la figura 4 se puede ver la presencia de Android en el mercado a escala mundial.

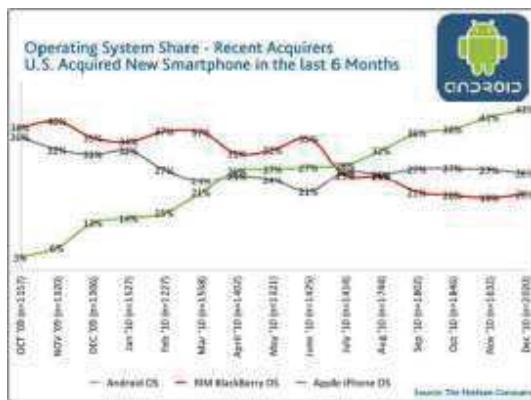


Fig. 3. Inclusión de Android en el mercado.

2.3 Arduino Mega

Arduino® es una plataforma de desarrollo de computación de código abierto, basado en un entorno para la creación de programas, el lenguaje de Arduino es una implementación de *Wiring* (Código abierto para microcontroladores) una plataforma de computación física que a su vez se basa en *Processing*, un entorno de programación multimedia [6]. El Arduino Mega es una placa microcontrolador basada ATmega1280. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), 16 entradas analógicas, un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reinicio.

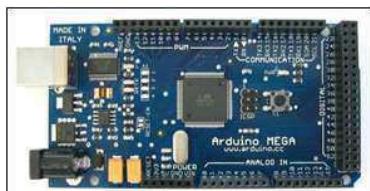


Fig. 4. Tarjeta Arduino Mega.

2.4 App Inventor

App Inventor® es una herramienta útil de programación, que permite el desarrollo de aplicaciones para dispositivos móviles que usen el sistema operativo Android basada en Open Blocks Java library similar a programaciones como Scratch y StarLogo.

El App Inventor está integrado por un editor de bloques que es la parte donde se diseña el programa y también consta un módulo en Web, donde aparte de ser el punto de entrada tenemos acceso a nuestros proyectos y, una vez abierto un proyecto, podemos

entrar a la sección de diseño de nuestra aplicación. Esta sección es donde podemos añadir los componentes y configurarlos apropiadamente. Si se trata de componentes visuales, entonces definimos también el diseño de la interfaz, en el momento del desarrollo se ejecuta la última versión del App Inventor disponible en su sitio web y los proyectos se guardan en línea (figura 5).

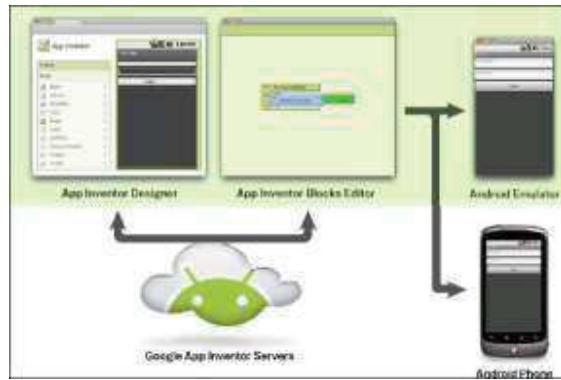


Fig. 5. Desarrollo con App Inventor.

2.5 Módulo HC-06

Este módulo de bluetooth es el dispositivo que utilizamos para la recepción de datos enviados desde una aplicación desarrollada en “App Inventor”. Se eligió este módulo de bluetooth por el bajo costo comparado con los módulos RN-41 y RN-42 los cuales tienen las mismas funciones y características, la única diferencia es el rango de cobertura. Siendo así un módulo que cubre las necesidades para el desarrollo del proyecto.

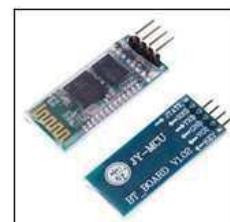


Fig. 6. Módulo HC-06

La configuración del módulo se realiza ajustando los tres aspectos esenciales de nombre, contraseña y velocidad de transmisión mediante comandos AT que se envían mediante una tarjeta Arduino.

2.6 Diseño Hardware

El desarrollo de esta aplicación se basa en el prototipo de una casa habitación, con el objetivo de controlar la intensidad de luz en recámaras a través de un regulador o atenuador, control de apertura de accesos por medio de un acoplamiento mecánico con motor, activación de alarma y simulador de presencia mediante un sensor de proximidad. Se propone utilizar una comunicación inalámbrica de tipo Bluetooth, ya que prácticamente dentro del alcance de la señal no presenta perdidas de datos y en general tiene un comportamiento estable y seguro, además no depende de un servidor o servicio de internet, y si se tiene una comunicación Bluetooth clase 1 se puede compensar una de las desventajas de este tipo de comunicación que es el alcance. Éste tipo de comunicación será el medio por el que se transfieran los datos entre el dispositivo móvil y la tarjeta Arduino la cual mediante el módulo HC-06 recibirá las señales de control para los respectivos actuadores instalados en la casa.

2.7 Diseño Software

Para el desarrollo de esta aplicación se utilizó el software APP Inventor en el cual se fue diseñando cada una de las interfaces que estarían en contacto directo con el usuario, en esta aplicación se diseñan los gráficos, botones, letras, símbolos u otros elementos para el usuario. Los materiales necesarios para el desarrollo de esta aplicación son:

- Computadora PC
- Navegador Google Chrome
- Dispositivo Android (Tablet, Celular, etc.)
- Versión java actualizada.

Con esta herramienta se definen los componentes de la aplicación con sus respectivas propiedades, así como la interfaz que se utilizará para la comunicación desde el dispositivo móvil. Una vez realizado lo anterior, ahora se procede a darle funcionamiento a todos los elementos. El encargado de esto es el editor de bloques, en éste se define el funcionamiento de cada elemento de la aplicación y los datos que se enviarán para el control de la casa habitación, así como la interacción de cada uno de los bloques con sus respectivas características (figura 8).

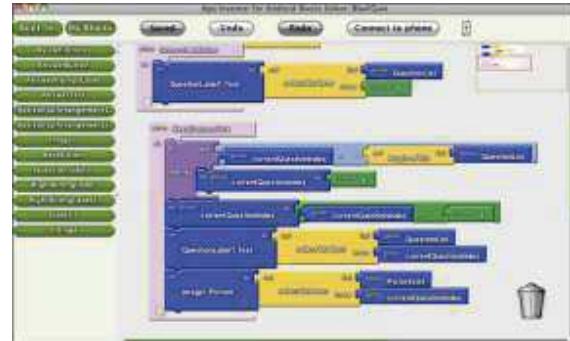


Fig. 7. Editor de bloques de App inventor.

3. Análisis de resultados

En la primera pantalla (figura 8) que se diseñó para el usuario se le pide introducir la clave de control de acceso, al controlarse el acceso a una casa habitación es indispensable la solicitud de una contraseña que si no se introduce correctamente no procede con la aplicación. Si el acceso es correcto la segunda pantalla es un menú creado para que el usuario seleccione que control quiere realizar; accesos, iluminación o alarma.

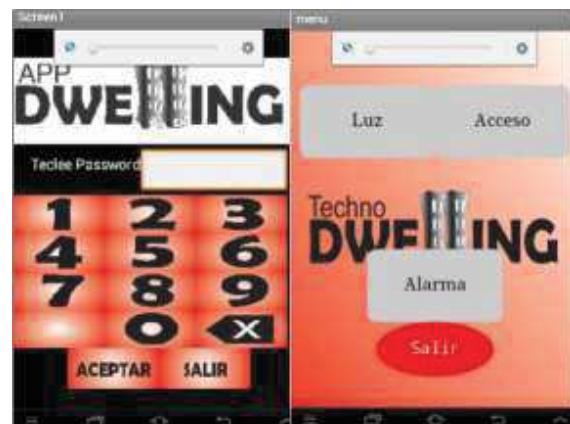


Fig. 8. Acceso y menú principal de la aplicación.

De acuerdo a la función seleccionada dentro del menú principal, se presentan una serie de submenús en los cuales el usuario puede tener un control más definido, por ejemplo en la figura 9 se presentan 5 diferentes intensidades luminosas programadas para 3 lámparas principales dentro de la casa, así como la apertura y cierre de una puerta principal de acceso y un portón de acceso vehicular.



Fig. 9. Menú de iluminación y control de accesos.

En la figura 10 se tiene el control para activar la alarma doméstica con simulador de presencia en 1, 2, 3 y 4 horas.



Fig. 10. Menú de alarmas y prototipo.

Los resultados se percibieron en un prototipo a escala, donde se tuvo una comunicación eficiente, estable, prácticamente insensible a perturbaciones, en contra detectamos una perdida leve en alcance con objetos intermedios, para esta aplicación es importante mantener estabilidad en conexión ya que permitirán garantizar la seguridad del sistema. El modulo HC-06 proporciona una velocidad de transmisión y recepción de datos constante y estable, sin embargo aunque el fabricante especifica que esta comunicación puede atravesar objetos sólidos sin problema, las pruebas realizadas se percibe una pérdida de alcance de la señal (figura 11).

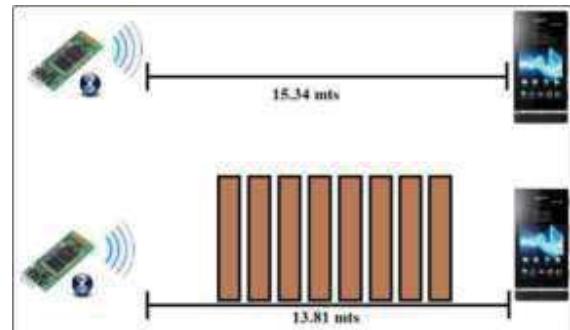


Fig. 11. Alcance de señal libre y con 8 barreras.

Con un entorno libre de obstáculos se puede tener una comunicación estable a una distancia poco mayor a 15 metros a la redonda del receptor. Se hace notar que hay una pérdida con interferencia de 8 barreras que puede ser un número alto para estas aplicaciones, a pesar de esto se sigue considerando viable uso de Bluetooth por las ventajas presentadas en la sección 2.1. Aun así se puede mejorar mucho el rango de alcance si se trabaja con dispositivos de comunicación Bluetooth de clase 1.

Los resultados en la realidad se muestran en las figuras 12 y 13; donde logró controlar el encendido e intensidad de las lámparas, la activación de alarma y el control sobre los accesos de la casa.



Fig. 12. Control de iluminación en casa.

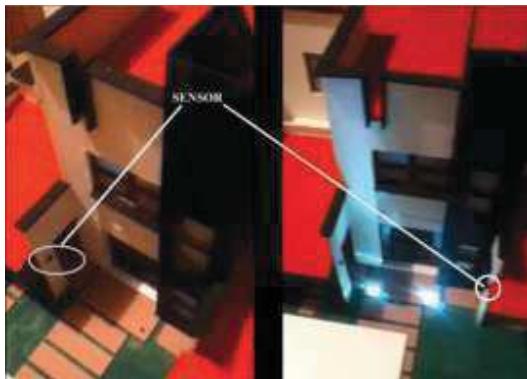


Fig. 13. Control de accesos y alarma.

APP inventor también posee un emulador que realiza una conexión con un teléfono virtual para simular las funciones es que se programaron en la aplicación, esta función es importante porque permite realizar una primera prueba de funcionamiento del sistema creado. Una vez descargada en el dispositivo, se instala como cualquier aplicación para Android.



Fig. 14. Simulador de App inventor.



Fig. 15. Simulación de aplicación del proyecto.

Dentro del desarrollo de la aplicación parte de los resultados se pudieron prever en las simulaciones hechas por el emulador de APP inventor y por Isis de Proteus®. El uso de estas herramientas para la programación de los dispositivos es indispensable para identificar posibles fallas o errores de funcionamiento o comunicación. En la figura 16 se muestra la simulación del sistema y el diagrama de conexión.

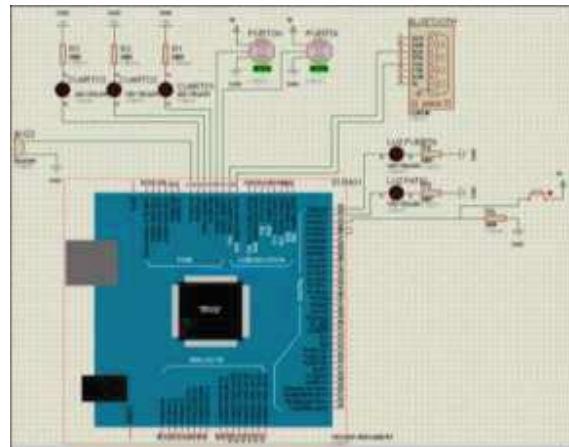


Fig. 16. Simulación en Isis de Proteus.

4. Resultados

Se obtuvo una aplicación de costo accesible, sobretodo en la parte de control e interfaz con el usuario, se utilizaron distintos elementos que están al alcance del público de manera general, como Comunicación y Software libre, un dispositivo móvil que ya es muy común en la mayor parte de la sociedad y controladores confiables y de fácil manejo a un costo relativamente bajo. Se obtuvo una comunicación estable a un alcance determinado, el comportamiento real del sistema estuvo de acuerdo a las simulaciones previas.

5. Conclusiones

Los avances en la tecnología electrónica, de control y comunicaciones, nos van proporcionando cada vez más elementos disponibles y accesibles para desarrollar aplicaciones que puedan beneficiar a más partes de la sociedad y generar conocimiento para estudiantes y docentes. La integración de distintos elementos en hardware y software de la aplicación mostrada se puede extender a más elementos para generar un sistema de más capacidad que proporciona las distintas ventajas que se pretenden con la domótica.

En los resultados obtenidos se demostró que para cierta clase de aplicaciones domóticas de bajos recursos se tiene un sistema suficientemente estable y confiable. Se describió una aplicación que puede potencializarse pero representa una alternativa económica. Se puede mejorar significativamente al tener elementos de mayor calidad y prestaciones, como en éste caso el alcance de la señal de

comunicación y la seguridad en el sistema de control que son las áreas de oportunidad a mejorar en trabajos futuros en este mismo tema.

Referencias

- [1] Huidobro J. M., Millán R. J. “*Domótica: edificios inteligentes*”, Creaciones COPIRIGHT España, 2004.
- [2] Junestrand S., Passaret X. Vázquez D. “*Domótica y Hogar Digital*”, Thomson Paraninfo, España, 2005.
- [3] Andaluz Ortiz, Víctor, Yépez Rodríguez, Juan. “*Diseño y construcción de un control domótico utilizando bluetooth por medio de un pda*”, Tesis, Escuela Politécnica Nacional, Quito, Ecuador. 2008.
- [4]. S.L.Jin, W.S. Yu, S.C. Chung, A. “*Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and Wi-Fi, Industrial Electronics Society*”. IECON 2007. 33rd Annual Conference of the IEEE. 2007.
- [5] Huidobro J. M., Millán R. J. “*Manual de Domótica*”, Creaciones COPIRIGHT España, 2010.
- [6] Brian W. Evans. “*Arduino Programming Handbook: a Beginner’s Reference*”, Editorial, USA, 2 edición, 2008.

Concentrador Fresnel Utilizando un Controlador Lógico Programable

Terán Valenzuela María Guadalupe, Vega Castro Jorge Luis, Lugo Coporo Marcos Raúl y Ozuna Huerta Gustavo.

Universidad de Sonora

Resumen

En el siguiente artículo se mostrara el desarrollo necesario para habilitar un concentrador solar fresnel, utilizando un controlador lógico programable (PLC), el diseño de los sensores, y de la etapa de potencia para lograr el seguimiento de la trayectoria aparente del sol.

1. Introducción

La programación de un controlador lógico programable permite dar la robustez necesaria al sistema para que efectúe su objetivo correctamente [1], este puede operar en múltiples condiciones el trabajo al aire libre, las altas temperaturas y los climas extremos de la región (Noroeste de México).

2. Problema

El siguiente problema es lograr el seguimiento de la trayectoria aparente del sol, moviendo un sistema de espejos de este a oeste controlando el giro de un motor de corriente directa conectado a una transmisión lineal como se muestra en la siguiente figura 1.

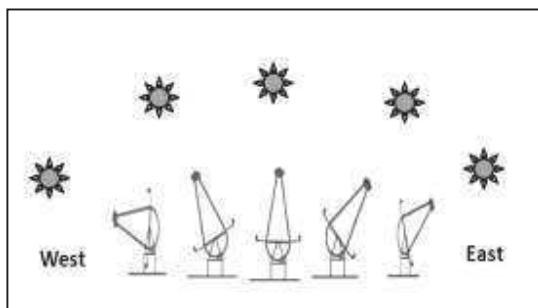


Fig. 1. Seguimiento solar.

3. Prototipo



Fig. 2. Concentrador Fresnel prototipo.

4. Control

El sistema toma la información de dos sensores foto resistivos que indicaran al PLC la posición de la estructura móvil, dos salidas las cuales proporcionarán la señal a la etapa de potencia para poder hacer efectivo el movimiento del motor y repositionar los rayos solares. La programación cuenta además con un temporizador activado por dos limit switch para que al llegar al final de carrera se demore 5 segundos en regresar al inicio, el cual estará indicado por otro limit switch, en otras palabras servirá como un botón de "reset" para empezar el ciclo nuevamente al día siguiente, la estructura de la programación se puede observar en la figura 3.

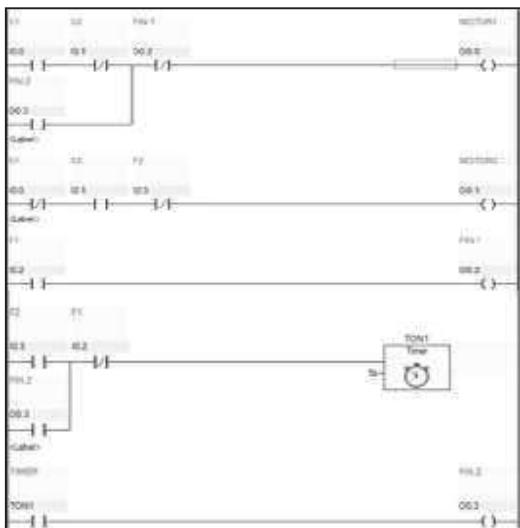


Fig. 3. Diagrama escalera.

5. Sensores

Los sensores propuestos se fabricaron con fotorresistencias y un circuito comparador el cual indica cual tiene más luz en un tiempo y posiciones determinadas. El funcionamiento del circuito es tomar el valor de las fotorresistencias y compararlo con una resistencia variable, en este caso un potenciómetro, para ajustar el nivel de luz que recibirán antes de mandar la señal resultante al PLC; el circuito consta de dos amplificadores operacionales en un arreglo como comparador analógico [2] como se muestra en la figura 4.

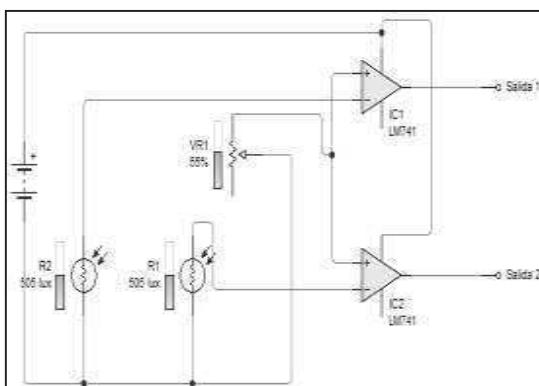


Fig. 4. Comparador analógico.

6. Transductor de potencia.

En esta etapa se analizará la forma de tomar la señal enviada del circuito comparador y tratarla para que se ajuste y así pueda ser leída por el PLC (24 v)

[4]. Esta señal actuará como entrada 1 y entrada 2, señal de la fotorresistencia 1 y señal de la fotorresistencia 2, respectivamente. La etapa de potencia cuenta con un optoacoplador que servirá de protección para el comparador y un transistor que comutará al relevador, por el relevador pasaran 24 v que llegarán al PLC en forma de señal, el diagrama se muestra en la figura 5.

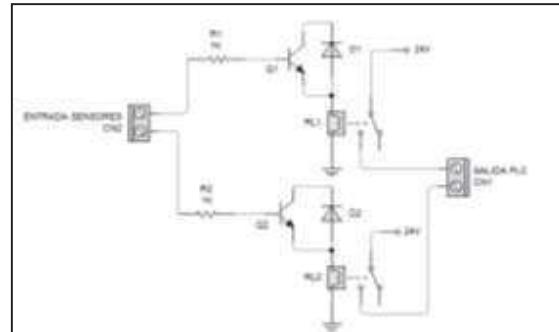


Fig. 5. Transducción de potencia.

7. Puente H.

Para que la estructura móvil realice el movimiento requerido para seguir la posición del sol durante el día se requiere de un motor que haga un movimiento angular, además necesita poder invertir su polaridad para girar en sus dos sentidos ya que al final del día o final de carrera el concentrador requerirá regresar a la posición inicial para comenzar el ciclo nuevamente, para hacer esto posible se utilizará un circuito conocido como "El puente H" el cual es un sistema de relevadores que hace posible cambiar la dirección de trabajo de un motor invirtiendo las polaridades de la corriente según se le indique mediante botones o señales de sensores, el puente H propuesto se ilustra en la figura 6.

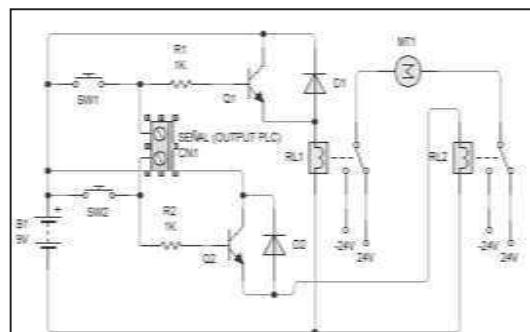


Fig. 6. Puente H

8. Estructura del sensor.

La estructura física de los sensores [3] consiste en una placa donde se colocarán las LDR (resistencia dependiente de la luz) y dividirlas por una pared sólida [5] para que se genere una gama de sombras que van desde lo más tenue a lo más intenso, la sombra generada en cierto punto permitirá generar una variación resistiva y obtener la diferencia analógica para efectuar el seguimiento del sol, los sensores deberán tener una calibración específica según sea la posición donde sean colocados, entre más se aleje de la pared la sombra será más tenue, la figura 7 muestra lo anterior.

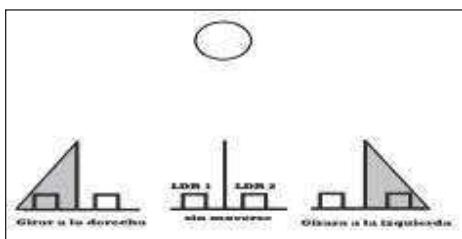


Fig. 7. Estructura del sensor.

9. Actuador

El actuador que se usó para mover el prototipo es un actuador de tipo lineal adecuado mecánicamente para que permita generar un movimiento angular cuyo objetivo es posicionar la estructura móvil del concentrador solar [3] donde se obtenga la mayor incidencia de reflexión, este actuador es accionado con un voltaje de 24 volts, tiene la capacidad de mover 50 kilogramos aproximadamente, la fuerza del motor está sobrada debido a que se utilizó una estructura fabricada principalmente por aluminio, el recorrido lineal total del actuador es de 12 pulgadas pudiendo así generar un movimiento angular de 160 grados, el motor se ilustra en la siguiente figura 8.



Fig. 8. Actuador lineal.

10. Análisis de resultados

La siguiente etapa del proyecto muestra los resultados obtenidos al hacer una prueba piloto del concentrador en la ciudad de Hermosillo Sonora, el día tres de agosto del presente año, en este se realizaron mediciones con un termómetro laser en grados Celsius apuntándolo al target del sistema de espejos, que está constituido por un tubo de 2/4 de pulgada de cobre por el cual pasara el líquido a calentar, las mediciones se efectuaron en intervalos de media hora empezando a las 7 am y concluyendo las mediciones a las 5:30 pm, las temperaturas fueron afectadas de forma indirecta por factores climáticos como el viento, la humedad, las nubes, etc., sin embargo en la ciudad de Hermosillo se tiene un clima consistente y con alta incidencia de rayos solares durante el día, lo cual permitió obtener resultados satisfactorios en las primeras pruebas, dichos resultados de un día de seguimiento solar se muestran en la figura 9.

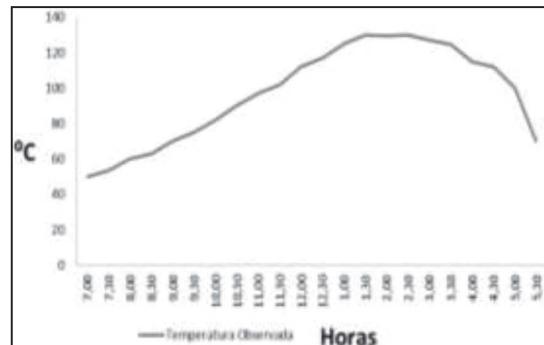


Fig. 9. Gráfico de temperaturas.

11. Conclusión

En lo que concierne al equipo de trabajo el objetivo se cumplió satisfactoriamente, el prototipo sigue la trayectoria aparente del sol, se realizaron pruebas para medir la eficiencia del sistema y obtener valores de las temperaturas alcanzadas a lo largo del día, se han obtenido temperaturas de alrededor de 130 grados Celsius, esto da pie al siguiente grupo de investigación el cual aprovechará estas temperaturas para hacer el diseño termodinámico del sistema y utilizarse en la desalinización de aguas marinas; para potabilizarla, ya que es un recurso necesario y escaso en la región de Sonora, el equipo de trabajo espera que la publicación de este documento, sirva a otros investigadores que estén trabajando o quieran incursionar en esta área.

Referencias

- [1] Pérez J. Pineda M. “*Automatización de maniobras industriales*”, Alfaomega Grupo Editor, S.A. de C.V., México, Primera edición, 2008.
- [2] Alciatore D., Histon M. “*Introducción a la mecatrónica y los sistemas de medición*”, McGraw-Hill Interamericana, México D.F., Tercera edición, 2008.
- [3] Kumar S. “*Introducción a la robótica*”, McGraw-Hill, India, Primera edición, 2008.
- [4] Maloney, Timothy J. “*Electrónica industrial moderna*”, PEARSON EDUCACIÓN, México, 5^a. Edición, 2006.
- [5] Farhan A. “*Mechatronics Design of solar tracking System*”, *International journal of current Engineering and Technology*. Impressco, Volumen 3, Número 2, 13 páginas, 2013.

Diseño del Sistema de Control para un Mezclador Volumétrico de Concreto Utilizando un Micro-controlador

Lugo Coporo Marcos Raúl, Vega Castro Jorge Luis, Gómez Palomares Jesús Leonel
y Ozuna Huerta Gustavo

Universidad de Sonora

Resumen

El siguiente artículo muestra los pasos para la elaboración de un sistema de control para un mezclador volumétrico de concreto móvil, en el cual se puede controlar la resistencia del material, con la modificación de las cantidades de cada agregado tales como el agua, la grava, y la arena utilizando un microcontrolador.

Concreto, Microcontrolador, Resistencia

1. Introducción

El siguiente documento muestra cómo se instaló un micro controlador para sustituir un sistema de control basado en un Controlador Lógico Programable PLC [1] el sistema que se automatizó ya contaba con un PLC pero este se dañó, al instalar un micro controlador [2] se abrieron más posibilidades para el control de este, así como un menú en pantalla y aceptar la modificación de las propiedades del concreto en este caso la resistencia de este. La maquinaria está montada sobre un tractor camión como se muestra en la figura 1.



Figura 1. Dosificador Móvil de Concreto

Este nuevo sistema de control se instaló y se capacito al operador para poder modificar las fórmulas de resistencia de este, mediante un teclado y una pantalla LCD, en donde el operador mediante un menú muy sencillo puede cambiar la configuración de la mezcla del agua, arena, grava y aditivos para la producción de concreto en sitio cambiando la resistencia del concreto

2. Prototipo

Para el diseño del sistema de control se tomó en cuenta las modificaciones y mejoras con respecto al sistema de control anterior, en el cual no se podían alterar las propiedades del concreto automáticamente ya que el producto solo se dosificaba, en el nuevo sistema de control se pueden modificar la resistencia del concreto cambiando solo los valores de la siguiente tabla 1 [3].

Tabla 1. Proporciones para resistencia de concreto

psi	Resistencia		Cantidades para 1 m ³		Proporciones	Litros máximos por cada saco de cemento (agua)
	MPa	Kgf/cm ²	Sacos de cemento (m ³)	Arena m ³		
1500	10.3	105	5.5 (0.22)	0.57	0.772	12.6/3.51
2000	13.8	141	6.1 (0.244)	0.549	0.786	12.25/3.22
2500	17.2	178	7.1 (0.284)	0.523	0.77	11.84/2.71
3000	20.7	211	7.6 (0.304)	0.523	0.77	11.72/2.54
3500	24.1	246	8.3 (0.332)	0.488	0.774	11.47/2.35
4000	27.6	281	9.1 (0.364)	0.47	0.772	11.29/2.12
4500	31.0	316	9.6 (0.384)	0.453	0.776	11.18/2.0

Donde se modifica solamente el valor de Kgf/cm² y el sistema abrirá o cerrará las compuertas para dosificar las cantidades de cemento, arena, grava y litros de agua, los valores de la tabla se obtuvieron de la siguiente formula (1), (2), (3) y (4).

$$\text{Cemento} = 0.0008 \text{Resistencia} + 0.364 \quad (1)$$

$$\begin{aligned} \text{Arena} &= -0.0006 \text{Resistencia} + 0.6297 & (2) \\ \text{Grava} &= 1 \times 10^{-5} \text{Resistencia} + 0.7782 & (3) \\ \text{Aqua} &= 0.5062 \text{Resistencia} + 86.85 & (4) \end{aligned}$$

3. Control

Para controlar el sistema se decidió utilizar un microcontrolador 16F887 que cuenta con 35 entradas o salidas dependiendo como se configuren como: 14 entradas analógicas, 2 puertos PWM, puerto serial, 3 timers programables, puerto I²C, memoria EEPROM, modo de espera y trabaja hasta los 20 MHz [4]. Donde se conectaron las entradas y salidas con el siguiente orden como se observa en la tabla 2 y tabla 3.

Tabla 2

Entrada	Descripción
Teclado	4X4
Sensor Inductivo	Cuenta Revoluciones del Motor
Botón Encendido	Enciende el Sistema
Botón de Paro	Paro de Emergencia
Encoder G	Muestra la Posición de Compuerta de Grava
Encoder A	Muestra la Posición de Compuerta de Arena

Tabla 3

Salidas	Descripción
LCD	Pantalla 16x2 Caracteres
Motor G	Controla Compuerta Grava
Motor A	Controla Compuerta Arena
Válvula Agua	Controla el Flujo de Agua

La estructura del sistema se puede observar en la figura 2, donde se muestran la disposición de las entradas y salidas con respecto al micro controlador.

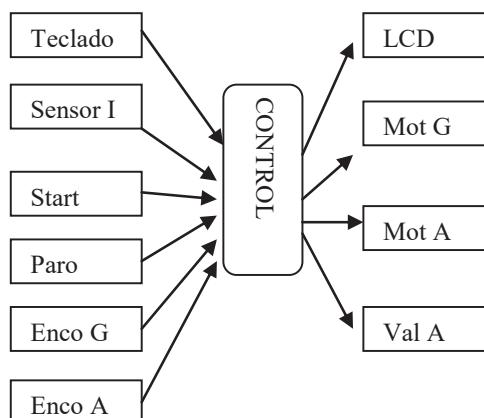


Figura 2. Sistema de Control

4. Diseño

El diseño del circuito se realizó en PCB Wizard donde se tomaron en cuenta instalar entradas y salidas fáciles de conectar como tiras de pines y de terminal blocks, así como la utilización de opto acopladores para proteger el circuito de control debido a que los sensores operan con 12V y a diferencia del micro controlador que opera con 5V [5] y amplificadores operacional para acondicionar las señales [6], el circuito se muestra en la figura 3.

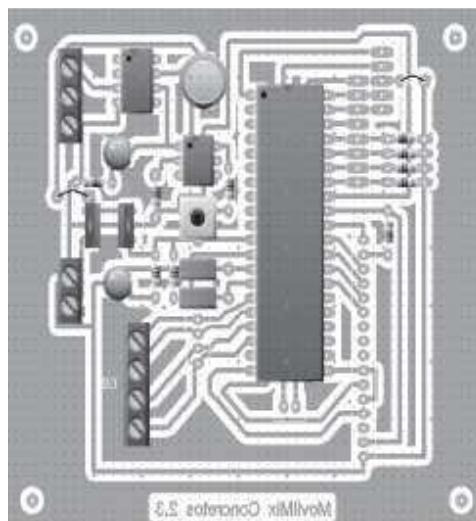


Figura 3. Circuito de Control

El circuito ya elaborado e instalado se puede observar en la siguiente figura 4 y 5, donde se muestra como se instaló y se protegió debido a que la maquinaria se limpia a diario con agua a presión y debemos evitar un corto circuito o sulfatación de terminales.



Figura 4. Vista Interior del Circuito Instalado

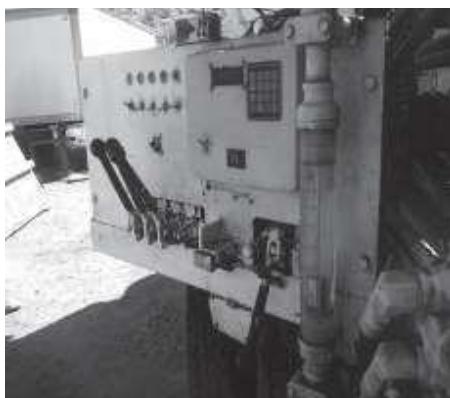


Figura 4. Vista Exterior del Circuito Instalado

5. Comparación de resultados

Según el cliente, cada metro cúbico de concreto ocupa 216 litros de agua según la resistencia del concreto. [3]. Sabemos que para obtener un metro cúbico de concreto se necesitan en total 1125 conteos por lo tanto tenemos que mandar 0.192 litros por conteo.

Se realizaron 5 pruebas y se registraron los conteos, los kilogramos de cemento dosificado así como el tiempo requerido obteniendo en promedio 150 conteos y 57.552 kilogramos de cemento por prueba, por lo tanto se requieren 2.6 conteos por kilogramo de cemento. Los datos se muestran en la tabla 3 que se muestra a continuación.

Tabla 3. Resultados de muestreo

PRUEBA No.	1	2	3	4	5
Conteos	150	150	151	149	150
kg	56.9	57.36	57.8	57.68	58.02
TIEMPO	28.47	28.88	28.76	28.45	28.44
Promedio conteos					150.00
Promedio Kg de cemento					57.552

A continuación se muestran los resultados obtenidos después de realizar 3 pruebas abriendo distintas compuertas a las que llamaremos A, B y C, con 3 repeticiones cada una, del agregado grueso, es decir, la grava y otras 3 pruebas con el agregado fino, arena con sus 3 respectivas repeticiones. Se registró el número de conteos y el peso neto en kilogramos de grava y arena respectivamente. Los datos obtenidos así como los promedios se encuentran en las tablas 4, 5, 6, 7, 8 y 9.

Tabla 4. Muestra de grava en compuerta 1

PRUEBA No.	A		
	1	2	3
Compuerta	1	1	1
Conteos	151	152	150
Peso Neto (Kg.)	78.2	73.1	69.85
Promedio conteos			151
Promedio Peso			73.716
Peso por conteo según apertura compuerta			0.488

Tabla 5. Muestra de grava en compuerta 2

PRUEBA No.	B		
	1	2	3
Compuerta	2	2	2
Conteos	110	110	111
Peso Neto (Kg.)	86.5	90.6	87.85
Promedio conteos			110.333
Promedio Peso			88.316
Peso por conteo según apertura compuerta			0.800

Tabla 6. Muestra de grava en compuerta 3

PRUEBA No.	C		
	1	2	3
Compuerta	3	3	3
Conteos	81	82	79
Peso Neto (Kg.)	97.3	98.05	93.2
Promedio conteos			80.666
Promedio Peso			96.183
Peso por conteo según apertura compuerta			1.192

En la compuerta 1 obtuvimos un promedio de 0.488 kg por conteo, 0.800 kg en la compuerta 2 y 1.192 kg en la compuerta 3. Dichos resultados pueden verse en la figura 5. Si el cliente quiere controlar el peso de grava por conteo solo debe sustituir el valor de X con el número de compuerta abierta en la ecuación $Y = 0.3521X + 0.1228$.

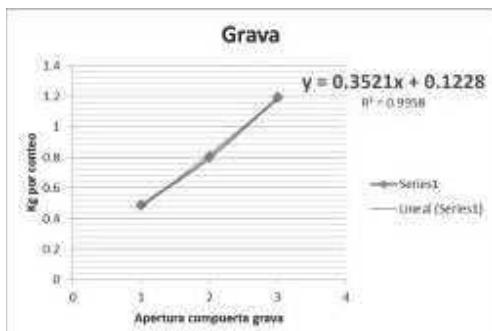


Figura 5. Gráfica de kg de grava por apertura de compuerta

Las siguientes tablas muestran los datos obtenidos a partir de las muestras para la arena. Los pesos por conteo obtenidos son: para la compuerta 1 se tienen 0.683 kg, para la compuerta 2 se tiene un peso promedio de 1.282 kg y para la compuerta 3 el promedio es de 1.791 kg.

Con estos datos se obtuvo la siguiente ecuación $Y = 0.5538X + 0.1449$, con la cual podemos obtener los kg de arena por conteo. En la figura 6 se pueden observar dichos resultados en una gráfica que compara los kg por conteo contra la abertura.

Tabla 7. Muestra de arena en compuerta 1

A			
PRUEBA No.	1	2	3
Compuerta	1	1	1
Conteos	151	149	151
Peso Neto (Kg.)	99.55	104.4	104.45
Promedio conteos			150.333
Promedio Peso			102.8
Peso por conteo según apertura compuerta			0.683

Tabla 8. Muestra de arena en compuerta 2

B			
PRUEBA No.	1	2	3
Compuerta	2	2	2
Conteos	80	80	80
Peso Neto (Kg.)	101.35	103.75	102.6
Promedio conteos			80
Promedio Peso			102.566
Peso por conteo según apertura compuerta			1.282

Tabla 9. Muestra de arena en compuerta 3

PRUEBA No.	C		
	1	2	3
Compuerta	3	3	3
Conteos	50	50	50
Peso Neto (Kg.)	88.4	89.95	90.35
Promedio conteos			50
Promedio Peso			89.566
Peso por conteo según apertura compuerta			1.791

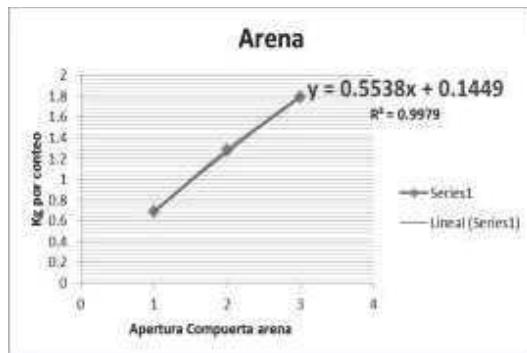


Figura 6. Gráfica kg de arena por apertura de compuerta

También se realizaron pruebas de resistencia al concreto generado en el mezclador volumétrico para comparar su resistencia con la resistencia esperada o teórica [3], realizamos 7 muestras destructivas con diferentes proporciones de mezcla de cemento, grava y arena en el laboratorio de resistencia de materiales de la Universidad de Sonora, en donde fueron realizadas pruebas de compresión mecánica. Los datos obtenidos se muestran en la tabla 10. Como se puede apreciar en la figura 7, en la cual graficamos los resultados de las pruebas de resistencia contra los valores esperados de dichas proporciones en Kgf/cm^2 , de los cuales encontramos que existe un error del 5.5%.

Tabla 10. Resultados de muestreo

Proporción	Resistencia	
	Teórica	Muestra
1:2.6:3.51	105	114.3
1:2.25:3.22	141	152.9
1:1.84:2.71	176	165.5
1:1.72:2.54	211	203.1
1:1.47:2.33	246	260.3
1:1.29:2.12	281	276.8
1:1.18:2.02	316	329.6



Figura 7. Gráfica de resistencias

6. Conclusiones y recomendaciones

Después de haber realizado el circuito e instalarlo se recomienda hacer las pruebas de filtración de agua y humedad antes de poner a trabajar el sistema, para evitar contratiempos con los que el equipo de trabajo se enfrentó y no se vuelva repetir para algún proyecto futuro de esta naturaleza, el sistema de control esta funcionado de manera satisfactoria y se ampliaron las posibilidades con respecto al sistema que ya existía, el tomar la decisión de general nuevos dispositivos y

dejar de depender de un proveedor extranjero da pie a la creación de tecnología Mexicana.

Referencias

- [1] Pérez J. Pineda M. “Automatización de Maniobras Industriales” Alfa Omega Grupo Editor S.A de C.V., México Primera Edición, 2008.
- [2] Valdés P. y Pallás R. “Microcontroladores Fundamentos y Aplicaciones con PIC” Alfa Omega Grupo Editor S.A de C.V., México, Primera Edición, 2007.
- [3] Sánchez D. “Tecnología del Concreto y del Mortero” Bhandar Editores, Colombia, Quinta Edición, 2001.
- [4] Angulo J., Romero S. y Angulo I. “Microcontroladores PIC Diseño Práctico de Aplicaciones Segunda Parte” McGraw-Hill Interamericana de España S.A.U., España, Segunda Edición, 2006.
- [5] Malvino A. “Principios de Electrónica” McGraw-Hill Interamericana de España S.A.U., España, Sexta Edición, 2000.
- [6] Franco S. “Diseño con amplificadores operacionales y circuitos integrados analógicos” McGraw-Hill Interamericana Editores S.A de C.V., México, Tercera Edición, 2005

Concentrador Parabólico Controlado con Lógica Difusa.

Ozuna Huerta Gustavo¹, Wozniak Marek², Vega Castro Jorge Luis¹ y
 Lugo Coporo Marcos Raúl¹

Universidad de Sonora¹, Mexico
 Politechnika Lodzka², Poland

Resumen

Este artículo propone y evalúa un sistema de control de seguimiento solar de una parábola controlada por lógica difusa para posicionarla correctamente y obtener la mayor incidencia de rayos solares en un punto específico, controlando el movimiento por medio de dos motores de corriente directa.

1. Introducción

El siguiente artículo muestra cómo hacer un sistema de seguimiento solar con dos grados de libertad para posicionar una parábola con respecto a la trayectoria aparente del sol dándole un movimiento azimutal y de elevación (este-oeste, norte-sur), utilizando un sistema de control difuso, como se muestra en la figura 1.

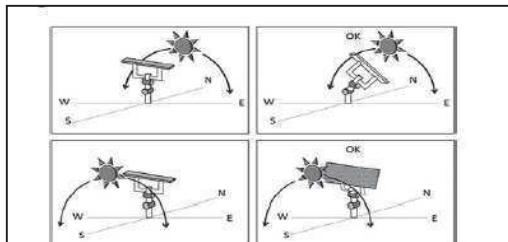


Fig. 1. Posicionamiento del sol

2. Problema

En la actualidad se han desarrollado numerosas fuentes de transformación de energía limpia y los medios para manipularlas tal es el caso de la energía generada por la incidencia de rayos solares en sistemas auto sostenibles; sin embargo, estos pueden resultar muy costos además de solo tener los índices más altos de eficiencia durante las horas pico. La finalidad o propósito es obtener una alternativa económica para seguir la trayectoria aparente del sol y obtener así la mayor incidencia de rayos solares durante el día.

3. Prototipo

Se propone utilizar un sistema parabólico de dos grados de libertad, como se muestra en la figura 2.



Fig. 2. Concentrador Parabólico prototipo.

El cual está compuesto por dos motores de corriente directa y un sistema de sensores ubicado en el foco de dicha parábola, los motores dirigirán el sistema de la siguiente forma:

Motor 1: Posición horizontal (norte-sur, sur-norte).

Motor 2: Posición vertical (este-oeste, oeste-este).

El modelo matemático está basado en las siguientes ecuaciones:

$$\frac{di}{dt} = \frac{V_{app}}{L} - \frac{R}{L}i - \frac{K\phi}{L} \quad (1)$$

$$\frac{dw}{dt} = \frac{K\phi}{J} - \frac{b}{J}w \quad (2)$$

Donde V_{app} es el voltaje de aplicación, R es la resistencia eléctrica, L es la inductancia, b es la fricción, J es el momento de inercia para el motor y este cuenta con un campo electromagnético [1], representado en la figura 3.

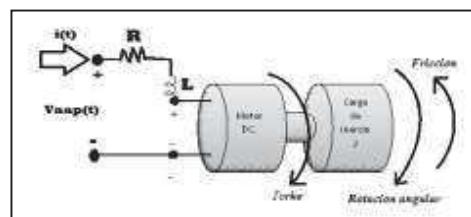


Fig. 3. Motor DC

4. Control

Para la etapa de control se propone el uso de lógica difusa debido a que este emula el razonamiento humano y puede tomar decisiones basadas en datos inexactos [2]. Esto es debido a que la luz del sol por la mañana es diferente que la de medio día, y los sensores no tendrían un valor lógico exacto. Se usó un microcontrolador en el cual las reglas difusas están programadas y posicionan el concentrador parabólico en los cuatro puntos cardinales.

Como sensores se utilizaron cuatro Resistencias Dependientes de la Luz LDR [3] como entradas del sistema, representadas con la letra "S" en el orden de "S1 y S2" (este oeste), "S3 y S4" (norte-sur), además un quinto sensor el cual es un fototransistor con protección Ultravioleta "S5" que detecta únicamente radiación infrarroja emitida por el sol [4], y controla la señal de los otros cuatro sensores.

La señal análoga de los sensores S1-S2 y S3-S4 se envía al microcontrolador con entradas análogas [5] para que este decida cuál es la estrategia adecuada a seguir, y controlar la dirección que tomarán el motor 1 y 2, la cual está constantemente retroalimentada, comparando el valor análogo de entrada del sensor para el posicionamiento preciso de la parábola, si la diferencia de estos está en el área difusa de OK [2]. En la figura 4 se puede apreciar la estructura física de los sensores.

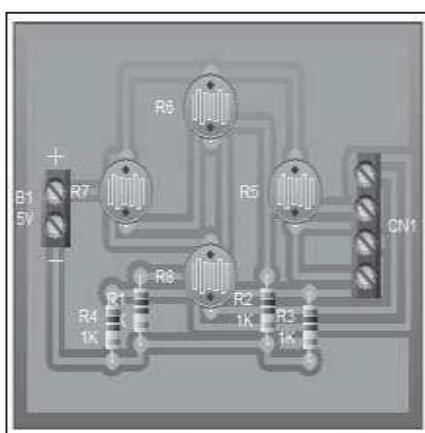


Fig. 4. Estructura física de los sensores.

En la figura 5 y 6 se puede observar la representación de los valores análogos de S1 y S2 que van desde 0 a 1024 estos corresponden a los valores intermedios de [0, 1] como se haría utilizando valores discretos, y muestra el área difusa de "Ok" donde la parábola estaría direccionada al sol Este-Oeste ó Norte-Sur. La información para los Valores verticales o de elevación Este- Oeste es similar a la de los valores horizontales o de azimut Norte-Sur debido a que los valores análogos de entrada serán los mismos.

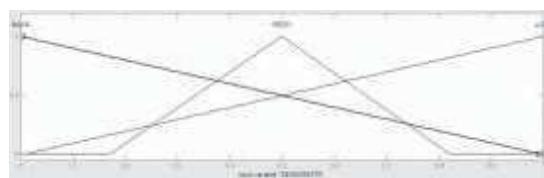


Fig. 5. Conjunto difuso de entrada Sensor-Oeste.

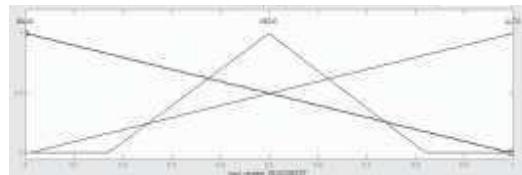


Fig. 6. Conjunto difuso de entrada Sensor-Este.

El conjunto difuso de salida mostrado en la figura 7 es similar para los dos motores debido a que ambos comparten las mismas propiedades eléctricas y mecánicas.



Fig. 7. Conjunto difuso de salida para el controlador difuso del motor EO.

El conjunto difuso de entrada del sensor S5 es el de detectar la radiación solar, este estará localizado fuera del sistema y enviará la señal cuando sea de día, para encender el sistema de control, esto quiere decir que los demás sensores comenzarán a captar el máximo de luz solar cuando el sensor S5 esté activado y dejarán de realizar su trabajo si es de noche [4], esto para evitar movimientos innecesarios

Si alguna luz incidiera en ellos como otras fuentes de luz estas no se tomarán en cuenta debido a que el sensor solamente es sensible a la luz solar. El conjunto difuso de salida de S5 se muestra en la figura 8.

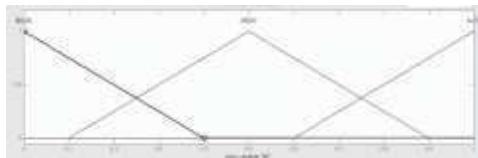
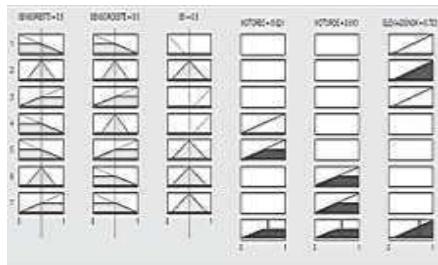


Fig. 8. Conjunto difuso de salida del sensor 5.

Las reglas de control difuso para el sistema en posición vertical o de elevación se muestran en la tabla 1.

Tabla 1. Reglas de control difuso para posición horizontal.



5. Simulación

El sistema se simulo en MatLab utilizando la librería de control difuso, en el cual se obtuvieron los siguientes resultados, por ejemplo: la interacción entre los sensores este y oeste con la salida del motor EO (este-oeste) donde se observa la respuesta del motor con respecto a la señal de los sensores como se muestra en la figura 9.

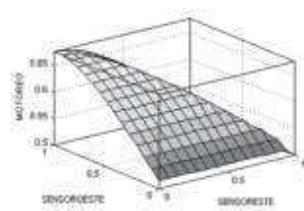


Fig. 9. Gráfico de salida de los sensores Este y Oeste con respecto a la salida Motor EO

Así como la interacción entre los sensores y la salida del sistema OK como se muestra en la figura 10, donde se observa como es la respuesta de la salida

elevación OK (los motores no se mueven) con respecto a los sensores del Este y Oeste.

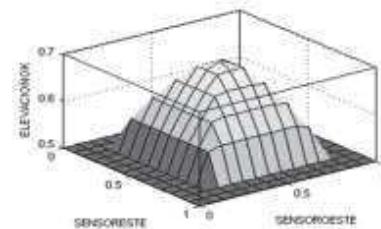


Fig. 10. Gráfico de salida de los sensores Este y Oeste con respecto a la salida Elevación OK

6. Conclusiones y Recomendaciones

Al simular el sistema en Matlab facilita predecir las respuestas del sistema, pero no es hasta que se realice el circuito y se descargue el sistema de control difuso a un microcontrolador es cuando podrá observarse la respuesta real de este, en estos momentos el equipo de investigación busca instalar el sistema en un concentrador parabólico real y tomar las muestras del calor obtenido de este, se espera en un trabajo futuro presentar estos resultados y discutirse con otros investigadores del área.

Referencias

- [1] W. Elshamy, DC motor model, MATLAB central archives, 2006
- [2] Del Brío M., Molina S. “Redes neuronales y Sistemas Borrosos”, Alfaomega, México, 3ra edición, 2007.
- [3] Bolton W. “Mecatrónica. Sistemas de control electrónico en la ingeniería mecánica y eléctrica”, Alfaomega, México, 4ta edición, 2010.
- [4] Kumar S. “Introducción a la robótica”, McGraw-Hill, India, Primera edición, 2008.
- [5] Angulo J., Romero S., Angulo I. “Microcontroladores PIC: Diseño práctico de aplicaciones 2da parte”, McGraw-Hill, España, 2da edición, 2006.

Monitoreo en Tiempo Real de las Variables Físicas de un Panel Solar a Través de LabView

Moreno Castro Luis Fernando, Cruz López Miguel Alberto, Ramiro Ramiro José, Guerrero Rodríguez Erick Damián y Martínez Jiménez José Manuel

Instituto Tecnológico Superior De Zacapoaxtla
División de Ingeniería Mecatrónica
Carretera Acuaco-Zacapoaxtla, Km 8, Col. Totoltepec, Zacapoaxtla, Pue.
Tel: (233) 31 75000.

chico29_91@hotmail.com, migue_cruz15@hotmail.com, semirajo@msn.com, edgr.dam.11@gmail.com

Resumen

Se realiza la adquisición de datos de una celda solar, ubicada en la parte superior del edificio “Unidad de Prácticas” del Instituto Tecnológico Superior de Zacapoaxtla, mediante la captura de datos que se hace a través de una tarjeta de desarrollo Arduino y del software LabView, quienes son los encargados de capturar los datos que emite el panel solar y guardarlos en el ordenador (implementación de un Sistema SCADA). La captura de datos por LabView genera dos archivos uno en formato “.txt” y otro en formato “.xls”; de este modo se pueden analizar los datos con otras herramientas. Por lo tanto se puede analizar el comportamiento de la celda solar de distinta manera, además se puede ver el análisis de esta en tiempo real de acuerdo con el clima de la región, con lo cual se visualiza cómo se comporta la generación de energía con respecto a los cambios de clima a lo largo del día.

Palabras clave: Panel Solar, Análisis Grafico, Adquisición de Datos, Sistema SCADA.

1. Introducción

Las energías alternativas o renovables son aquellas cuyas fuentes de creación se renuevan constantemente, entre estas están la hidráulica, eólica y la solar, estas nos permiten tener un mejoramiento de los problemas que se tienen en el medio ambiente.

El sol es un elemento que se encuentra en todo nuestro planeta, este elemento además de ser un

elemento importante para la vida es usado como generador de energía ya que este puede producir en un año el equivalente a 60 toneladas de petróleo según la asociación nacional de energía solar (ANES).

Para poder aprovechar toda esta energía se usan celdas fotovoltaicas las cuales generan electricidad directamente de la luz visible, esta puede variar a lo largo del día, por lo cual se crea la necesidad de poder saber cuál es la generación de electricidad de la celda a lo largo de un día.

La adquisición de datos a lo largo del día de una celda fotovoltaica nos permite conocer los parámetros eléctricos que describen su funcionamiento, Parámetros que determinan la calidad del dispositivo y permiten realizar estudios comparativos entre módulos, estudios de degradación en el tiempo e inferir problemas en la generación de carácter externo al panel fotovoltaico, como ser sombras parciales o totales, contactos defectuosos, al igual que nos permite comparar su generación según la época del año.

Para poder implementar la adquisición de datos se utilizan los lenguajes de alto nivel se para diseño de interfaces que permitan el monitoreo y control.

Específicamente el uso de plataformas como LabView, que permite realizar interfaces amigables para el usuario, de manera que no se requiera un nivel de capacitación especializado.

La implementación de una interfaz, junto con un panel fotovoltaico, permite el registro de datos, presentación de información en línea, control de las variables de monitoreo. [1][5][6][7].

2. Adquisición de datos a través de LabView y Arduino

El sistema SCADA usualmente presenta la información al personal operativo de manera gráfica, en forma de un diagrama de representación.

2.1 Arduino- Labview

Para realizar nuestra adquisición de datos se usó una tarjeta Arduino, esta es una placa con un microcontrolador Atmel AVR y puertos de entrada y salida.

Junto con Arduino se utilizó el software de LabView (Laboratory Virtual Instrumentation Engineering Workbench) que es una plataforma para el desarrollo de sistemas, con un lenguaje de programación gráfico.

La combinación de estas dos herramientas nos permite realizar un sistema de adquisición de datos en tiempo real.

Todo depende del protocolo de comunicación con el cual se quiera comunicar, National Instruments ofrece toolkits para comunicarse con el dispositivo y poder configurarlo sin necesidad de escribir código en C. Si lo único que se quiere es enviar y/o recibir información del Arduino a la PC y viceversa, se pueden utilizar las librerías de VISA para hacerlo por el puerto serie.

La implementación de un sistema SCADA, acrónimo de Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos) es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia. Facilita retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores), y controla el proceso automáticamente.

Provee de toda la información que se genera en el proceso productivo (supervisión, control calidad, control de producción, almacenamiento de datos, etc.) y permite su gestión e intervención.

Por otra parte, es posible visualizar la programación con una interfaz Hombre - Máquina o HMI ("Human Machine Interface") que presenta los

datos a un operador (humano) y a través del cual éste controla el proceso, se les conoce también como software HMI o de monitorización y control de supervisión. Las señales del proceso son conducidas al HMI por medio de dispositivos como tarjetas de entrada/salida en el ordenador

Existe una "Estación Maestra" y se refiere al servidor y al software responsable para comunicarse con el equipo del campo en estos se encuentra el software HMI corriendo para la estación de trabajo [4].

2.2 Metodología

Nuestro programa para la adquisición de datos está diseñado en el software de LabView quien a través de un panel frontal podemos visualizar el **monitoreo de la señal en HMI**, vinculado con una tarjeta Arduino quien se encarga de recibir la señal analógica enviada por una celda solar.

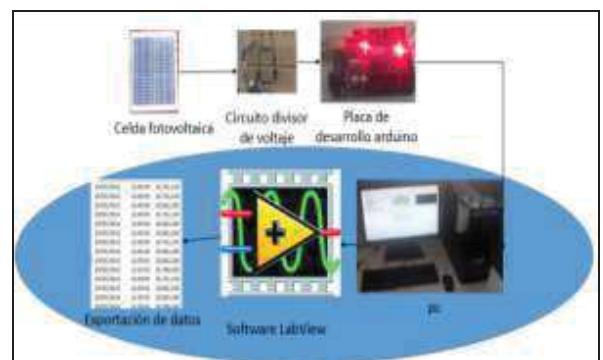


Fig.1. Diagrama a bloques del programa.

Para realizar la lectura del voltaje generado en nuestra celda se implementó un divisor de voltaje (el cual se normalizó) reduciéndolo a un parámetro permisible para nuestro Arduino como lo muestra la figura N°2[2], esto debido a que la tarjeta Arduino solo soporta un máximo de 5VDC.

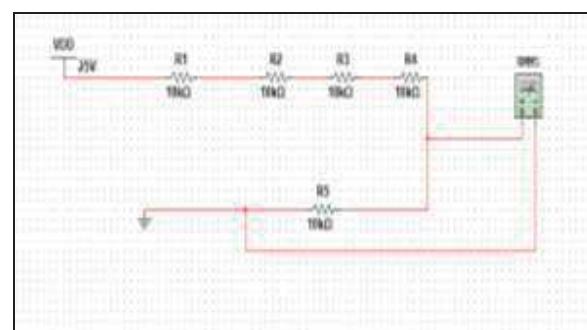


Fig. 2. Divisor de voltaje.

Con la captura del voltaje generado se continuo con la graficación de los datos obtenidos en LabView como se muestra en la figura N° 3, y estos a su vez se almacenan en una tabla por medio de un arreglo donde se exportan a formato “.xls” y “.txt” como se muestra en la figura N° 4A y 4B.

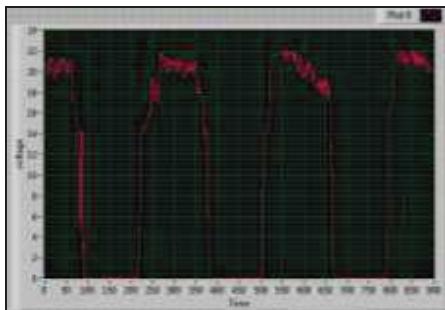


Fig.3. Datos obtenidos en tiempo real, mostrados en el HMI apoyándose de Labview

DATOS DE LA CELDA SOLAR		
Archivo	Edición	Formato
26/05/2014	16:42:25	19,38048
26/05/2014	16:42:26	19,35696
26/05/2014	16:42:27	19,55696
26/05/2014	16:42:28	19,33344
26/05/2014	16:42:29	19,33344
26/05/2014	16:42:30	19,38992
26/05/2014	16:42:31	19,33344
26/05/2014	16:42:32	19,33344
26/05/2014	16:42:33	19,38992
26/05/2014	16:42:34	19,38992
26/05/2014	16:42:35	19,38992
26/05/2014	16:42:36	19,2864
26/05/2014	16:42:37	19,38992
26/05/2014	16:42:38	19,2864
26/05/2014	16:42:39	19,30992
26/05/2014	16:42:40	19,2864
26/05/2014	16:42:41	19,2864
26/05/2014	16:42:42	19,26288
26/05/2014	16:42:43	19,26288
26/05/2014	16:42:44	19,26288
26/05/2014	16:42:45	19,26288
26/05/2014	16:42:46	19,23936
26/05/2014	16:42:47	19,23936
26/05/2014	16:42:48	19,23936
26/05/2014	16:42:49	19,23936
26/05/2014	16:42:50	19,23936

Fig.4.A Archivo”.txt”

Portapapeles				Fuente
A1	B	C	D	26/05/2014
1 26/05/2014	09:07:12	1,606,416		
2 26/05/2014	09:10:01	1,665,216		
3 26/05/2014	09:15:01	2,027,424		
4 26/05/2014	09:20:01	201,096		
5 26/05/2014	09:25:01	2,018,016		
6 26/05/2014	09:30:01	2,015,664		
7 26/05/2014	09:35:01	2,020,368		
8 26/05/2014	09:40:01	2,015,664		
9 26/05/2014	09:45:01	2,062,704		
10 26/05/2014	09:50:01	2,067,408		
11 26/05/2014	09:55:01	206,976		
12 26/05/2014	10:00:01	2,074,464		
13 26/05/2014	10:05:01	2,076,816		
14 26/05/2014	10:10:01	2,062,704		
15 26/05/2014	10:15:01	2,079,168		
16 26/05/2014	10:20:01	208,152		
17 26/05/2014	10:25:01	2,086,224		

Fig.4.B. Archivo “.xls”

La medida de voltaje en el software de LabView se comprobó con una lectura de voltaje por medio de un multímetro con valores RMS como se muestra en la figura N° 5.

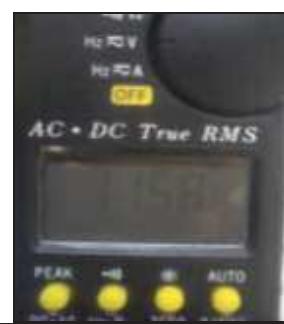


Fig.5. Lecturas de voltaje por medio de software y multímetro.

Mediante la adquisición de datos LabView, nos permite obtener la visualización, el monitoreo, la supervisión y el control del comportamiento de una celda solar, además podemos ser capaces de determinar la cantidad de energía que podemos generar por medio de una celda fotovoltaica. Esta la podemos determinar para un día, para una semana, un mes, por estaciones del año o al año y así conocer si se satisfacen nuestras necesidades de energía.

Para el cálculo de la energía recolectada se usó la exportación de datos en un archivo “.xls” y “.txt” con lo cual podemos calcular nuestra adquisición de energía.

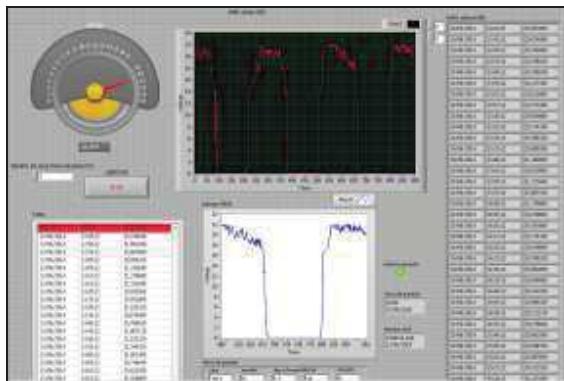


Fig.6. Programa de adquisición de datos en LabView.

El diagrama a bloques de la figura N° 7 muestra dos ciclos while estos están conectados a nuestros insert into Array, esta conexión nos da la opción de guardar los datos obtenidos en el momento en que se captura y sin necesidad de esperar el tiempo en el cual termina las mediciones nuestro experimento.

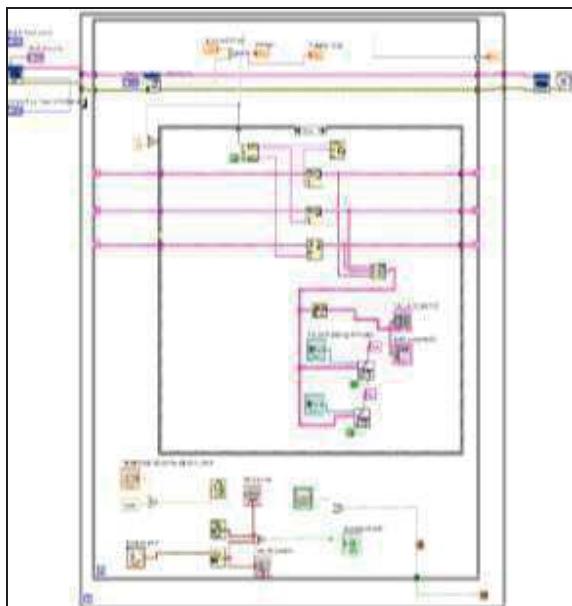


Fig.7. Diagrama a bloques en LabView.

Esta configuración se realizó para **prevenir afectaciones** en nuestra adquisición de datos por si ocurre una **falla de energía** o se apagan nuestra pc, o por si alguna persona externa a esto aborta la aplicación cuando se esté ejecutando la misma, permitiendo así que nuestros datos adquiridos no se pierdan.

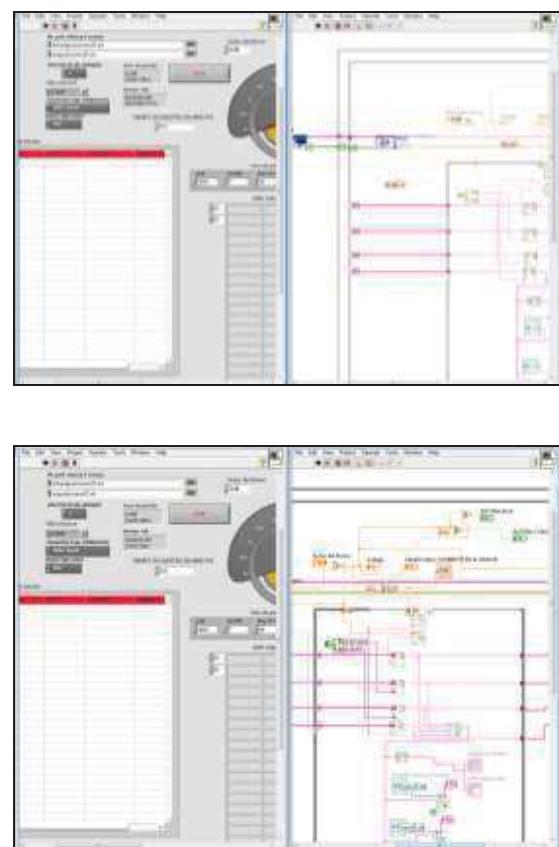
En este mismo diagrama se ve la inserción de una estructura case la cual permitió que los datos con valor de 0 volts no se exporten a nuestros documentos, con la finalidad de no crear datos en

exceso y que estos a su vez consuman demasiado rendimiento y espacio en nuestra PC, además permite no tomar mediciones cuando no se está generando energía. [3].

3. Análisis de resultados

En este trabajo se obtuvo un análisis de una señal, de manera gráfica por medio del software de LabView con el cual además se generó una base de datos, con la cual se puede trabajar implementando otras herramientas de análisis.

Con la implementación de este sistema se puede obtener un registro claro del comportamiento de una celda fotovoltaica a lo largo del día, esto puede realizarse de manera autónoma sin la necesidad de contar con la operación de alguna persona durante tiempos determinados (por minutos u horas) para generar la base de datos y así realizar un análisis experimental especializado.



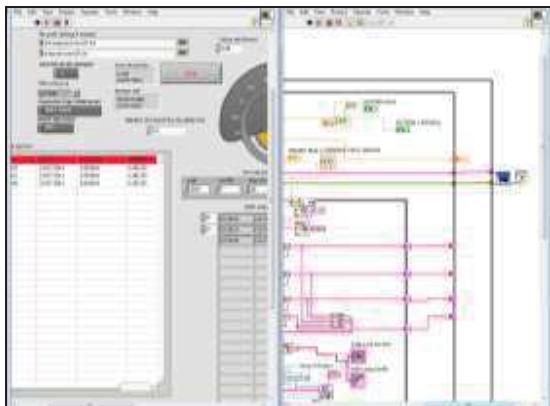


Fig.8. Análisis de la señal desde panel frontal y monitoreo de su trayectoria en el diagrama de bloques en LabView.

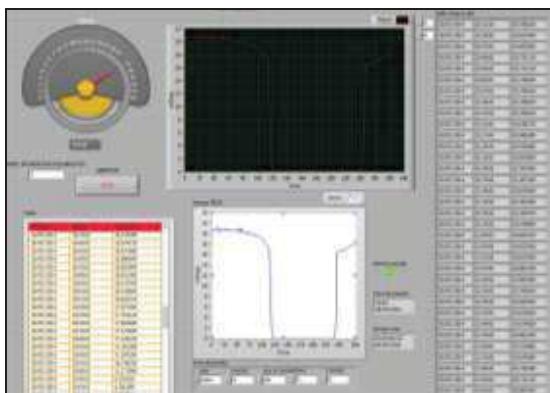


Fig.9. Adquisición de datos vistos en el panel frontal.

4. Conclusiones

Mediante el siguiente trabajo se pudo implementar la adquisición de datos para el análisis de una señal por medio de Arduino y LabView.

Con esto se pudo monitorear y tener un registro de la producción de energía de una celda fotovoltaica. Como nuestra entrada es una señal la cual se normalizo, este mismo análisis se puede efectuar para otro tipo de señales de las cuales se desea saber su comportamiento de manera gráfica.

Con la adquisición de datos, se pueden generar bitácoras que nos reporten el funcionamiento o las propiedades que queramos conocer y estar monitoreando por ejemplo, el comportamiento de una señal por medio de sensores, voltaje, corriente, temperatura, etcétera. Al igual, se puede implementar un control más exacto de la señal con la cual se está trabajando para obtener los resultados reales sobre los resultados ideales para los distintos procesos.

Con los conocimientos adquiridos durante nuestra formación mecatrónica se pudo implementar nuestro trabajo, al igual nos dio las bases para el desarrollo e implementación de sistemas automatizados que permitirán aplicarse en el desarrollo de la región Sierra Nororiental del Estado de Puebla.

Referencias

- [1] Méndez J, Cuervo R. "Energía Solar Fotovoltaica", Fc. España, 2°Ed, 2007.
- [2] Hay W, Kemmerly J, Durbin S. "Análisis de Circuitos e Ingeniería", McGraw-Hill, China, 7° Ed., 2007.
- [3] Del rio J, Shariat S, Sarria D."LabView Programación Para Sistemas De Instrumentación", Alfaomega, México, 1°Ed, 2012.
- [4] Rodríguez A. "Sistemas Scada", Alfaomega, México, 3°Ed, 2013.
- [5] Balfour J, ShawM, Bremernash N,"Introduction to Photovoltaic", Jones & Bartlett, United State Of America, 1° Ed, 2013
- [6] Firman A, Cáceres M, Toranzos V."monitoreo y caracterización de sistemas fotovoltaicos conectados a red", IV congreso brasileño de energía solar, INENCO universidad nacional de salta, pag.1, sao Paulo, 2012.
- [7] Munera J."sistema de control automático para monitorear y controlar el ambiente al interior de un invernadero", Universidad pontificia bolivariana, pag.8, Medellin Colombia, 2012.

Simulador Basado en un Sistema Bípedo Simple

Cuatlaxahue-Formacio Sandra y Sánchez-Sánchez Pablo

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica
Av. San Claudio y 18 Sur, Tel: 2 29 55 00
cformaciods@gmail.com - leitable@ece.buap.mx

Resumen

El presente trabajo muestra el modelado matemático y la simulación de un sistema bípedo simple en los programas de Matlab® y SolidWorks®, donde se usan librerías específicas que permitan simular el modelo 3D del sistema que presenta una estructura similar a los sistemas robóticos utilizados en la rehabilitación para la extremidades inferiores como es el famoso caso de Lokomat®. Se despliega el cálculo de los modelos cinemático directo y dinámico, así como la aplicación de un controlador clásico en la robótica para el seguimiento de trayectorias propuestas con referencia a la marcha humana a través de polinomios de quinto orden. Durante el modelado y la simulación se tomara al sistema libre de interacción con el ambiente llevando al análisis del sistema en un caso ideal.

Palabras clave: Simulador, bípedo, cinemática, dinámica, marcha, polinomios.

1. Introducción

La movilidad de los robots en un entorno de trabajo se ha convertido en un factor significativo para la evolución de la robótica, los robots deben ser capaces de esquivar obstáculos y mantenerse estables a pesar de las perturbaciones existentes. Desde hace más de tres décadas, muchos investigadores se han centrado al estudio y desarrollo de dispositivos que pretenden imitar el sistema motriz de los humanos, este tipo de robos son conocidos comúnmente como bípedos, se pueden encontrar inicios de investigación en este campo por parte de los investigadores Ichiro Kato y Vokobratovic hasta llegar a las más avanzadas técnicas de control por parte de compañías industriales en la actualidad [1]. El diseño de mecanismos que tengan un movimiento parecido al movimiento humano y la comprensión de éste han llevado a mejorar el diseño de prótesis añadiendo elementos activos y la aplicación de sistemas

robotizados en la rehabilitación médica [2], tales como el sistema robotizado Lokomat desarrollado por Colombo, este sistema se caracteriza por la sincronización del control de un exoesqueleto de miembros inferiores, una banda móvil y la carga y descarga del peso corporal del paciente con el fin de obtener un movimiento con gran semejanza a la marcha humana durante las sesiones [3]; el desarrollo de exoesqueletos para rodilla y tobillo como los presentados en [4,5] como sistemas de apoyo durante la marcha o sistemas que son únicamente para la rehabilitación como el que podemos encontrar en [6] donde el sistema no es diseñado de forma horizontal para generar movimientos específicos que se encuentran en sesiones de rehabilitación manual.

Es claro que aún falta mucho por demostrar y entender la complejidad de la locomoción bípeda en los sistemas robóticos [7]; debido a este tipo de complejidad en la robótica, la modelización y la simulación se está convirtiendo en técnicas esenciales para que los ingenieros comprendan el comportamiento de los sistemas robóticos, estas técnicas permiten mejorar el diseño del sistema y eliminar errores antes de haber desarrollado ningún tipo prototípico [8].

Debido a los avances que hay en los robots bípedos y los beneficios que han proporcionado en el área médica, el presente trabajo está centrado al modelado y la simulación de un sistema bípedo de pocos grados de libertad como posible inicio de estudio y desarrollo de un prototipo físico similar a los exoesqueletos utilizados en la rehabilitación para personas que padecen algún tipo de discapacidad motriz en las piernas que hoy en día existen.

2. Metodología

El desarrollo del trabajo se divide en cuatro etapas: la primera es la obtención del modelo cinemático directo, descripción espacial de los

eslabones y efectores finales con respecto a un sistema de coordenadas de referencia fija con el uso de matrices de transformación homogénea; la segunda, es la obtención del modelo dinámico, ecuaciones matemáticas de movimiento del sistema usando el método de Euler-Lagrange; tercera, aplicación de control de posición, se aplica un controlador del tipo PD con compensación de gravedad con el fin de que el sistema mueva de un punto de inicio a uno final cada una las articulaciones en un tiempo determinado; y cuarta es la planificación de trayectorias a través de polinomios de quinto orden para obtener movimientos suaves con el uso de restricciones en las velocidades y aceleraciones en cada uno de los puntos que describen la trayectoria, para posteriormente aplicar control punto a punto y obtener el movimiento deseado durante la simulación.

La estructura del sistema bípedo está inspirado a los exoesqueletos de pocos grados de libertad como en el caso del sistema robótico Lokomat®, el sistema bípedo a simular cuenta con tres grados de libertad en cada pierna para visualizar el movimiento de cada articulación que compone en general las piernas humanas; y la base del sistema se encuentra situada en el eslabón que representa la cadera, ver Figura 1. El diseño en 3D del sistema bípedo se desarrolló en el programa de SolidWorks® y a través del método de elementos finitos que posee este programa se obtienen las características físicas del diseño, información fundamental para el simulador, y que por medio de programación orientada a objetos se envían archivos del diseño a Matlab® que son llamados con librerías específicas que permiten simular sistemas mecánicos [9], así con la comunicación entre estos dos programas se visualizan los datos de salida del sistema en gráficas y movimiento del diseño para su mejor entendimiento.



Fig. 1. Estructura del diseño en SolidWorks®

Durante el modelado y la simulación, se considera al sistema bípedo sin ningún tipo de interacción con el ambiente orillando a esto a identificar como variables de entrada al sistema los torques aplicados en cada articulación y como variables de salida solo las posiciones, velocidades y las aceleraciones articulares.

2.1 Modelos matemáticos

En este apartado de obtuvieron los modelos cinemático y dinámico del sistema bípedo, el primero de ellos da referencia a la descripción espacial de los eslabones con respecto al sistema de coordenadas fija situado en la base [10], y posteriormente con la ayuda de del primer modelo se obtienen las ecuaciones de movimiento del modelo dinámico [11]. Para el modelado se dividió el sistema completo en dos subsistemas que son el Sub-sistema de la Pierna Derecha {SPD} y el Sub-sistema de la Pierna Izquierda {SPI}, dando un enfoque a la solución del modelado desde una perspectiva de manipuladores robóticos.

Modelo cinemático. Con el método matricial de Denavit-Hartenberg y la definición de los sistemas de coordenadas mostradas en la Figura 2, se obtuvieron las matrices de transformación que representan la ubicación de los eslabones con respecto la base, de manera breve se muestran para cada pierna las matrices que les corresponden. Para {SPD}:

$${}^B A_{0d} = \begin{bmatrix} 0 & 0 & 1 & L \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

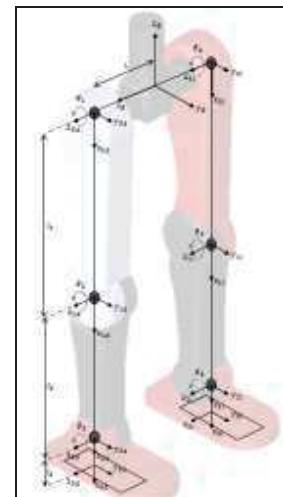


Fig. 2. Definición de sistemas de coordenadas

$${}^B A_{1d} = \begin{bmatrix} 0 & 0 & 1 & L \\ \operatorname{sen}(\theta_1) & \cos(\theta_1) & 0 & l_1 \operatorname{sen}(\theta_1) \\ -\cos(\theta_1) & \operatorname{sen}(\theta_1) & 0 & -l_1 \cos(\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$${}^B A_{2d} = \begin{bmatrix} 0 & 0 & 1 & L \\ S(\theta_{12}) & C(\theta_{12}) & 0 & l_2 S(\theta_{12}) + l_1 S(\theta_1) \\ -C(\theta_{12}) & S(\theta_{12}) & 0 & -l_2 C(\theta_{12}) - l_1 C(\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$${}^B A_{3d} = \begin{bmatrix} 0 & 0 & 1 & L \\ S(\theta_{123}) & C(\theta_{123}) & 0 & l_3 S(\theta_{123}) + l_2 S(\theta_{12}) + l_1 S(\theta_1) \\ -C(\theta_{123}) & S(\theta_{123}) & 0 & -l_3 C(\theta_{123}) - l_2 C(\theta_{12}) - l_1 C(\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

dónde $S(\theta_1) = \operatorname{sen}(\theta_1)$, $C(\theta_1) = \cos(\theta_1)$, $S(\theta_{12}) = \operatorname{sen}(\theta_1 + \theta_2)$, $C(\theta_{12}) = \cos(\theta_1 + \theta_2)$, $S(\theta_{123}) = \operatorname{sen}(\theta_1 + \theta_2 + \theta_3)$ y $C(\theta_{123}) = \cos(\theta_1 + \theta_2 + \theta_3)$.

Para {SPI} las matrices de transformación son las misma que las ecuaciones (1)-(4), pero estas están evaluadas con respecto a los ángulos θ_4 , θ_5 y θ_6 , y la distancia en el eje x es $-L$.

Modelo dinámico. Partiendo del modelo cinemático y el método de Euler-Lagrange se obtienen las ecuaciones de movimiento de cada subsistema {SPD} y {SPI} con la distribución de masas que se presentan en la Figura 3, estas son exactamente las mismas solo que cada una de ellas están evaluadas con respecto a los ángulos que les corresponde a cada articulación. Para {SPD}, en la articulación de la cadera se tiene:

$$\begin{aligned} r_3 = & [m_2 l_{c2}^2 + m_2 l_2^2 + m_2 l_{z2}^2 + m_2 l_{x2}^2 + 2(m_2 l_{cz2} l_2 + m_2 l_{z2} l_1) \cos(\theta_2) + m_2 l_{cz2}^2 \\ & + 2m_2 l_{cz2} l_2 \cos(\theta_1) + 2m_2 l_{cz2} l_1 \cos(\theta_2 + \theta_1) + l_1 + l_2 + l_3] \ddot{\theta}_1 \\ & + [m_2 l_{cz2}^2 + m_2 l_{cz2} l_2 \cos(\theta_2) + l_3] \ddot{\theta}_2 \\ & + [m_2 l_{cz2}^2 + m_2 l_{cz2} l_1 \cos(\theta_2 + \theta_1) + m_2 l_{cz2}^2 + 2m_2 l_{cz2} l_2 \cos(\theta_2) \\ & + m_2 l_{cz2} l_1 \cos(\theta_2 + \theta_2) + l_2 + l_3] \ddot{\theta}_3 \\ & + [m_2 l_{cz2}^2 + m_2 l_{cz2} l_2 \cos(\theta_2) + m_2 l_{cz2} l_1 \cos(\theta_2 + \theta_2) + l_3] \ddot{\theta}_4 \\ & - [2(m_2 l_{cz2} l_2 + m_2 l_{cz2} l_1) \sin(\theta_2) \dot{\theta}_1 \\ & + 2m_2 l_{cz2} l_2 \sin(\theta_2) \dot{\theta}_2 + 2m_2 l_{cz2} l_1 \sin(\theta_2 + \theta_2) (\dot{\theta}_2 + \dot{\theta}_3)] \dot{\theta}_1 \\ & - [2m_2 l_{cz2} l_2 \sin(\theta_2) \dot{\theta}_3 + m_2 l_{cz2} l_1 \sin(\theta_2 + \theta_2) (\dot{\theta}_2 + \dot{\theta}_3)] \dot{\theta}_2 \\ & - [m_2 l_{cz2} l_2 \sin(\theta_2) \dot{\theta}_3 + m_2 l_{cz2} l_1 \sin(\theta_2 + \theta_2) (\dot{\theta}_2 + \dot{\theta}_3)] \dot{\theta}_3 \\ & + [m_2 l_{cz2} + m_2 l_1 + m_2 l_2] g \sin(\theta_2) + [m_2 l_{cz2} + m_2 l_2] g \sin(\theta_2 + \theta_2) \\ & + m_2 l_{cz2} g \sin(\theta_2 + \theta_2 + \theta_3) \quad (5) \end{aligned}$$

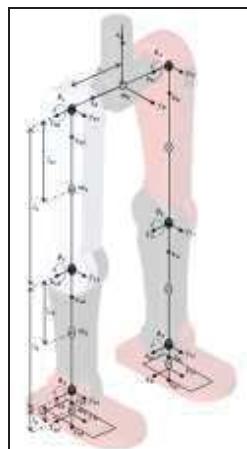


Fig. 3. Distribución de masas

En la rodilla:

$$\begin{aligned} r_2 = & [m_2 l_{c2}^2 + m_2 l_2^2 + (m_2 l_{cz2} + m_2 l_2) l_1 \cos(\theta_2) + m_2 l_{cz2}^2 + 2m_2 l_{cz2} l_2 \cos(\theta_2) \\ & + m_2 l_{cz2} l_1 \cos(\theta_2 + \theta_3) + l_2 + l_3] \ddot{\theta}_1 \\ & + [m_2 l_{cz2}^2 + m_2 l_{cz2} l_2 + m_2 l_{cz2}^2 + 2m_2 l_{cz2} l_2 \cos(\theta_3) + l_2 + l_3] \ddot{\theta}_2 \\ & + [m_2 l_{cz2}^2 + m_2 l_{cz2} l_2 \cos(\theta_2 + \theta_3)] \ddot{\theta}_3 \\ & + [(m_2 l_{cz2} + m_2 l_2) l_1 \sin(\theta_2) \dot{\theta}_1 \\ & + m_2 l_{cz2} l_1 \sin(\theta_2 + \theta_3) \dot{\theta}_1 - 2m_2 l_{cz2} l_1 \sin(\theta_2) \dot{\theta}_2 - [2m_2 l_{cz2} l_1 \sin(\theta_2) \dot{\theta}_3] \dot{\theta}_1 \\ & - [m_2 l_{cz2} l_1 \sin(\theta_2 + \theta_3) \dot{\theta}_3 + [m_2 l_{cz2} + m_2 l_2] g \sin(\theta_2 + \theta_3) \\ & + m_2 l_{cz2} g \sin(\theta_2 + \theta_2 + \theta_3)] \quad (6) \end{aligned}$$

Y en el tobillo:

$$\begin{aligned} r_5 = & [m_2 l_{cz5}^2 + m_2 l_{cz5} l_2 \cos(\theta_5) + m_2 l_{cz5} l_1 \cos(\theta_2 + \theta_5) + l_3] \ddot{\theta}_1 \\ & + [m_2 l_{cz5}^2 + m_2 l_{cz5} l_2 \cos(\theta_5) + l_3] \ddot{\theta}_2 + [m_2 l_{cz5}^2 + l_3] \ddot{\theta}_3 \\ & + [m_2 l_{cz5} l_1 \sin(\theta_2 + \theta_5) \dot{\theta}_1 \\ & + m_2 l_{cz5} l_1 \sin(\theta_2 + \theta_5) \dot{\theta}_2 + 2m_2 l_{cz5} l_2 \sin(\theta_5) \dot{\theta}_1 \\ & + [m_2 l_{cz5} l_2 \sin(\theta_5) \dot{\theta}_2 + m_2 l_{cz5} g \sin(\theta_2 + \theta_2 + \theta_5)] \quad (7) \end{aligned}$$

2.2 Estructura de control

Este apartado se divide en dos etapas, la primera es la aplicación de control de posición y la segunda es control de trayectoria. El controlador en lazo cerrado propuesto es el PD con compensación de gravedad debido a su estabilidad asintótica global que presenta en manipuladores de n grados de libertad, ya antes demostrado en los 80's. La ecuación general de este controlador es:

$$\tau = K_p \ddot{\theta} + K_v \dot{\dot{\theta}} + g(\theta) \quad (8)$$

donde la ley de control Ec. (8) necesita información de las posiciones y velocidades deseadas y la medición de las posiciones y velocidades de salida del sistema a cada instante.

Control de posición. El objetivo es hacer mover a cada articulación del sistema bípedo de una posición inicial a una posición deseada en un tiempo determinado. Entonces el control de posición debe determinar el torque de tal forma que [11]:

$$\lim_{t \rightarrow \infty} \theta(t) = \theta_d \quad (9)$$

Control de trayectoria. El objetivo es mover las articulaciones del sistema por una trayectoria descrita por una secuencia de puntos en su espacio de trabajo en un tiempo determinado [11], aplicando la misma ley de control.

2.3 Planificación de trayectorias

Un objetivo particular del simulador es hacer que el sistema realice movimientos parecidos a la marcha humana como los que presenta los exoesqueletos usados en la rehabilitación médica.

Entonces, como referencia para la definición de la trayectoria deseada es tomada de las gráficas de movimiento de las articulaciones en el plano sagital que presenta un grupo de personas durante un ciclo de la marcha, Figura 4. Donde dependiendo del rango del ángulo en que se encuentre la articulación, el movimiento se denomina como flexión o extensión [12].

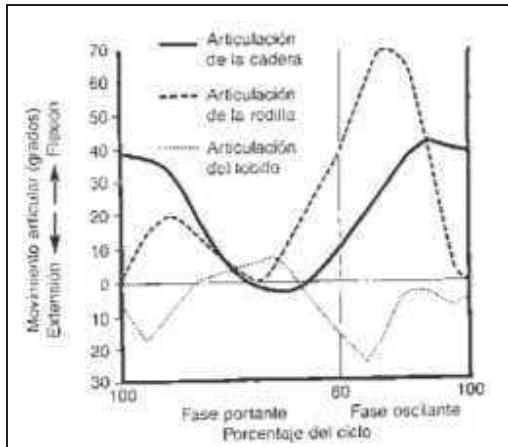


Fig. 4. Gráficas de las articulaciones en un ciclo de marcha

Teniendo la definición del movimiento deseado, se utilizaron polinomios de quinto orden (Ec. (10)) para generar la vía de puntos que definen la trayectoria, bajo las especificaciones de que las velocidades y las aceleraciones en cada punto deben ser nulas para obtener un movimiento suave [13].

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (10)$$

Después de varias pruebas, los polinomios que cumplieron los requisitos establecidos fueron graficados para visualizar la trayectoria que generan, Figuras 5-7.

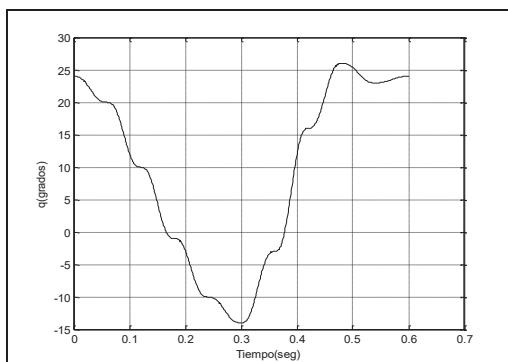


Fig. 5. Trayectoria propuesta para la cadera

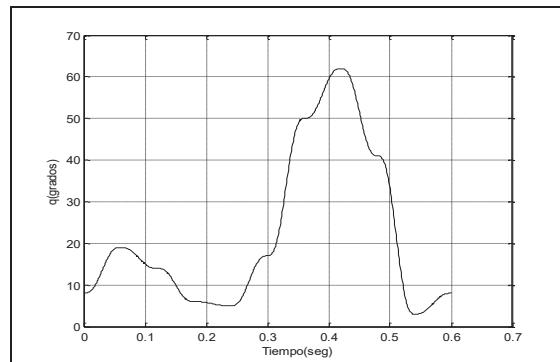


Fig. 6. Trayectoria propuesta para la rodilla

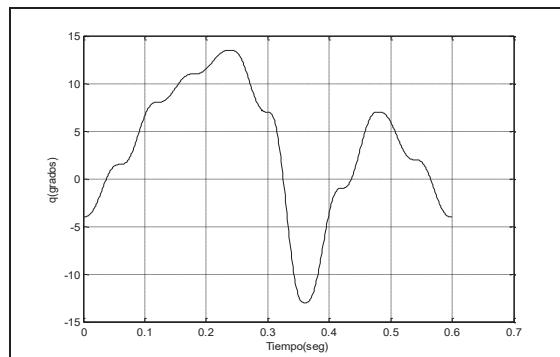


Fig. 7. Trayectoria propuesta para el tobillo

3. Análisis de resultados

Tras llevar a cabo todo lo anterior al simulador, se obtuvo el movimiento mostrado en la Figura 8. En las cinco imágenes superiores de la figura, se puede observar las dos fases que componen el ciclo de marcha, mientras la pierna derecha presenta la fase de apoyo, la pierna izquierda exhibe la fase de oscilación, y de manera viceversa se observa en las cinco imágenes inferiores.



Fig. 8. Secuencia de movimiento de un ciclo de marcha.

Las gráficas de salida de las posiciones articulares de la cadera, rodilla y tobillo de {SPD} son mostradas en las Figuras 9-11. La parte superior de estas figuras se presenta las posiciones y en la parte inferior se presenta las señales de error, diferencia entre la posición deseada y la de salida.

Las gráficas de salida de las posiciones articulares de {SPI} presentan la misma trayectoria que en las Figuras 8-10 pero estas tienen un punto de inicio diferente debido a la sincronización de las fases durante la marcha [12].

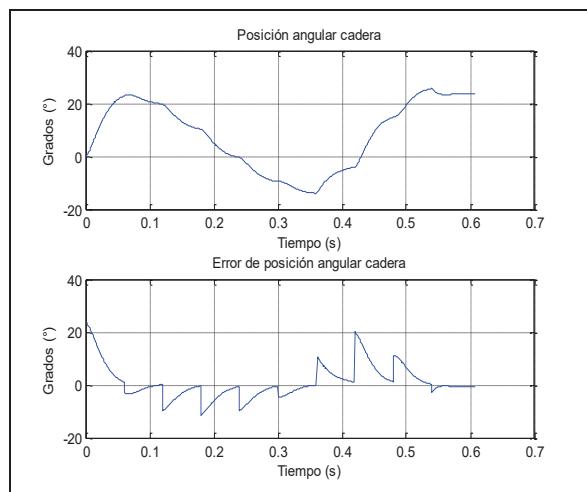


Fig. 9. Posición angular de salida y señal de error en la cadera

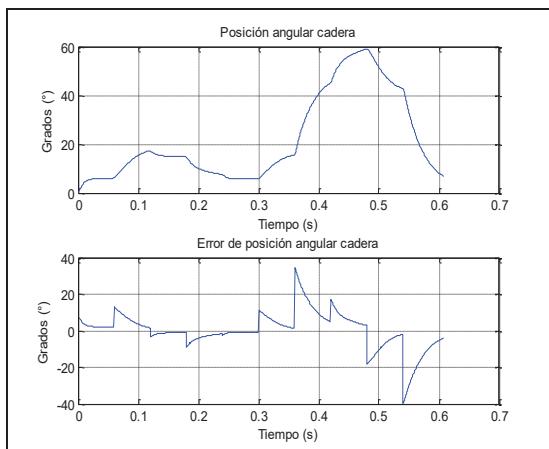


Fig. 10. Posición angular de salida y señal de error en la rodilla

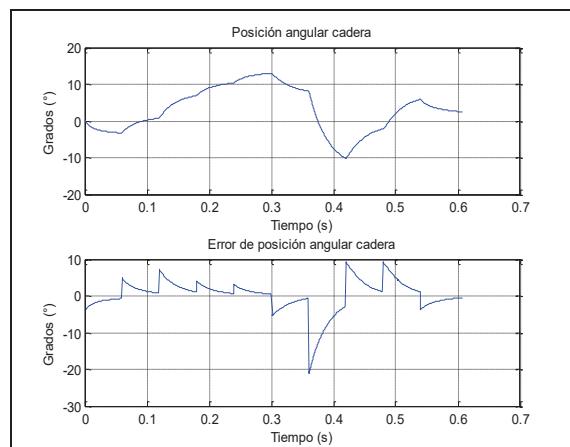


Fig. 11. Posición angular de salida y señal de error en el tobillo

4. Conclusiones

La deducción de los modelos cinemático directo y dinámico del sistema bípedo fue exitosa junto con la aplicación del controlador PD con compensación de gravedad, se pudo generar un movimiento muy similar a la de la marcha humana donde la diferencia entre las trayectorias deseada y las obtenidas en la simulación fue mínima, cabe mencionar que las trayectorias propuestas son una de tantas que pueden ser simuladas con la condición de que cumplan el rango y patrón de movimiento de la marcha normal.

El trabajo presentó una alternativa de la solución de análisis de un sistema bípedo de pocos grados de libertad, y con los resultados obtenidos, el sistema bípedo puede ser punto de apoyo para el análisis de una ótesis con aplicación de rehabilitación, tomando en cuenta el análisis más completo de los modelos matemáticos las fuerzas de fricción, las resistencias al movimiento y otras variables que dependan del tiempo y la velocidad, todo esto con el propósito al desarrollo físico de un prototipo.

Referencias

- [1] Kajita S., Espiau B., “Legged robots”; en Siliciano B. and Khatib O., “Handbook of robotics”, Springer, 2008.
- [2] Machiel Van der Loos H. F., Reinkensmeyer D. J., “Rehabilitation and health care robotics”; en Siliciano B. and Khatib O., “Handbook of robotics”, Springer, 2008.

- [3] Riener Robert, “*Technology of the Robotic Gait Orthosis Lokomat*”, Neurorehabilitation Technology, Springer, 2012.
- [4] R. López, J. Torres, S. Salazar y R. Lózano, “*Desarrollo de un exoesqueleto para rehabilitación de tobillo y rodilla*”, Revista Mexicana de ingeniería biomédica, Vol. 35, No. 1, Abril 2014.
- [5] Yanjiao Ma, Wei He and Shuzhi Sam Ge, “*Modeling and Control of a Lower-Limb Rehabilitation Robot*”, ICSR 2012, International Conference on Social Robotics 2012, Springer-Verlag Berlin Heidelberg.
- [6] Guzmán Valdivia C. H., Blanco Ortega A., Oliver Salazar M. A. y Azcaray Rivera H. R., “*Modelado y simulación de un robot terapéutico para la rehabilitación de miembros inferiores*”, Revista Ingeniería Biomédica, Vol. 7, No. 14, Colombia, 2013.
- [7] Guang Yi Lu P., “*Mechanical Engineering Challenges in humanoid robotics*”, M.I.T., U.S.A., 2011.
- [8] The MathWorks. “*Programación de robots con Matlab y Simulink*”, en www.mathworks.com.
- [9] The MathWorks. “*SimMechanics User Guide*”, 2010; obtenida el 30 de enero del 2014 de www.mathworks.com.
- [10] Fu K. S., González R. C., Lee C. S. G., “*Robótica: control, detección, visión e inteligencia*”, McGraw-Hill, U.S.A., 1987.
- [11] Kelly R., Santibáñez V., “*Control de movimiento de robots manipuladores*”, Prentice Hall, Madrid, 2003.
- [12] Nordin M., Frankel V. H., “*Biomecánica básica del sistema musculoesquelético*”, McGraw Hill, España, 2004.
- [13] Spong M. W., Hutchinson S., Vidyasagar M., “*Robot Dynamics and control*”, 2da. Edición, 2004.

Análisis de la Precisión en los Desplazamientos de una Plataforma Móvil por Control de Motores DC

¹Apolinar Garnica Erik, ¹Calderón Rocha José Carlos, ¹González Torres Martín Eduardo, ¹Rodríguez Aviña Juan Pablo, ¹Caudillo Estrada Juan José, ^{1,2,3}y León-Rodríguez Miguel.

¹Universidad Politécnica del Bicentenario, ²Universidad de Santiago de Chile, ³Centro de Investigaciones en óptica. carloscalderonrocha4@hotmail.com

Resumen

Típicamente, en la robótica móvil se utilizan motores de características especiales que facilitan su implementación en las tareas que realizan. Sin embargo, el costo incrementa y la complejidad del sistema también. En este manuscrito se presenta el análisis de la precisión en desplazamientos de una plataforma autónoma móvil. El control de los motores de DC es realizado por medio del micro controlador Pic 18f4550, que de manera sistemática permite el manejo de tiempos de las señales de control. El desplazamiento lineal mínimo de este sistema autónomo, correspondiente a una señal pulso de duración de 11ms, es de 1.473 mm y para un desplazamiento angular es de 0.5°. Se reporta de manera experimental que el error incrementa cuando en la estimación que modela su movimiento no es considerada la aceleración y permanece sin cambio este error cuando la velocidad permanece constante. Concluyendo y corroborando que una aceleración existe en el sistema que aún no ha sido considerada en el modelo al usar motores de DC. Las mediciones fueron realizadas en un ambiente de laboratorio con condiciones de iluminación controladas y un sistema de visión.

Palabras clave: Robótica móvil, Control digital, micro controladores.

1. Introducción

Actualmente, la robótica ha crecido en el área de robots móviles que típicamente, se han empleado para controlar movimientos ya sea lineales o angulares por medio de sistemas embebidos y micro-controladores [1] [2].

Dentro de la robótica móvil el principal dispositivo que permite el movimiento de sus articulaciones es el motor [1] [4]. Existe una gran variedad de motores aplicados en esta área, siendo los servomotores y motores a pasos los más utilizados para tales fines [4].

Así mismo los micro controladores son muy utilizados para controlar las direcciones y movimientos del sistema de manera versátil. El motor de DC es menos utilizado en la robótica móvil principalmente por: el torque pequeño, mayor complejidad al programar los movimientos, menor precisión en los movimientos, sin embargo una de los beneficios de utilizar este tipo de motores es un menor costo, fácil adquisición, funcionamiento silencioso y un mantenimiento sencillo [4].

En este manuscrito, presentamos el análisis de la resolución los desplazamientos de una plataforma autónoma móvil que utiliza dos motores de DC. Logrando moverlos por ciertos lapsos de tiempo obteniendo como respuesta mínima a un pulso de 11ms una distancia de 1.473 mm para movimientos lineales. En caso de los angulares, es de 0.5° para el mismo tiempo. Se reporta de manera experimental que el error incrementa, cuando en la estimación que modela su movimiento no es considerada la aceleración y permanece sin cambio este error cuando la velocidad permanece constante. Concluyendo y corroborando que una aceleración existe en el sistema que aún no ha sido considerada en el modelo al usar motores de DC. Una vez obtenidos estos datos se puede realizar un programa para que la plataforma recorra una trayectoria definida.

2. Metodología

Se creó un programa para el micro controlador con la finalidad de controlar el tiempo que permanecen en operación los motores del sistema. Esto nos permitirá tener información para determinar velocidades, aceleraciones y algunas otras respuestas temporales de interés que modelan el sistema.

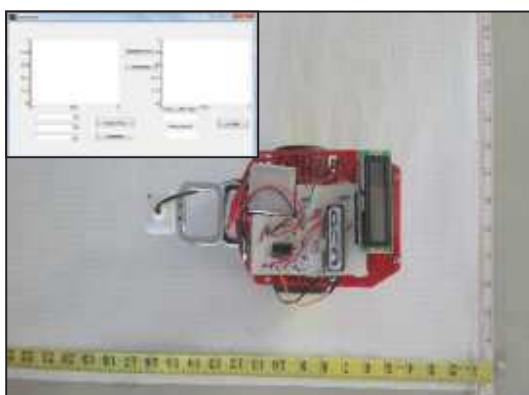


Figura 1. Sistema completo



Figura 2. Plataforma móvil

En las Figuras 1 y 2 se observa el diseño de la plataforma autónoma móvil. Esta contiene un sistema de tracción de tres llantas, dos de ellas emparejadas en uno de sus extremos mientras que al otro extremo la restante. Esta última es una rueda loca para soporte, a diferencia de las anteriores que están unidas a los ejes de los motores de DC.

Con fines de acoplamiento se conectó un puente H (L293) a los motores. El movimiento de la plataforma autónoma móvil se realizó con datos ingresados directamente al Pic, los cuales accionaran las llantas dependiendo de las condiciones del programa.

La investigación se realizó en las instalaciones del laboratorio de robótica de la UPB ubicado en el municipio de Silao, Guanajuato. Se adecuó este lugar para la correcta realización de la investigación.

Se construyó la estructura de un robot móvil tal como se mencionó anteriormente, el cual nos servirá como objeto de pruebas, este robot tendrá la tarea de trasladarse de un punto a otro. A si mismo se construyó un circuito de control para el robot usando un micro controlador PIC18f4550 programándose en la plataforma de PIC C Compiler.

La calibración fue determinada con un sistema de visión para el mínimo tiempo de respuesta (11 ms). Este sistema de visión y su interfaz se muestra en la Figura 1el cual estuvo soportado en ambiente de Matlab.

2.1 Calibración utilizando el sistema de visión

Esta calibración se desarrolló por medio de mediciones entre pixeles de imágenes obtenidas, utilizando las dimensiones físicas de una cuadricula como referencia (7x7 mm), Como se muestra en la Figura 3.

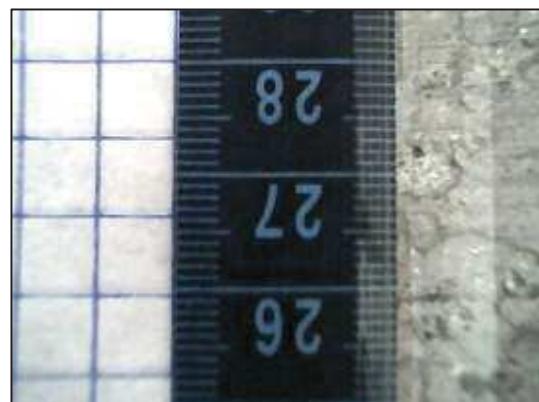


Figura 3. Posición de inicio



Figura 4. Posición final

Las Figuras 3 y 4 anteriores muestran la distancia que la plataforma ha recorrido con un pulso de 110 ms, dicha distancia es de 11.34 mm.

La fórmula utilizada para calcular la resolución del sistema es la siguiente:

$$d = \sqrt{(x_f - x_o)^2 + (y_f - y_o)^2}, \quad (1)$$

donde d es la distancia requerida expresada en pixeles, x_f es el número de pixel en dirección x de la posición final, x_o es el número de pixel medido en x de la posición inicial, y_f es el número de pixel medido en y de la posición final y y_o es el número de pixel medido en x de la posición inicial.

De modo que si se tiene un tiempo mínimo de respuesta de 11ms, una distancia entre pixeles de 19.99 de desplazamiento y 95 Pixeles corresponden a 7mm, entonces tendremos que la distancia recorrida es de 1.473 mm. Siguiendo la misma metodología se determinó una resolución angular de 0.5° para la mínima respuesta.

A continuación se muestra el procedimiento para determinar la respuesta temporal de los motores de DC al estímulo de una señal cuadrada. Para esto se utilizó un osciloscopio con dos puntas de prueba que se conectaron en cada motor. El primer comportamiento a determinar, en la respuesta temporal al estímulo de una señal cuadrada en los motores de DC fue, el tiempo que tarda en alcanzar el voltaje recomendado desde un nivel bajo, esto es de 0 a 5 v. La Figura 7 muestra esta respuesta. Se puede ver que el tiempo que tarda la señal de voltaje en subir es de aproximadamente 4ms. En la Figura 8 se muestra que un tiempo aproximado de 4ms le lleva a la señal descender de un nivel alto a un nivel bajo de voltaje. Esta respuesta corresponde para

movimientos lineales de la plataforma. En la Figuras 5 y 6 se muestran las correspondientes respuestas para los desplazamientos angulares. Esta respuesta de manera similar a la anterior es de 4ms en cada transición. Nótese también que tanto el inicio en las transiciones como el final son similares en ambos motores. Esta condición permitirá, también, ser considerada para la construcción de un modelo que describa con mayor exactitud la dinámica del sistema.

El modelo preliminar que se tiene para este sistema es el siguiente:

$$\begin{aligned} vdt &= ds \quad (\text{mov. lineal}) \\ wdt &= dr \quad (\text{mov. angular}), \end{aligned} \quad (2)$$

donde, v es la velocidad instantánea, dt es el gradiente temporal, ds el gradiente de posición lineal, w la velocidad angular instantánea y dr el gradiente de posición angular.

Si se desea una mayor precisión tendremos que considerar un modelo no lineal, condiciones iniciales como de frontera. Tal situación no contemplada en este trabajo.

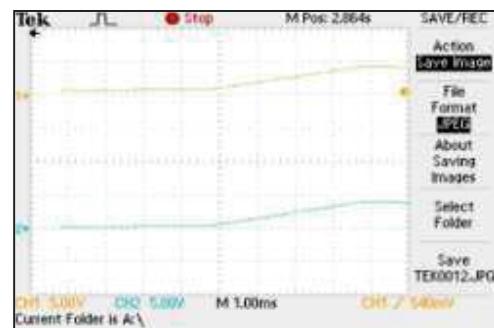


Figura 5. Transición de nivel bajo a alto al girar

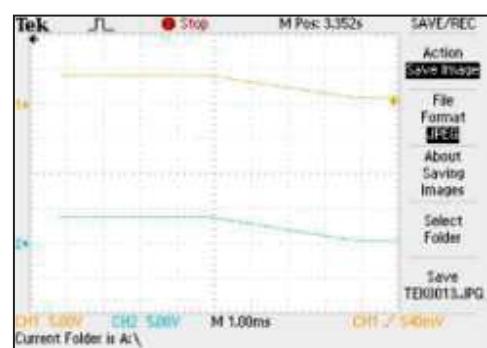


Figura 6. Transición de nivel alto a bajo al girar

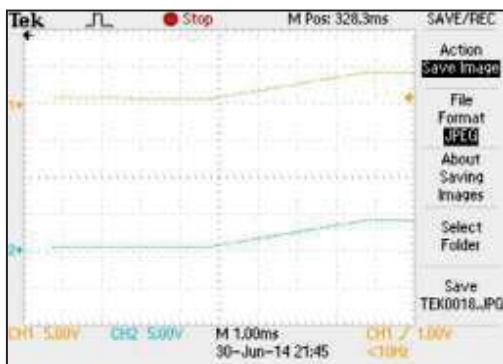


Figura 7. Transición de Subida lineal

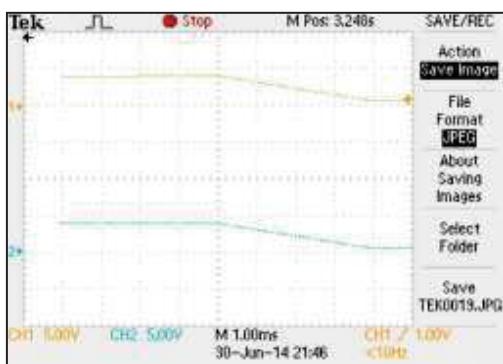


Figura 8. Transición de alto a Bajo lineal

3. Resultados experimentales

A continuación se muestran los resultados obtenidos del sistema autónomo móvil modelado en la Ec. 2. En la primer gráfica de la Figura 9 se muestra la distancia estimada tomando como referencia la resolución obtenida en la sección anterior en el modelo matemático. En esta misma gráfica se muestra, también, la distancia que ha recorrido la plataforma de manera experimental. La diferencia máxima entre estos resultados es de 23.49 mm correspondiente para el tiempo de 1430 ms.

En la gráfica central de la Figura 9 se muestra el comportamiento de la velocidad instantánea experimental del móvil. Puede verse un comportamiento no lineal para las primeras 6 velocidades instantáneas, principal motivo de la diferencia. Este comportamiento supone la existencia de aceleraciones dentro del sistema que no se contemplan y se corroboran en la gráfica de la misma Figura 9 en la parte inferior. Se supone que el principal motivo de la existencia de la aceleración

es debido a los tiempos de transición en los motores.

En la Figura 10 se muestra de manera similar los desplazamientos angulares, velocidades angulares instantáneas y aceleraciones instantáneas cuyo comportamiento es muy similar a su contraparte lineal descrita anteriormente. La diferencia máxima entre estos resultados es de 120 gra correspondiente para el tiempo de 1430 ms. Un comportamiento adicional, se ve involucrado para estos desplazamientos angulares, no identificado, lo presenta la gráfica de la aceleración angular.

Se programó una trayectoria de circuito cuadrado para el móvil, llegando a una posición estimada dentro del rango de error.

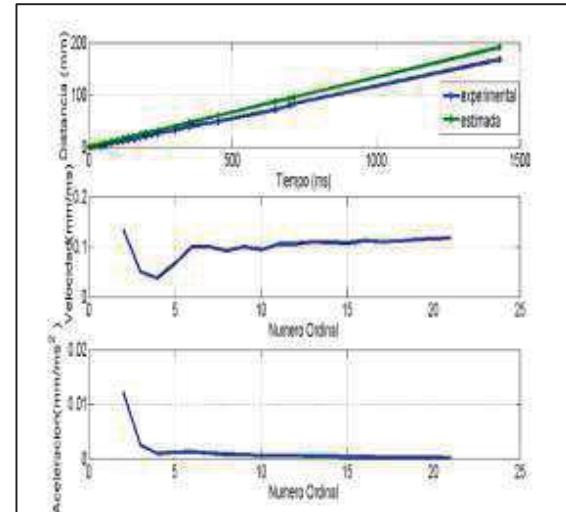


Figura 9. Grafica de posiciones, velocidades y aceleraciones.

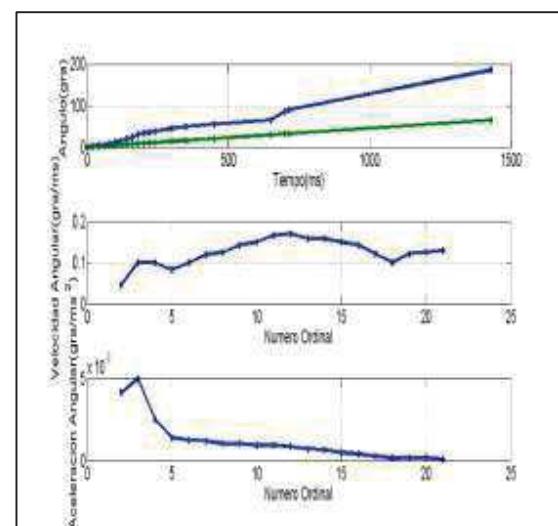


Figura 10. Grafica de los desplazamientos angulares

4. Conclusiones

En este trabajo se ha presentado una plataforma autónoma móvil, la cual realiza sus desplazamientos mediante tiempos programados debidamente en un microcontrolador que está conectado con la etapa de potencia de los motores.

El procedimiento realizado para esta plataforma fue en primera instancia la de programación de un código en la plataforma de Matlab para el control de los movimientos y respectivas pruebas experimentales realizadas. Posteriormente se construyó una placa donde se acoplaron todos los dispositivos que involucran el control de la plataforma. Es posible usar motores de DC en sistemas robóticos móviles y llegar a estimaciones de las posiciones con alto grado de exactitud según el modelo del sistema sea. En este trabajo se presentó un modelo simple para estimar las posiciones finales llegando a entender las causas de error que provocan una posición diferente. También determinamos a manera experimental la resolución de desplazamiento lineal del sistema (1.473 mm) y el angular (0.5°).

5. Bibliografía

- [1] Kumar S. “*Introducción a la robotica*”. McGraw-Hill, 403, 2008.
- [2] Craig J. “*Introduction to robotics mechanics and control*”. Addison Wesley Longman, 1989.
- [3] Hibbeler R. “*Mecánica para ingenieros. Dinamica*”, Macmillan, 1992.
- [4] Suarez J. “*Robot móvil para transporte automatizado*”, memoria presentada para optar el título de ingeniero en electrónica, badajoz, diciembre 2001

Velocidad de Respuesta de Brazo Robótico Utilizando Sensor Kinect

*¹Rodríguez Sánchez Eder Germain, ¹Martínez Gracia Cristian Roberto, ¹Moreno Castro María Viridiana, ¹Jiménez Juárez Jorge Alberto, ¹Banda Sierra Jonathan, ¹González Esquivel Arturo, ¹Preciado Prieto Alberto y ^{1,2,3}León Rodríguez Miguel.

¹Universidad Politécnica del Bicentenario, ²Centro de Investigaciones en Óptica, ³Universidad de Santiago de Chile.

*e-mail: erodriguezs_iro@upbicentenario.edu.mx

Resumen

La investigación realizada es acerca de la Realidad Virtual, la realidad virtual es por lo general un mundo virtual generado por ordenador en el que el usuario tiene la sensación de estar en el interior de este mundo. La principal aportación del proyecto es conocer las velocidades o tiempo de reacción entre el procesamiento de información del brazo robótico y el brazo del cuerpo humano. El sistema de experimentación empleado cuenta con un prototipo de brazo robótico, un software de simulación empleado a través de la programación de datos del sensor, y finalmente de la parte humana, es decir, el brazo humano. La imitación de movimientos del brazo robótico con el brazo humano, detectado a través del sensor empleado, nos generó el mayor tiempo de respuesta de 6 ms y velocidad mayor de 73.48 rad/s r respecto a la velocidad media del brazo robótico, estos datos fueron satisfactorios al realizar las pruebas experimentales, obteniéndose el resultado esperado.

Palabras clave: Realidad Virtual, Cinemática Directa, Robótica.

1. Introducción

Desde épocas muy remotas el hombre ha tenido la necesidad de construir máquinas que sean capaces de realizar tareas repetitivas o peligrosas. Los elementos que funcionan automáticamente se utilizan desde épocas tan remotas como la antigua Grecia. Sin embargo, es hasta mediados del siglo XX cuando se lograron materializar los primeros robots industriales [1]. Los robots industriales ocupan un lugar destacado dentro de la automatización de la producción y su protagonismo se ha ido consolidando en los últimos años [2] Despues de una caída en las ventas en la década de los noventa, el mercado de robots ha mantenido un crecimiento constante, en especial en Europa y Estados Unidos. Por su parte, Asia y particularmente Japón, sigue siendo líder nivel mundial [3]. Un brazo robótico es un mecanismo

con eslabones y articulaciones controladas y/o automatizadas, con funciones parecidas a las de un brazo humano; este puede ser la suma total del mecanismo o puede ser parte de un robot más complejo. Las partes de estos manipuladores o brazos son interconectadas a través de articulaciones que permiten, tanto un movimiento rotacional (tales como los de un robot articulado), como un movimiento translacional o desplazamiento lineal [4], [5]. El movimiento de brazos robóticos simples es por medio de motores. Un motor transforma algún tipo de energía (eléctrica, de combustibles fósiles), en energía mecánica capaz de realizar un trabajo [6]. Los motores empleados en este trabajo fueron servomotores. Los servomotores se caracterizan por su capacidad para posicionarse con precisión y de forma rápida en la posición deseada, dentro de su intervalo de operación, por medio de una señal PWM de control [6]. El movimiento realizado corresponde con el análisis de la cinemática directa. Este análisis está fundamentado en el álgebra vectorial y matricial para representar y describir la localización de un objeto en el espacio tridimensional con respecto a un sistema de referencia fijo. Dado que, un robot se puede representar como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia [7].

En esta investigación se reporta una caracterización preliminar usando un sensor Kinect para la imitación de un prototipo de brazo robótico con un brazo real a nivel experimental. Se alcanzó el grado de precisión del 85% en el desarrollo del sistema propuesto, tomando en cuenta las imprecisiones y el tiempo que nos llevó el realizar correctamente el sistema. Con los resultados obtenidos se puede concluir que el tiempo de respuesta nunca llegará a ser cero. Debido a que no se cuenta con un procesador que tenga una respuesta inmediata.

Todo esto para realizar la comunicación del sensor, la interfaz y el movimiento del brazo robótico.



Figura 6 Interfaz que muestra los ángulos de acuerdo a los datos obtenidos del sensor.

El brazo no es de una persona en específico, el sensor es capaz de realizar la esqueletización en cualquier persona e identificar el brazo de la misma transmitir los datos y realizar la imitación.

El flujo de la señal y la interpretación se da de la siguiente manera. El sensor registra las posiciones del usuario (Figura 6). Estos datos son recibidos por el microprocesador, en este caso el LaunchPad, y éste a su vez, transmite la información a los servomotores que controlan el movimiento del brazo robótico.

Una vez que el sistema estaba en funcionamiento completo y las señales se registraban adecuadamente en la interfaz desarrollada, se procedió a realizar las pruebas experimentales. El tiempo de respuesta de dicho brazo, se calculó de acuerdo a 3 diferentes velocidades (v1, v2, v3) angulares que se muestran en la Tabla 2. Donde ti es el tiempo inicial, tf es el tiempo final, tt es la diferencia temporal entre ti y tf, pa es el punto inicial y pb es el punto final.

Tabla 2 Cálculo de velocidades angulares

	ti	tf	tt	pa	pb		vel.ang.
v1	00:09:08	00:09:27	00:00:19	90	180	real	82,57 rad/s
	00:09:08	00:10:05	00:00:57	90	210	robotico	36,74 rad/s
v2	00:12:12	00:12:18	00:00:06	90	180	real	261,78 rad/s
	00:12:12	00:12:28	00:00:16	90	135	robotico	49,08 rad/s
v3	00:13:13	00:15:11	00:01:58	90	180	real	1,48 rad/s
	00:13:13	00:15:15	00:02:02	90	210	robotico	1,04 rad/s

Las velocidades angulares se calcularon utilizando los datos de los movimientos tanto del brazo real y el robótico. Una vez que se conocen el tiempo que tarda en recorrer una distancia angular del punto A al punto B fácilmente puede ser obtenida la velocidad angular con la siguiente ecuación:

$$\omega = \frac{\theta}{t}, \quad (1)$$

Donde, ω es la velocidad angular, θ es la distancia angular recorrida y t es el tiempo.

De igual manera, es posible calcular la velocidad del brazo robótico al tratar de imitar el movimiento humano. Es posible también determinar el tiempo de reacción que tiene el sistema al tratar de realizar los movimientos de imitación.

3. Análisis de resultados

Al realizar las pruebas experimentales se obtuvieron los siguientes resultados. En la Figura 7 se muestra la posición de la persona, la interfaz y registro de posiciones y el prototipo del brazo en ese orden de izquierda a derecha.



Figura 7 Sistema empleado.

Se anexa la comparación de posiciones (grados) entre: el brazo real y el prototipo del brazo con su respectiva gráfica en la Figura 8. Se puede ver que existe una diferencia de 5° en la posición mínima para el brazo y 4° para el codo que tendremos que considerar para una posterior calibración en la determinación de la exactitud. En la Tabla 3 se muestran las comparaciones entre los ángulos obtenidos tanto del brazo robótico como el brazo de la persona. Respecto a los grados de libertad que tiene el hombro humano, son muy parecidas.

Tabla 3 Tabla de comparación de las posiciones entre el brazo del prototipo y el real.

Brazo Real			
Codo		Hombro	
Mínima	Máxima	Mínima	Máxima
30°	176°	0°	180°
Prototipo Brazo			
35°	180°	0°	180°

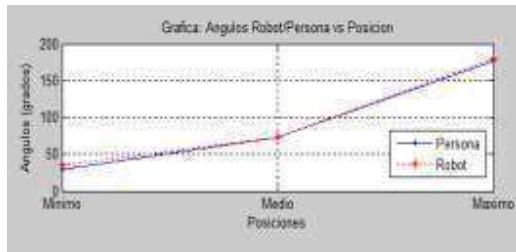


Figura 8 Gráfica de comparación de ángulos

Se muestra la primera velocidad tomada de ambos brazos y el tiempo que tardan ambos en realizar el movimiento, de un ángulo de 90° a 180°. En la figura 9 se muestra de manera gráfica claramente la diferencia entre velocidades, y los datos se muestran en la Tabla 4.

Tabla 4 Datos de la velocidad 1

Tiempo		105	2
Brazo real		8 ms	002 ms
Brazo robótico		1.9 rad/s	0
inicia en 90 grados y termina en 180 (ambos)			

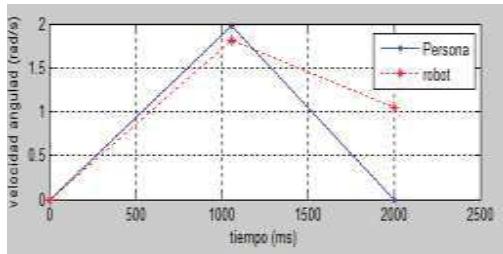


Figura 9 Comparación velocidad 1 (velocidad lenta).

En la comparación de la velocidad 1, ambas velocidades son muy parecidas ya que el brazo de la persona se mueve lentamente en un tiempo de aproximadamente 2 segundos y cuando el brazo de la persona ya llegó al ángulo de 180 el brazo robótico sigue moviéndose es por esto que se aprecian dos velocidades del brazo robótico, una en la posición final del brazo de la persona y la otra en la posición final del brazo robótico. A continuación se muestra la Tabla 5 con los datos de la velocidad 2, con su respectiva grafica (Figura 10).

Tabla 5 Datos de la velocidad 2

Tiempo		19 ms	57 ms
Brazo real		82.67 rad/s	0
Brazo robótico		73.48 rad/s	36.74 rad/s
inicia en 90 grados y termina en 180 (210 brazo robótico)			

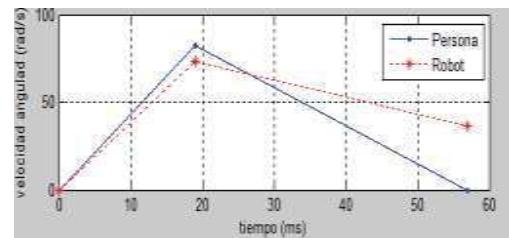


Figura 10 Comparación velocidad 2 (velocidad media).

Como se puede observar, al aumentar la velocidad, la diferencia de reacción se hace mayor. Ambas velocidades se van dispersando un poco más. Por último, la Tabla 6 y la Figura 11 es cuando la diferencia de velocidad de reacción aumenta en gran magnitud.

Tabla 6 Datos de la velocidad 3

Tiempo		6 ms	16 ms
Brazo real		261.78 rad/s	0
Brazo robótico		0 rad/s	49.08 rad/s
inicia en 90 grados y termina en 180 (135 el brazo robótico)			

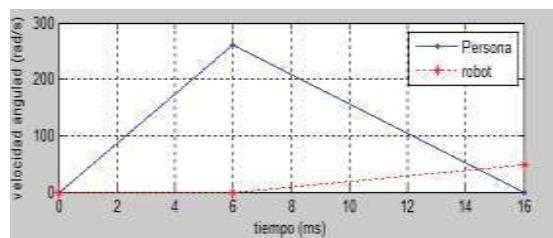


Figura 11 Gráfica de comparación velocidad 3.

El brazo robótico no reaccionó de manera adecuada, ya que mientras el brazo de la persona completa su recorrido hasta el final y regresa a su posición inicial en 6 milisegundos, el brazo robótico solo alcanza el 64% del rango máximo de posición en el mismo lapso de tiempo y regresa a su posición inicial.

4. Conclusiones

De acuerdo a las pruebas experimentales que se realizaron se concluyó que el tiempo de respuesta de un prototipo de brazo robótico contra la velocidad promedio de un brazo real, no será la misma, aunque el tiempo de respuesta sea mínimo, no es posible que sea cero.

Además de que al aumentar la velocidad del brazo real, el tiempo de respuesta del brazo robótico es menor y tiende a omitir algunas acciones y esto se debe a que el tiempo en enviar los datos entre las interfaces es lo grande como para no reaccionar al instante. Como nueva herramienta el sensor Kinect en la realidad virtual es presentado. Es necesario realizar pruebas exhaustivas tanto en su caracterización como en la precisión del sistema. También se presenta las herramientas de Energía y Processing para la obtención de datos provenientes del Kinect.

5. Bibliografía

[1] Groover M. “*Industrial robotics*”, Ed. Mc. Graw Hill, 1986.

[2] Talley, T.J., Using a state-wide high school robotics competition as the basis for a capstone design experience in electrical engineering, Frontiers in Education Conference, 1997, 775 - 777

[3] Enrique Ruiz-Velasco Sánchez, “Educatrónica: innovación en el aprendizaje de las ciencias y la tecnología”, Ediciones Díaz de Santos, 2007.

[4] Aníbal Ollero Baturone, “Robótica: manipuladores y robots móviles”, Marcombo, 2001.

[5] Pierre López, Jean-Numa Foulc, “Introducción a la robótica”, Arcadia, 1987.

[6] David González Calleja, “MOTORES CFGM”, Editorial Paraninfo, 2011.

[7] Juan Boned Tur, “Estudio de la cinemática directa e inversa de un robot móvil para el seguimiento de trayectorias arbitrarias”, Universitat de Barcelona. Departament d'Electrònica, 2009.

Implementación de Movimientos Complejos en Modelo Mecatrónico de Pruebas de Rodilla

Rangel D.¹, Suárez C.^{1,2}, Chagolla H.², Olvera A.¹ y Estévez M.¹

1.-Centro de Física Aplicada y Tecnología Avanzada, UNAM. Boulevard Juriquilla No. 3001, Col. Juriquilla.
C. P. 76230. Querétaro, Qro., México
ranged@sata.unam.mx

2.-División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Querétaro. Av. Tecnológico Esq.
M. Escobedo. S/N. C. P. 76000. Querétaro, Qro., México

Resumen

Se presenta la implementación de movimientos complejos similares a los hechos por la articulación natural de rodilla en un modelo mecatrónico de pruebas de meniscos sintéticos. Este sistema es capaz de controlar 5 moto-reductores de DC con "encoder" a través de una interfaz gráfica realizada en LabVIEW®, con el objetivo de controlar los grados de libertad que generan movimientos complejos combinados en tiempo real. El modelo fue diseñado en base a microcontroladores de la serie PIC18F de Microchip, el cual permite comunicación USB "Full-Speed", logrando velocidades de transferencia de datos de hasta 12Mbps, velocidad suficiente para esta aplicación. Se implementaron funciones de velocidad para la articulación tibia-fémur con señales de control tipo PWM para tener aceleraciones del modo exponencial y logarítmico. Este tipo de funciones simulan la aceleración que provee la articulación de la rodilla al iniciar la marcha al caminar o el frenado después de una carrera. De acuerdo a los resultados obtenidos es posible implementar éstos y otros movimientos complejos en este modelo mecatrónico debido al acoplamiento biomecánico y cinemático de los elementos tibia-fémur, con funciones no lineales que simulen los movimientos naturales que puede generar la articulación de la rodilla y analizar los esfuerzos a los que están sometidos los meniscos sintéticos.

Palabras clave: Modelo, Mecatrónico, Control, PWM, Movimientos, Prototipo, Rodilla.

1. Introducción

Se ha observado que el desgaste natural de los cartílagos y la frecuencia en la lesión de éstos, provoca que ellos con frecuencia tengan que ser sustituidos para que el paciente recupere la movilidad natural y sin dolor. Por ello es necesario generar técnicas para la elaboración de dichos cartílagos de manera artificial [1].

Los meniscos son de los cartílagos que se desgastan con más frecuencia debido a que soportan constantemente los impactos de actividades rutinarias como lo es caminar, correr, saltar, entre otras cosas [2].

La generación de un sustituto viable de menisco es dividido en cuatro principales etapas:

- Selección del material basado en las características del menisco humano.
- Síntesis y moldeado del material.
- Pruebas de biocompatibilidad en ratas.
- Pruebas biomecánicas.

Es necesario que el material sea biocompatible con el organismo, que posea flexibilidad y dureza semejantes al tejido del menisco y sea resistente al desgaste mecánico y químico.

La etapa de síntesis y moldeado es de vital importancia pues es en esta fase en la que se obtiene la porosidad necesaria que proporciona las diferentes propiedades mecánicas y permite la vascularización. Por ello se está utilizando la técnica de electro-spinning para controlar la porosidad del material.

En la etapa de pruebas mecánicas se busca caracterizar las propiedades del material mediante la construcción e instrumentación de un mecanismo que emule las condiciones de esfuerzo que presenta la rodilla. Este mecanismo debe permitir simular la compresión y el desgaste del menisco por fricción de manera controlada [3], así como simular las diferentes posiciones que adopta éste cuando es sometido a sobresfuerzos. La implementación de

movimientos complejos en este modelo mecatrónico para prueba de meniscos sintéticos es de vital importancia para el análisis, prueba y adecuación de éstos nuevos materiales de reemplazo, antes de ser implantados en la rodilla, con el propósito de recuperar la movilidad en la articulación.

2. Materiales y Métodos

Se diseñó y construyó un prototipo biomecatrónico automatizado donde se recaban datos de los esfuerzos a los que está sometido el nuevo material diseñado y con ello mejorar sus características. Para lograr este propósito se desarrolló un prototipo biomecatrónico de la siguiente manera:

Se estudió la cinemática y dinámica de la articulación de la rodilla para identificar los grados de libertad que reproducieran la movilidad compleja de la misma. Se identificó cada grado de libertad por cada plano de movimiento ya sea de rotación o translación, ya sea de movimiento lineal o rotacional. Se analizó la conjugación de diferentes movimientos para llevar a cabo rutinas de flexión, compresión, marcha, salto etc. Se seleccionaron los elementos mecatrónicos que emulan esa actividad, tales como motores de CD, servomotores o motores a pasos, transmisiones con poleas o cadenas, elementos de tracción, acoplamientos con engranes y elementos de amortiguamiento durante la compresión, analizando al mismo tiempo el comportamiento no lineal de los materiales que componen los huesos, músculos y cartílagos en una articulación natural, debido a su densidad, porosidad y modulo elástico. Se seleccionaron los elementos electromecánicos que constituyen el prototipo mecatrónico, así como los materiales de los soportes mecánicos, flechas de acoplamiento, huesos de prueba con las características similares a los huesos naturales en cuanto a propiedades mecánicas adecuadas etc.

También se diseñaron y construyeron los sistemas electrónicos de control y de potencia para mover de manera controlada y precisa cada movimiento de la articulación. Se aplicaron diferentes algoritmos de control de posición, velocidad y aceleración para verificar y adecuar el rendimiento que tiene cada uno de éstos en el movimiento articulado del prototipo, además, como en este trabajo, se implementaron movimientos complejos que realiza la rodilla natural. Se realizó la programación de los dispositivos electrónicos de control de posición y velocidad de los elementos móviles en la articulación junto con la programación de los microcontroladores, para conformar arreglos programados maestro-esclavo.

Se hizo la programación virtual del prototipo para ser manejado desde una PC. Se utilizó programación gráfica para animar el prototipo y que el usuario

pueda controlar la secuencia de movimientos rutinarios o transitorios.

Inicialmente las pruebas se realizaron con motores que ya tienen una reducción de engranes y un "encoder" acoplado al eje. En conjunto se diseñó y construyó un prototipo que simula las características de la articulación de rodilla como se muestra en la Fig. 1.

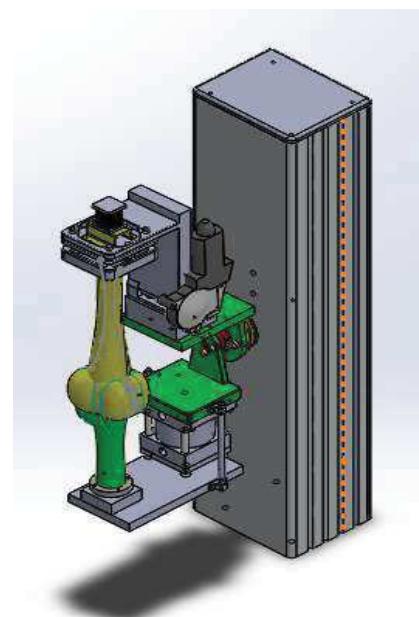


Fig. 1 Prototipo de rodilla para pruebas de meniscos sintéticos con movimientos complejos.

Para poder controlar el sentido de giro de los motorreductores, se configuró un puente H basado en transistor MOSFET para cada motor, el cual se encarga de la etapa de potencia, dado que éstos llegan a consumir hasta 10A. Si se alimenta el puente H con 24V, estará llegando aproximadamente el mismo voltaje al moto-reductor. El puente H realizado es el que se muestra en la Fig. 2, donde se puede apreciar la vista superior e inferior del circuito.



Fig. 2 Vista superior e inferior del circuito del puente H con MOSFET.

Pasando a la parte de comunicación con la PC, se optó por trabajar con el microcontrolador PIC18F4550 de 40 pines como pic maestro, y los pic 18F4431 como esclavos, ya que se requería manejar comunicación USB, controlar 5 puentes H, leer 5 "encoder" y generar 5 señales PWM; lo que resulta en un total de 10 entradas y 15 salidas digitales.

Los microcontroladores se programaron con el software “PIC C Compiler”, siguiendo una estructura como muestra el diagrama de flujo general de la Fig. 3 para el PIC 1 y PIC 2.

En este diagrama se tiene por un lado en el PIC1 la lectura de los “encoders” y el envío de los datos por USB así como mandar datos provenientes de la PC al PIC 2, mientras que en el PIC 2 se encarga de recibir estos valores y procesar aquellos que se utilizan para las salidas de los puentes H, controlando velocidad y sentido de giro de los motores.

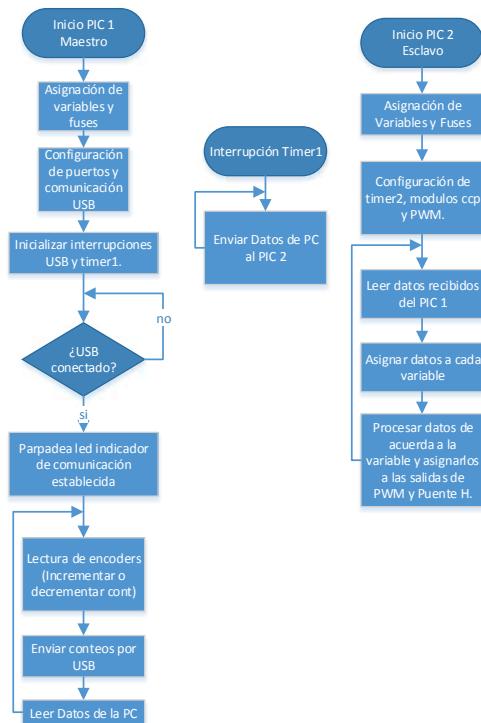


Fig. 3 Diagrama de flujo general que muestra los procesos principales del programa desarrollado.

Se realizó una interfaz gráfica en LabView, en la que se puede seleccionar el dispositivo de comunicación USB (VISA resource name). Se utilizan controles numéricos con los cuales se puede establecer la posición de giro a la que se quiere llegar en cada motor. La interfaz cuenta con un controlador PID, que fue sintonizado a través de matlab para aproximar las ganancias del controlador y así obtener parámetros de control adecuados para el sistema [4].

El controlador consta de un SubVI que consiste en una estructura de función adicional al VI de PID que nos proporciona LabView, la cual permite controlar la salida de los Puentes H de acuerdo al comportamiento de la salida de control del PID, Fig. 4.

Los parámetros de control son establecidos de manera constante para que no sean alterados durante el funcionamiento del sistema.

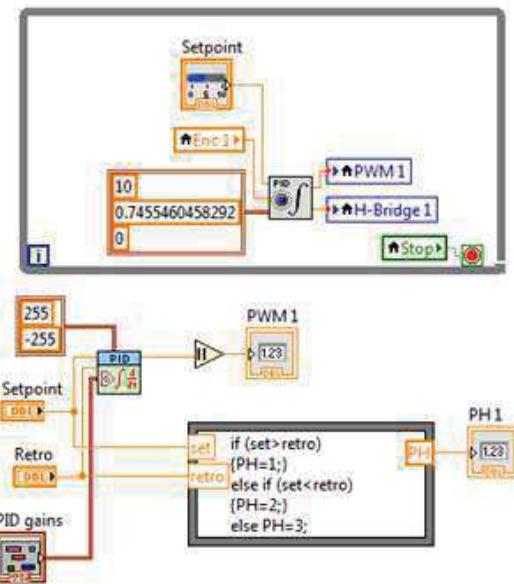


Fig. 4 SubVI desarrollado con su esquemático correspondiente.

Estos parámetros, “PID gains”, son calculados a través de la autosintonía de matlab por medio de simulink, donde se establece un control de lazo cerrado con la función de transferencia del sistema, Fig. 5.



Fig. 5 Bloque de control PID en Matlab.

Por medio del bloque de *PID Controller*, se puede manipular la curva de respuesta que tendrá el sistema con solo manejar los controles que proporciona la herramienta de sintonía de matlab, “*PID Tuner*”, Fig. 6, recuadro rojo.

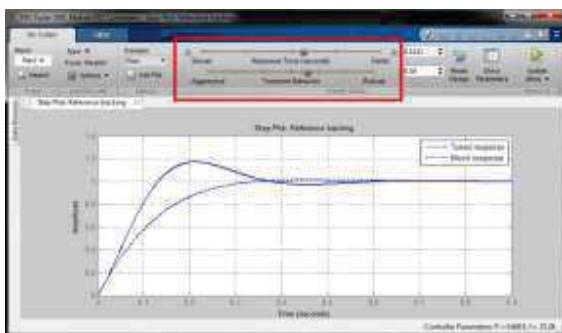


Fig. 6 Ventana de auto sintonía para el PID Controller.

Con esto obtenemos las ganancias del controlador necesarias para el bloque de PID en LabView, una vez ajustada la curva de la manera deseada.

Para la parte de pruebas del movimiento angular de los motores, se diseñó una interfaz gráfica en la que se puede obtener la curva de respuesta del motor, con la finalidad de obtener datos representativos, como el tiempo de asentamiento, el tiempo de levantamiento y el error producido.

También se realizaron pruebas en las que se aprecia la respuesta del motor a señales de entrada no lineales como lo son la función exponencial y logarítmica, y la posición alcanzada respecto a la misma, que consisten en variar el ancho de pulso de PWM durante un segundo, desde el mínimo ciclo de trabajo hasta alcanzar el máximo ancho de pulso respecto a cada función, como las mencionadas, las cuales son dependientes del tiempo [5].

3. Resultados y discusión

Los movimientos con funciones complejas se efectuaron variando el pulso de la señal PWM aplicada a los motores de control de la articulación fémur-tibia durante 1 segundo, al transcurrir el segundo la señal deja de aplicarse para apreciar el paro del motor contra su propia inercia, para ello se capturaron los datos de manera gráfica realizando varias pruebas y promediándolas; con el objeto de tener una curva de respuesta típica de los movimientos.

En la Fig. 7 vemos las señales aplicadas a la tibia, que corresponden a una función no lineal. Esta función corresponde a una variación del ancho de pulso tipo exponencial.

En el caso de la función exponencial se pretende que la energía entregada a cada motor, que mueve la tibia y el fémur, tenga un inicio débil y conforme transcurre el tiempo se incremente de manera repentina, como cuando se inicia el proceso de

marcha al caminar, siendo esto un transitorio mecánico para la articulación. En la función logarítmica inicia con gran energía entregada al motor y disminuye de manera drástica al transcurrir el tiempo. Esto puede verse como un freno mecánico al detenerse una persona cuando desacelera intempestivamente.



Fig. 7 Variación de ancho de pulso respecto a una función no lineal tipo exponencial.

La progresión de las funciones no lineales en el tiempo al ocurrir en 1 segundo (Fig. 8), son capaces de producir movimiento complejos en la articulación de la rodilla.

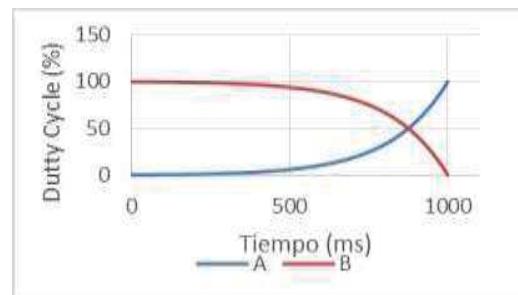


Fig. 8 Señales de aceleración aplicadas a cada motor en la articulación tibia-fémur.

Estas funciones complejas permiten conocer la cinemática y dinámica del modelo en situaciones de movimientos transitorios. Se pretende tener así mismo señales que simulen movimientos como el inicio de marcha al caminar, el frenado después de una aceleración etc., que vienen a complementar los movimientos rutinarios en un modelo mecatrónico de rodilla y que servirá para poner las condiciones críticas de esfuerzos a los meniscos sintéticos que serán probados en este modelo de rodilla.

En las Figs. 9-12, se presentan los casos típicos de movimientos complejos aplicados tanto al elemento de la tibia como al del fémur. En la aplicación de la función exponencial (Figs. 9 y 11) se observó un retardo debido a la poca energía para vencer la inercia del motor y de inmediato se acelera y desacelera. En la aplicación de la función logarítmica (Figs. 10 y 12)

se observa un incremento casi lineal, debido a que la energía entregada en cada instante da velocidad muy constante a los elementos mecánicos de la articulación, lo cual permite tener incrementos de posición muy controlados.

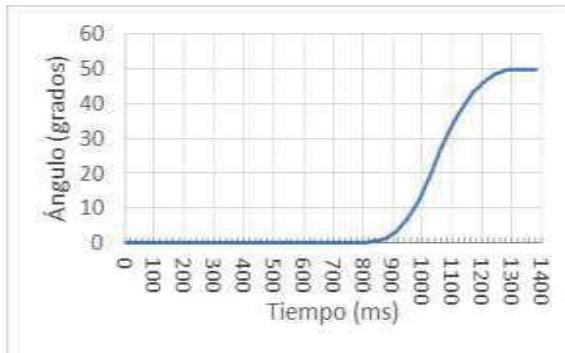


Fig. 9 Posicionamiento angular de la tibia con señal A.

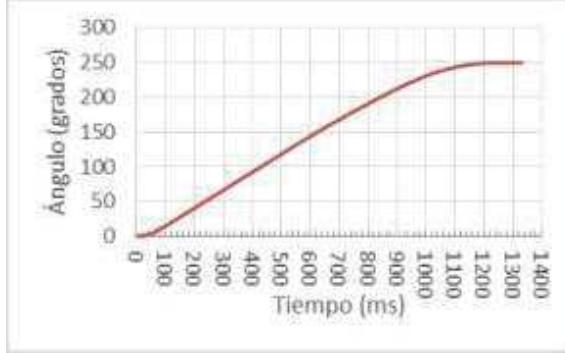


Fig. 10 Posicionamiento angular de la tibia con señal B.

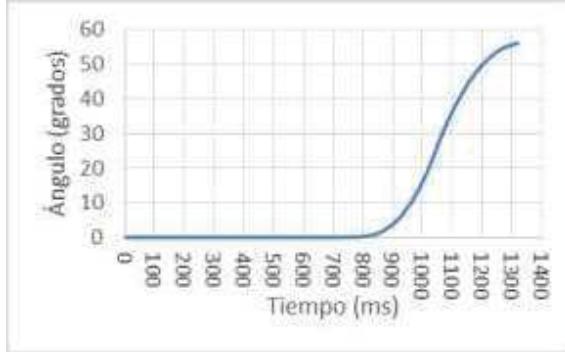


Fig. 11 Posicionamiento angular del Fémur con señal A.

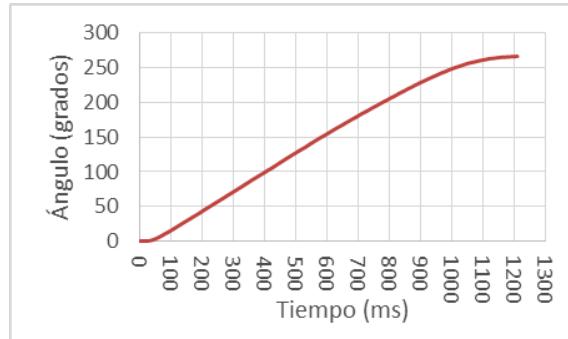


Fig. 12 Posicionamiento angular de Fémur con señal B.

Como podemos observar el tipo de señal aplicada varía el posicionamiento final, esto se debe principalmente a la relación que hay por el sistema de transmisión de cadena. Ya que al ser iguales los motores y con la misma reducción, la relación aplicada al sistema de cadenas es diferente, pues se requiere ejercer torques de diferentes magnitudes en las articulaciones. Esto es debido a que una rodilla responde de manera adaptable a los esfuerzos y movimientos realizados. Es por eso que también se aplicaron funciones no lineales en el modelo experimental para ver cómo responde la articulación y se adapta al movimiento complejo. También se puede observar que aunque se haya aplicado una señal de entrada durante 1 segundo, la señal de salida (posicionamiento angular) no termina exactamente al transcurrir ese segundo debido a la inercia generada por el motor y las piezas ensambladas, pero como se puede ver, el sistema se para en 300ms aproximadamente, por lo que el paro del movimiento puede considerarse como instantáneo.

Por otro lado, considerando que la velocidad y aceleración son diferentes en cada prueba, podemos determinar que es posible cambiar el tipo de aceleración deseada siempre y cuando salgamos de una zona muerta de trabajo del motor con respecto del PWM, que corresponde aproximadamente a un 30% del ciclo de trabajo (duty cycle), el cual es un poco elevado debido a que la fuerza que tiene que ejercer el motor es mayor acoplado a la estructura que con el eje del rotor libre o sin carga, como se realizó en pruebas anteriores sin carga y con carga. Por otro lado vemos que la velocidad del motor es adecuada, ya que las articulaciones alcanzan sin problema un ángulo de 180° en menos de 1 segundo, lo cual será útil para realizar pruebas aceleradas con cientos de repeticiones de determinado movimiento.

4. Conclusiones

El uso de algoritmos sencillos para el control de velocidad y posición en motores es un tema que ha adquirido gran importancia a partir de la automatización de sistemas mecatrónicos.

Aplicar un esquema de control con auto sintonía, desde un software como LabView es de gran utilidad, ya que permite tener una idea experimental muy aproximada de la respuesta de un sistema mecánico sin carga, el cual, al montar los motores y otros elementos en su estructura responderá de manera semejante ante las fuerzas ejercidas en conjunto con la estructura.

El hacer implementaciones experimentales en un modelo que emula las características biomecánicas de un articulación como lo es la rodilla humana es muy importante, ya que este mecanismo permiten llevar a cabo pruebas de los esfuerzos a los que estarán sometidos los elementos constitutivos de la propia rodilla como lo son los huesos, cartílagos, meniscos etc., y pueden analizarse la articulación tanto en movimientos complejos, cuya duración puede ser transitoria, como aquellos movimientos rutinarios, por ejemplo de marcha, en un régimen permanente. Este tipo de aplicaciones auguran grandes avances en otros campos, como lo es la biomecánica. En este caso la implementación del control de motores en un sistema biomecatrónico de rodilla, para la prueba de meniscos sintéticos con movimientos complejos, permite conocer y analizar la dinámica real en un modelo de pruebas que debe corresponder fehacientemente a la naturaleza de esta articulación humana.

human cadaveric knees, Arch. Orthop. Trauma Surg. 124, 575-584, 2004.

[3] McCann L., Ingham E., Jin Z. and Fisher J., *Influence of the meniscus on friction and degradation of cartilage in the natural knee joint*, Osteoarthritis and Cartilage, 17, 995-1000, 2009.

[4] Kuo B. C., “*Sistemas de Control Automático*”, Prentice Hall, Edo. de México, 7^a edición, 1996.

[5] Guess M., Maletsky P., Computational modelling of a total knee prosthetic loaded in a dynamic knee simulator, Med. Eng. & Phys. 27, 357-367, 2005.

Agradecimientos

Los autores agradecen al Consejo de Ciencia y Tecnología del Estado de Querétaro (CONCYTEQ) por el financiamiento y apoyos otorgados a este proyecto.

Referencias

- [1] JiaHsuan L., Müller O., Wünschel M., Bauer S., Wülker N., *Forces in anterior cruciate ligament during simulated weight-bearing flexion with anterior and internal rotational tibial load*, J. Biomech., 41, 1855-1861, 2008.
- [2] Agneskirchner J.D., Hurschler C., Stukenborg-Colsman C., Imhoff A.B., Lobenhoffer P., *Effect of high tibial flexion osteotomy on cartilage pressure and joint kinematics: a biomechanical study in*

Modelado, Simulación y Control PID de un Evaporador con Refrigerante HFC-134a en Matlab

Aguilar Anastacio José Carlos A, Martín Salazar Perea y Pedro Romano Apórtela

Tecnológico de Estudios Superiores de Ecatepec, División de Ingeniería Mecatrónica e Industrial.
Av. Tecnológico S/n Esq. Av. Carlos Hank González. Col. Valle de Anáhuac, Ecatepec de Morelos.
Estado de México, C.P. 55210. Tel. 50002330, mail: jcaaa72@yahoo.com.mx,

Resumen

En este artículo se presenta el modelo matemático de un intercambiador de calor que se utiliza en los sistemas de refrigeración por compresión mecánica. Se plantea el modelo dinámico del evaporador considerando las propiedades termofísicas del refrigerante, tales como, la calidad del vapor, temperatura frigorífica, el coeficiente global de transferencia de calor, etc.

Una vez obtenido el modelo matemático, se agregan controladores como el PD, PI y PID, con la finalidad de establecer el control adecuado de la temperatura en un tiempo reducido. Se utiliza Simulink de Matlab para simular el comportamiento del intercambiador de calor con los diferentes controladores, los cuales influyen directamente en el comportamiento del sistema y se pueda lograr un control adecuado para el ahorro de energía.

Palabras clave: Intercambiador de calor (evaporador), refrigerante HFC-134a, propiedades termofísicas, calidad del refrigerante, simulación.

1. Introducción

Actualmente, el gasto de energía resulta ser prioridad tanto para las amas de casa, como para las industrias que utilicen sistemas de refrigeración y más aún cuando dichos sistemas se encuentran operando las 24hrs del día, los 365 días del año. A un así agregando el mal funcionamiento del sistema, filtraciones, puertas abiertas, fugas, etc. Todo esto repercute en el alto costo económico por concepto de consumo de energía eléctrica, que irremediablemente todos debemos de pagar tanto las industrial como los hogares. Disminuir el consumo de energía, tan solo en los sistemas de refrigeración y aire acondicionado es un reto, que todavía no se ha logrado completamente.

Actualmente las investigaciones en relación a los sistemas de refrigeración, están enfocadas en la búsqueda de nuevas sustancias refrigerantes biodegradables que no atenten con la sustentabilidad del medio ambiente. Por otro lado las investigaciones en cuestión de la eficiencia del sistema eléctrico o el ahorro de energía en función de las variables termofísicas del propio sistema de refrigeración siguen siendo temas de investigación, con lo que se buscan otras alternativas, para reducir el consumo de energía eléctrica en los sistemas de refrigeración durante su funcionamiento.

La contribución de este trabajo, es mostrar la obtención del modelo matemático a partir de la ecuación de la energía, con el uso de las variables termofísicas del sistema de enfriamiento de un refrigerador, tales como la calidad del refrigerante, la temperatura a la cual se requiere mantener la cámara fría (T_{CF}), el coeficiente global de transferencia de calor, el flujo máscio del refrigerante, etc. Para este análisis se contempla el refrigerante HFC-134a, por ser un refrigerante comercial y mayormente utilizado en los sistemas de refrigeración de uso doméstico.

Obtenido el modelo del evaporador con el refrigerante HFC-134a, se procede a realizar simulaciones de control con Simulink de Matlab, para analizar el comportamiento del sistema en las diferentes condiciones de control como son proporcional (P), proporcional-integral (PI), proporcional-derivativo (PD) y proporcional-integral-derivativo (PID). Aplicando como señal de prueba un escalón, las conclusiones que se formulan después de las pruebas efectuadas son interesantes, al momento de realizar el PID, ya que el sistema responde adecuadamente en el factor de amortiguamiento y en el tiempo de asentamiento aceptable. Este tipo de resultados nos ayudan a obtener otras alternativas de control hacia los sistemas de refrigeración y aire acondicionado, dado que el control PID tiene múltiples ventajas y una de

ellas es la reducción del consumo de energía, en comparación con el control ON-OFF, dado que es el control mayormente utilizado por dichos sistemas.

En trabajos futuros se pretende realizar un prototipo para demostrar físicamente el control en función de las variables termofísica.

2. Sistema de Refrigeración

El funcionamiento de la gran mayoría de los sistemas de refrigeración por compresión mecánica, es por control On-Off, es decir; se ajusta a cierta temperatura en la cámara fría, donde internamente se encuentra el intercambiador de calor, el ajuste se realiza por medio del termostato del sistema, mientras se mantenga por debajo de la temperatura de referencia, éste permanecerá apagado, pero en cuanto sobrepasa la temperatura establecida en él termostato, el sistema entrara en funcionamiento provocando que disminuya gradualmente la temperatura de la cámara fría, y al momento de cruzar el valor de referencia dado por el termostato el sistema se apagará. Es así como la gran mayoría de los sistemas de refrigeración funcionan en la actualidad, en la Figura 1, se observa los componentes básicos que conforman el sistema de refrigeración.

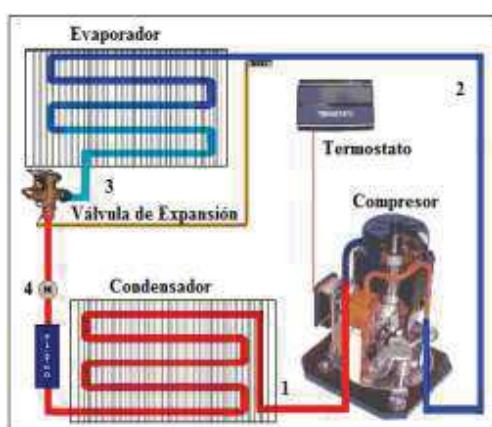


Fig. 1. Sistema de Refrigeración

Estos son: compresor, condensador, válvula de expansión, termostato y evaporador. Este último es el que se plantea para obtener su modelo matemático. Para esto, es necesario considerar las dimensiones del evaporador, el diámetro de las tuberías del serpentín, el tipo de material con el cual es construido y el tipo de refrigerante utilizado para la transferencia de calor, que en este caso es el refrigerante HFC-134a, en la figura 2, se muestra el esquema típico de un

intercambiador de calor, fabricado con láminas de aluminio con tuberías formadas en el mismo cuerpo.

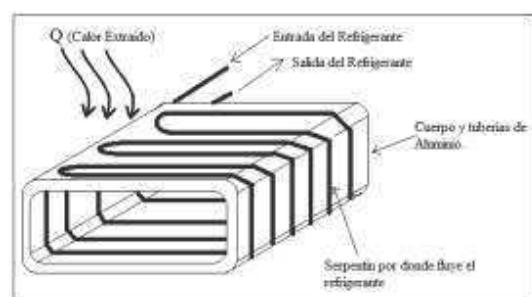


Fig. 2. Intercambiador de calor tipo evaporativo

El intercambiador de calor debe de proveer una transferencia continua y eficiente de calor desde el medio o productos que se desea extraer el calor, hacia el refrigerante que circula dentro de las tuberías, la localización del evaporador se encuentra dentro de la cámara fría del sistema de refrigeración.

2.1 Ciclo de Refrigeración por comprensión

El ciclo de compresión mecánica se lleva a cabo por cuatro procesos, estos son: compresión, condensación, expansión y evaporación.

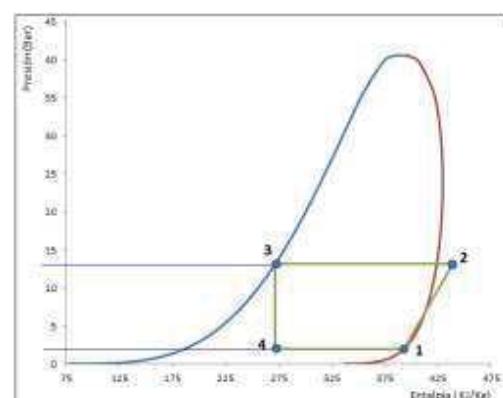


Fig. 3. Grafica Presión – Entalpía del HFC-134a

En la figura 3, se observa la gráfica Presión – Entalpía del refrigerante lo que nos indica los procesos que se llevan a cabo dentro del sistema con ciertas condiciones de trabajo, enumerados de la siguiente manera:

- 1-2 Compresión
- 2-3 Condensación
- 3-4 Estrictamente
- 4-1 Evaporación.

2.2 Análisis termodinámico

Se presenta la metodología para establecer el control de temperatura en el evaporador del sistema de refrigeración, considerando el cambio de fase de la sustancia de trabajo, es decir, considerando la variación de la calidad del vapor generada por el proceso de estrangulación al variar la temperatura de condensación. Se analiza el comportamiento del control de temperatura, al utilizar HFC-134a, Miher Sen, propone un primer modelo considerando una sola fase del fluido [2].

En la Figura 4 se muestra el diagrama T-h del ciclo de refrigeración del fluido de trabajo HFC-134a, se observa, que la calidad del refrigerante (x) cambia a medida que varía la temperatura de condensación (T_{cond}) y la temperatura ambiente (T_{amb}), ($T_{cond}=T_{amb}+10^{\circ}\text{C}$), es decir, para una temperatura de $T_{cond}=40^{\circ}\text{C}$, $x=0.334$, $T_{cond}=45^{\circ}\text{C}$, $x=0.376$ y $T_{cond}=50^{\circ}\text{C}$, $x=0.416$, manteniendo constante la temperatura de evaporación, $T_{eva}=-10^{\circ}\text{C}$.

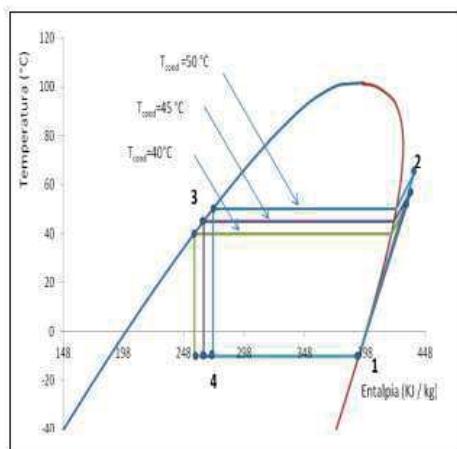


Fig.4. Grafica Temperatura – Entalpía, del HFC-134a.

Lo que representaría una variación en la operación del sistema, debido a que a mayor calidad del vapor, menos efecto refrigerante se obtendrá y además la transferencia de calor se verá reducida, afectando al control de la temperatura, mientras que por el lado de la compresión se observa que incrementa la temperatura del refrigerante sobrecalentado, al variar la temperatura de condensación.

2.3 Modelo Dinámico del Intercambiador

El modelo dinámico del intercambiador de calor tipo evaporador se obtiene a partir de un

balance de energía, así mismo, se utiliza el método de resistencia interna despreciable, donde se realiza la suposición de que la temperatura del sólido es espacialmente uniforme en cualquier instante durante el proceso transitorio, lo que implica que los gradientes de temperatura dentro del sólido son insignificantes, para tal efecto se debe evaluar el número de Biot que debe ser menor a 0.1, lo que representa que la resistencia a la conducción dentro del sólido es mucho menor que la resistencia a la convección a través de la capa límite del fluido, entonces es razonable la suposición de una distribución de temperatura uniforme y sólo depende del tiempo y no de la longitud de la pared del intercambiador de calor [1]. Miher Sen propone un primer modelo considerando una sola fase del fluido [2].

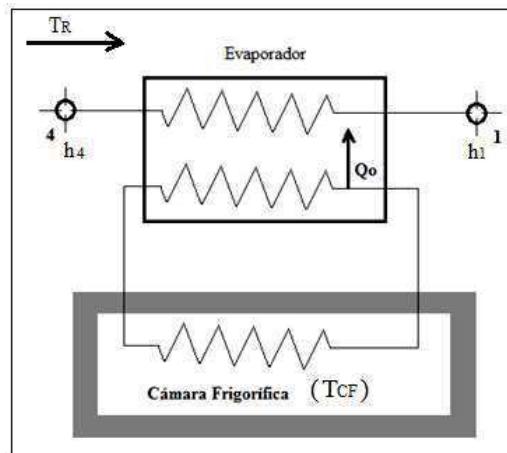


Fig.5. Diagrama esquemático del evaporador.

Se plantea el balance de energía a partir de la primera ley de la termodinámica, para el evaporador con base al diagrama de la Figura 5, donde se tiene energía entrante, almacenada y saliente en forma de calor, en el caso de la energía almacenada esta es la carga térmica que se tiene que desprender de los objetos en la cámara fría. [1]

$$\dot{E}_{\text{entrante}} + \dot{E}_{\text{generada}} - \dot{E}_{\text{saliente}} = \Delta E_{\text{almacenada}} \quad (1)$$

Al asociar con las propiedades termodinámicas, el flujo de calor, la variación de la energía depende directamente de la variación de la temperatura en el evaporador, la masa del refrigerante y la capacidad calorífica. [1]

$$\Delta E_{\text{alm}} = \frac{dE}{dt} = mC\rho v \frac{dT_R}{dt} \quad (2)$$

Para encontrar el efecto de refrigeración como una función de la temperatura se calcula el Coeficiente Global de Transferencia (U_E), a partir del el espesor de las paredes del propio evaporador, las propiedades del refrigerante y el espesor de la capa de hielo que se forma en el serpentín, considerando el área (A) de la placa transversal que forma el evaporador

$$\dot{m}h_4 + U_E A_E (T_{CF} - T_R) - \dot{m}h_i = \rho Cpv \frac{dT_R}{dt} \quad (3)$$

La T_R , representa la temperatura de operación en el evaporador que es 10°C menor a la temperatura de la cámara fría, (T_{CF}). Así mismo, la T_R se encuentra principalmente en función de la entalpía, cambio de fase y la calidad. La calidad (x_4), del refrigerante está dada por

$$x_4 = \frac{h_4 - h_{4\text{liqsat}}}{h_{fg} - h_{f4}} = \frac{h_4 - h_{4\text{liqsat}}}{h_{fg}} \quad (4)$$

Despejando la entalpía en el punto 4, de los procesos del sistema de refrigeración figura 4, se tiene:

$$h_4 = h_{4\text{liqsat}} + h_{fg}x_4 \quad (5)$$

Sustituyendo h_4 en función de la calidad y la entalpia en el cambio de fase en la ecuación (3),

$$\dot{m}[h_{4\text{líquido}} + h_{fg}x_4] - mh_i + U_E A_E (T_{CF} - T_R) = \rho Cpv \frac{dT_R}{dt} \quad (6)$$

En la ecuación (6) se contempla la entalpía del estado 4, donde se aplica la regresión lineal para un intervalo de temperatura del refrigerante (T_R), $T_R = -10^\circ\text{C} \pm 15^\circ\text{C}$, figura 6.

$$h_{fg}(T_R) = L - GT_R \quad (7)$$

Dónde: L es el origen y G la pendiente

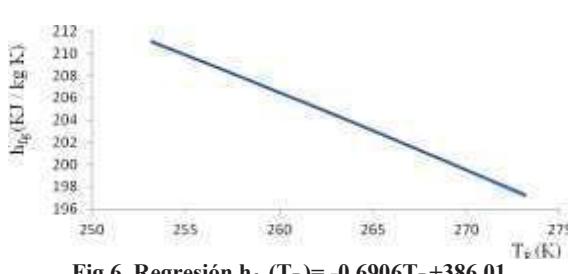


Fig.6. Regresión $h_{fg}(T_R) = -0.6906T_R + 386.01$

Entonces

$$\dot{m}(x-1)(L - G(T_R)) + U_E A_E (T_{CF} - T_R) = \rho Cpv \frac{dT_R}{dt} \quad (8)$$

De la ecuación (8), se observa que la temperatura del refrigerante depende de las variables

$$T_R = T_R(\rho, C_p, v, U_E, A_E, \dot{m}, L, G, t) \quad (9)$$

De acuerdo al análisis paramétrico con respecto a la variación de la temperatura del refrigerante, (T_R), se tiene que las variables que afectan la dinámica del sistema es la entalpía en el cambio de fase, flujo másico, el calor específico, el volumen, la densidad, la calidad, etc. sin embargo, para realizar la simulación de las leyes de control en el dominio de la frecuencia se traslada la ecuación (8) en una ecuación adimensional, es decir, en términos de θ_E que es la temperatura adimensional del evaporador y la constante térmica del tiempo (τ).

$$T_R = T_{CF} - \theta_E \Delta T_c \quad (10)$$

Proveniente del método de separación de variables, [3]

$$\theta_E = \frac{T - T_1}{T_2 - T_1} \quad (11)$$

Donde la constante térmica del tiempo es:
 [3]

$$\tau = \frac{t}{tc} \quad (12)$$

La ecuación se encuentra en términos de la constante térmica y la temperatura adimensional

$$\begin{aligned} \dot{m}(x-1)(L - G(T_{CF} - \theta_E \Delta T_c)) + U_E A_E (T_{CF} - T_{CF} + \theta_E \Delta T_c) = \\ \rho Cpv \left(\frac{dT_{CF}}{d\tau} - \frac{d\theta_E \Delta T_c}{d\tau} \right) \end{aligned} \quad (13)$$

Reduciendo términos

$$-\frac{d\theta_E}{d\tau} = \theta_E + (x-1) \left(1 - \frac{G}{L} T_{CF} \left(1 - \frac{\theta_E \Delta T_c}{T_{CF}} \right) \right) \quad (14)$$

Se propone realizar cambios de variables como α y γ , dado que se encuentran en función de L y G.

$$\alpha = \frac{G}{L} T_{CF}; \quad \gamma = \frac{\dot{m}L}{U_E A_E T_{CF}} = \frac{\Delta T_c}{T_{CF}}$$

Dónde: $G = -0.6906 \text{ KJ/kg.K}$ y $L = 386.01 \text{ KJ/KG}$, estos valores se obtiene al utilizar la correlación lineal de la entalpía de cambio de fase (h_{fg}), del HFC-134a, como se muestra en la figura 6.

Entonces el modelo dinámico del evaporador se reduce a:

$$-\frac{d\theta_E}{d\tau} = \theta_E + (x-1)(1-\alpha(1-\gamma\theta_E)) \quad (15)$$

Se obtiene la función de transferencia en el dominio de la frecuencia, utilizando el método de la transferencia de Laplace. Agrupando y definiendo los términos de la ecuación (15). Dónde: $b = (\alpha-1) (x-1)$ y $a = 1 + \alpha\gamma(x-1)$. Entonces la ecuación se reduce a:

$$b = \frac{d\theta_E}{d\tau} + a\theta_E \quad (16)$$

Aplicando la transformada de Laplace

$$bL\{1\} = \left\{ L \frac{d\theta_E}{d\tau} \right\} + aL\{\theta_E\} \quad (17)$$

Se obtiene:

$$\theta_E(s) = \frac{b}{s(s+a)} \quad (18)$$

La ecuación (18), representa la función de transferencia del intercambiador de calor tipo evaporativo. En la Tabla 1, se muestran las ecuaciones de transferencia para las diferentes temperaturas del condensador (T_{cond}), la constante $\alpha = 0.4886$, $U_E = 2.182 \text{ kW/m}^2 \text{ K}$ y $A_E = 0.15 \text{ m}^2$.

Tabla 1. Función de transferencia para las diferentes T_{cond} .

$T_{cond} = 40^\circ\text{C}$	$T_{cond} = 45^\circ\text{C}$	$T_{cond} = 50^\circ\text{C}$
$\gamma = 0.1123$	$\gamma = -0.1190$	$\gamma = -0.1266$
$x = 0.3392$	$x = 0.3761$	$x = 0.4138$
$b = 0.3379$	$b = 0.3190$	$b = 0.2997$
$a = 0.9637$	$a = 0.9637$	$a = 0.9637$
$m = 0.026 \text{ kg/s}$	$m = 0.0275 \text{ kg/s}$	$m = 0.0293 \text{ kg/s}$
$\theta_E(s) = \frac{0.3379}{s(s+0.9637)}$	$\theta_E(s) = \frac{0.3190}{s(s+0.9637)}$	$\theta_E(s) = \frac{0.2997}{s(s+0.9637)}$

La calidad del refrigerante repercute directamente en el modelo matemático, al afectar los coeficientes de la función de transferencia, el coeficiente α no depende de la temperatura por lo tanto se mantiene constante, pero γ es función del refrigerante dado que depende de las temperaturas de operación y del flujo máscio, por consiguiente son variables que influyen en el control del sistema en las temperaturas de 40, 45 y 50°C.

2.4 Control PID

El modelo matemático obtenido en función de la frecuencia, es controlable ya que se puede someter a pruebas de control en lazo cerrado con los datos

obtenidos de la tabla 1. El sistema se puede observar en la figura 7, donde el modelo matemático obtenido, ahora se considera la planta del sistema y es sometido a una señal de prueba tipo escalón

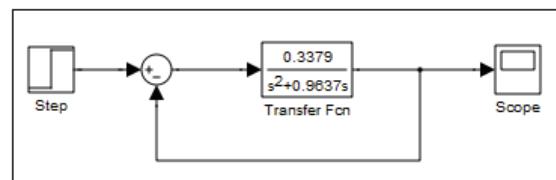


Fig. 7. Sistema con retroalimentación unitaria

La retroalimentación se considera unitaria dado que el sensor es uno a uno, este sensor puede ser de temperatura o presión. La respuesta de este sistema se puede observar en la figura 8.

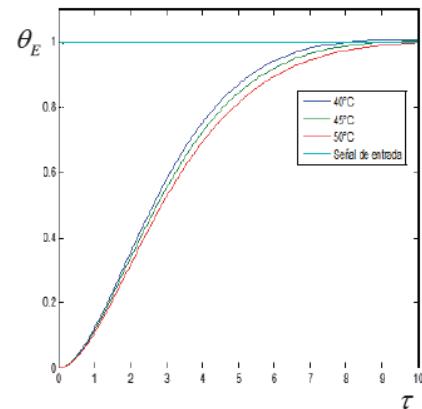


Fig. 8. Respuesta de la planta con retroalimentación unitaria.

Se aprecia la respuesta del sistema para las diferentes temperaturas de experimentación, según la tabla 1. Se nota que la temperatura tiende a la estabilidad después de un tiempo adimensional (τ) de 10, independientemente de la variación de la temperatura de operación en el condensador de 40, 45 y 50°C.

Se emplea un controlador PID por que ofrece una estabilidad en estado estacionario, y es uno de los controladores más empleados en el control industrial de ciertos sistemas. Se demuestra que se tiene una respuesta más adecuada para la planta en comparación con el sistema sin controlador PID.

Donde la ecuación 19, es el controlador PID que se agrega a la planta, para someterlo a la prueba del escalón unitario

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (19)$$

El sistema se puede observar en la figura 9.

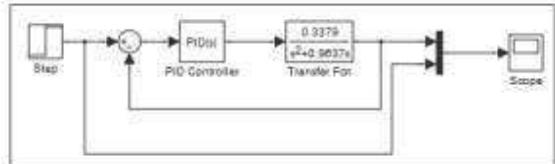


Fig. 9. Sistema del controlador PID y la planta

La planta y el controlador (G_{pc}) proporcionan la función de transferencia en lazo abierto como lo muestra la ecuación 20.

$$G_{pc}(s) = \frac{0.3379(K_p T_i s + K_p + T_d s^2)}{T_i(s^3 + 0.9637s^2)} \quad (20)$$

Por lo tanto la función de transferencia ($C(s)/R(s)$), en lazo cerrado en retroalimentación unitaria se tiene como lo muestra la ecuación 21.

$$\frac{C(s)}{R(s)} = \frac{0.3379(K_p T_i s + K_p + T_d s^2)}{T_i s^3 + (0.9637 + 0.3379 T_d) T_i s^2 + 0.3379 K_p T_i s + 0.3379 K_p} \quad (21)$$

Como se puede observar en la función de transferencia del controlador y la planta se tiene en el denominador un polinomio característico de tercer orden, donde aplicando el criterio de Routh y la sintonización de Ziegler y Nichols, se puede lograr obtener las constantes de tiempo T_i y T_d , para que al mismo tiempo se puedan calcular los valores de las constantes K_p , K_i y K_d del propio sistema.

Como prueba se sometió el controlador PID y la planta con valores de $K_p=5$, $K_i=2$ y $K_d=6.9$ para observar el comportamiento del sistema, obteniendo como respuesta la gráfica, de la figura 10, se muestra la respuesta del controlador PID, la cual es aceptable, con un tiempo de estabilidad mayor al que se obtiene sin el controlador PID. [4][5]

3. Análisis de resultados

La θ_E , al variar la temperatura del condensador no representa grandes cambios en los tiempos de estabilidad del sistema, debido a que como acción correctora se está variando el flujo mísico para

compensar la disminución del efecto refrigerante y mantener estable el sistema, si el flujo se estableciera constante en el funcionamiento del intercambiador de calor tipo evaporativo, los transitorios se incrementarían en la señal de respuesta.

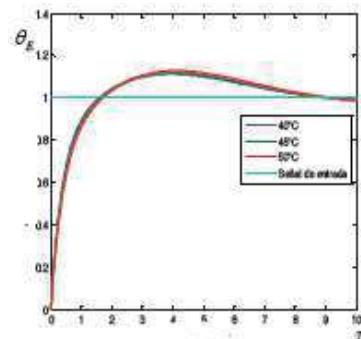


Fig. 10. Respuesta del controlador PID, con $K_p=5$, $K_i=2$ y $K_d=6.9$

4.- Conclusiones

En este trabajo se realizó una metodología matemática para establecer el control de temperatura en un intercambiador de calor considerando el cambio de fase de la sustancia de trabajo.

Así como la aplicación del controlador PID para mantener una mejora en el control del sistema, manipulando los valores de K_p , K_i y K_d en función de la respuesta de la propia planta sin el controlador.

Referencias

1. Seborg, Edgar, Mellichamp, Doyle, "Process Dynamics and Control", Wiley, p. 1-501, 2011.
2. Mihir Sen, "A review of principles and applications of thermal control" Ingeniería Mecánica Tecnología y Desarrollo, Vol 1, No. 4, p. 115-131, 2004.
3. Sorour Alotaibi, Sen, Mihir, Yang Kwang-Tzu. Thermal control of heat exchangers. ASME, 35th National Heat Transfer Conference, California, 2001.
4. Katsuhiko Ogata. Ingeniería de Control Moderna. Ed. Prentice Hall, 2003.
5. Golnaraghi Kuo. Automatic control system. Wiley, 2008.

Adquisición de Datos en LabVIEW para un Sistema de Seguimiento Solar

Martínez Jiménez José Manuel¹, Vergara Betancourt Angel²,
Moreno Castro Luis Fernando y Ramiro Ramiro José

Instituto Tecnológico Superior De Zacapoaxtla, División de Ingeniería Mecatrónica
Carretera Acuaco-Zacapoaxtla, Km 8, Col. Totoltepec, Zacapoaxtla, Pue.
Tel: (233) 31 75000. email: 1jomaji92@gmail.com, 2a.vergara@live.itsz.edu.mx

Resumen

En este proyecto, se realiza el diseño y la implementación de un sistema automatizado de adquisición de datos, para un sistema mecatrónico de seguimiento solar de 2 grados de libertad. El diseño se basa en la plataforma Arduino, cuya función es de trabajar como esclavo del programa realizado en LabVIEW, lo que permite monitorear información en tiempo real (In situ Sistema SCADA). Como resultado, se tiene un sistema que permite obtener datos de un sistema de captación de radiación solar (fotovoltaico y/o térmico), los cuales son exportados en un formato ".xls" o en ".txt" para su posterior uso y visualización.

Palabras clave: Adquisición de datos, LabVIEW, Arduino, Seguidor Solar, SCADA.

1. Introducción

La función de una adquisición de datos es, obtener información ya sea de un proceso o fenómeno físico y procesarla para una tarea posterior (visualización, control, retroalimentación, etc.). El programa de adquisición de datos tiene como meta adquirir, analizar y presentar información en la PC. Esta adquisición puede llevarse en tiempo real, para lo

cual se requieren sistemas que discretizan la información para procesamiento [1-3]. En nuestro caso, esta tarea la lleva a cabo el microcontrolador ATmega328 de la tarjeta Arduino UNO [4]. Una vez obtenida la información es procesada mediante el software de código gráfico LabVIEW, el cual se encarga de almacenar dicha información y presentarla gráficamente para su interpretación [5,6].

El interés por desarrollar este sistema de adquisición es para monitorear a lo largo del día y del año, los niveles de radiación solar captados por un sistema de captación ya sea fotovoltaico (variación de voltaje) o bien térmico (variación de temperatura) [7,8]. La información obtenida servirá por una parte para caracterizar la región Nororiental del Estado de Puebla en cuanto a la eficiencia de estos sistemas generadores de energía alternativa, y por otra parte, como señales de retroalimentación para controlar un sistema de seguimiento solar.

2. Sistema de adquisición de datos

La adquisición de datos se lleva a cabo para señales de voltaje enviadas por un sistema mecatrónico de seguimiento solar de dos grados de libertad. La señal es acondicionada y enviada a la tarjeta Arduino para su procesamiento en LabVIEW. En la Fig. 1 se observa un esquema de este proceso.



Fig. 1. Esquema del sistema de adquisición de datos.

2.1 Sistema de seguimiento solar

El sistema de seguimiento solar, consiste en un mecanismo de dos grados de libertad, controlado través la tarjeta de Arduino MEGA ADK, con un par de servomotores estándar GOTEK GS-5515MG, los cuales permiten movimientos angulares para el desplazamiento azimutal y de elevación del sistema captador de radiación. Este mecanismo se ha acondicionado con una interfaz a dispositivos móviles mediante el sistema Android, para su control inalámbrico, sin embargo este desarrollo no será punto de discusión en este artículo. El prototipo utilizado está compuesto por un grupo de tres celdas fotovoltaicas, que son las que medirán la cantidad de energía captada. Adicionalmente, se le incluyó como sensores, un conjunto de resistencias fotoresistivas distribuidas de manera uniforme, para detectar la posición de mayor intensidad lumínica y con ello, reorientar el sistema hacia esa dirección. En la figura 2. Se observa el desarrollo de este prototipo.



Fig. 2. Prototipo de seguidor solar de dos grados de libertad

2.2 Acondicionamiento de señal e interfaz con la tarjeta Arduino.

Para la lectura de las señales de voltaje enviadas por las celdas fotovoltaicas, fue necesario ajustar la señal a los rangos operacionales de la tarjeta arduino que son de 0 a 5V/DC. Para ello se implementó un sencillo sistema de acondicionamiento de señal, conformado, por un arreglo de 5 resistencias de $10\text{K}\Omega$ en serie (divisor de voltaje), que hace posible la lectura de voltaje, en una entrada analógica del arduino; además, este sistema también protege y evita que la tarjeta se dañe a causa de una corriente elevada en la señal de entrada proporcionada por la celda fotovoltaica. La tierra del divisor del voltaje y la tierra del Arduino, están unidas con la finalidad de trabajar sobre el mismo sistema. Véase la figura 3.

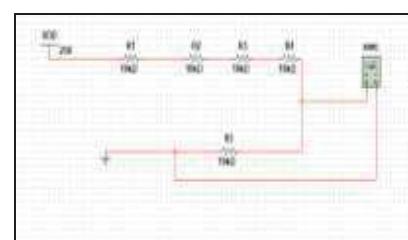


Fig.3 Divisor de voltaje.

En la figura 4 se observa el uso de la tarjeta Arduino para comunicación con la PC y con el código de programación de LabVIEW.

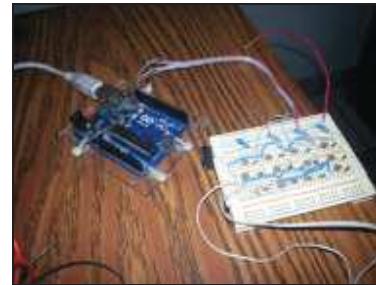


Fig.4. Acondicionamiento de señal e interfaz a PC mediante tarjeta Arduino.

2.3 Diseño de sistema de adquisición de datos en LabVIEW

El diseño del programa en LabVIEW, hace posible, la observación de cómo se comporta el sistema de captación solar en tiempo real y a lo largo del día. Además permite adquirir datos cada determinado tiempo. El programa contiene elementos que permiten graficar los datos adquiridos por el Arduino, para luego ser guardados en un archivo de formato “.xls” o en “.txt”. Véase la figura.5

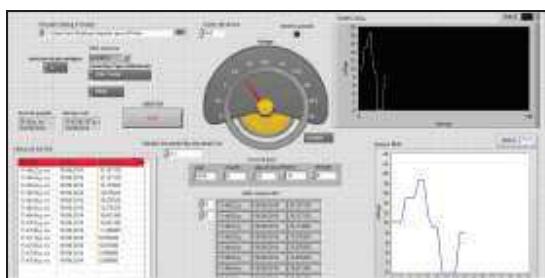


Fig. 5. Diseño del sistema de adquisición de datos (panel frontal).

El programa guarda los datos continuamente con la finalidad de preservar la información monitoreada en caso de que la PC sea apagada por accidente, o que diversos factores obliguen el cierre del programa LabVIEW. También está diseñado para para adquirir los datos durante un lapso determinado con un cierto tiempo de muestreo, y detenerse al minuto, hora, día y año que le sea programado

Éste sistema puede monitorear las señales a través del puerto USB, o bien de manera remota mediante comunicación Bluetooth con un alcance máximo de 40 metros sin interferencia, y con interferencia de 24 metros.

Es importante mencionar que la comunicación de Arduino con LabVIEW, se lleva a cabo mediante un programa llamado *LIFA_Base*, el cual se carga en el microcontrolador. Este programa agrega el toolkit de Arduino para interface con LabVIEW. Está librería abarca otras librerías que hacen más fácil la programación grafica en Labview. Su extensión es un archivo INO, que permite ser grabado en el IDE de Arduino.

La comunicación entre LabVIEW y ARDUINO se lleva a cabo de la siguiente manera. Mediante el comando *init* se define el tipo de Arduino a utilizar al igual que el número de puerto al que se conecta y el tipo de conexión que se usará. Este comando es el

que nos permite realizar la interfaz Arduino-LabVIEW. Véase la figura.6

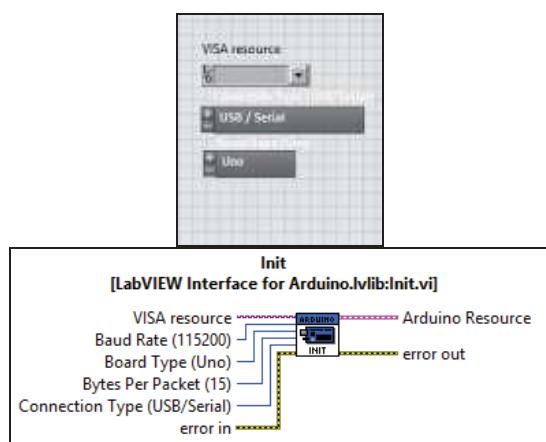


Fig.6. Interfaz gráfica en la que se muestra la comunicación con Arduino.

El segundo comando a usar es *Analog Read Pin*, nos permite leer la entrada analógica y el voltaje real, en el basta con decirle que pin usaremos de Arduino y generar un indicador de voltaje que por default contiene. Véase la figura. 7.

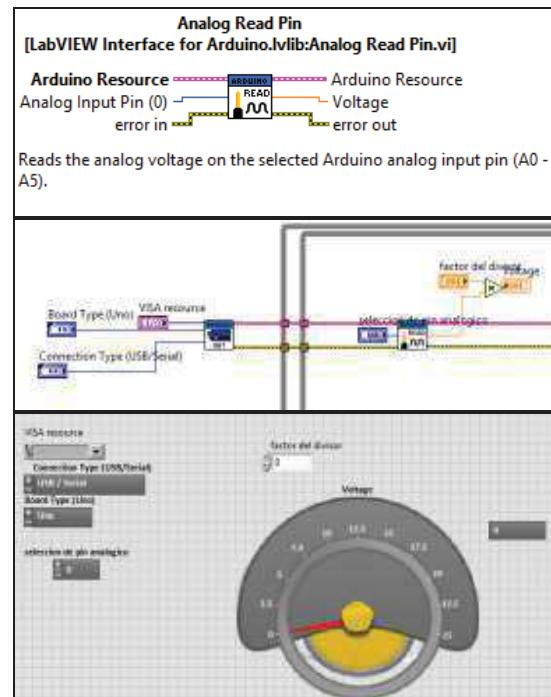


Fig.7 Componentes del lector de una entrada analógica.

La interfaz Arduino-LabVIEW se cierra con el comando *close*, el cual se coloca al final del código gráfico de LabVIEW. Con este comando indicamos que terminamos la comunicación y nos permite saber si existió un error en el sistema. (Figura 8).

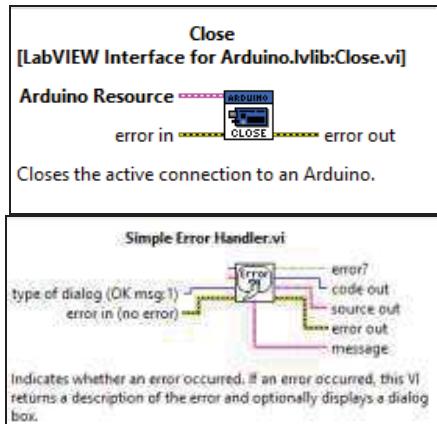


Figura.8 Finalización del programa.

Para guardar datos se usó comando *write to spreadsheet file.vi*. Con este comando se define la dirección donde se guardaran los datos obtenidos y el formato en el que se guardara el archivo. (Figura.9) El código gráfico de todo el sistema de adquisición de datos, se presenta en la figura 10.

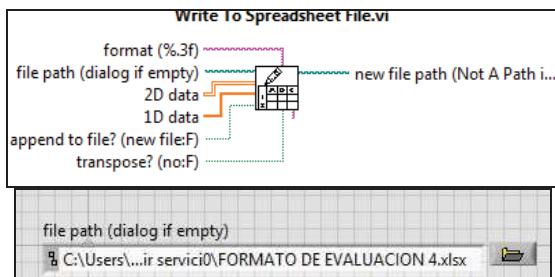


Fig. 9 Extensión y destino del archivo a guardar.

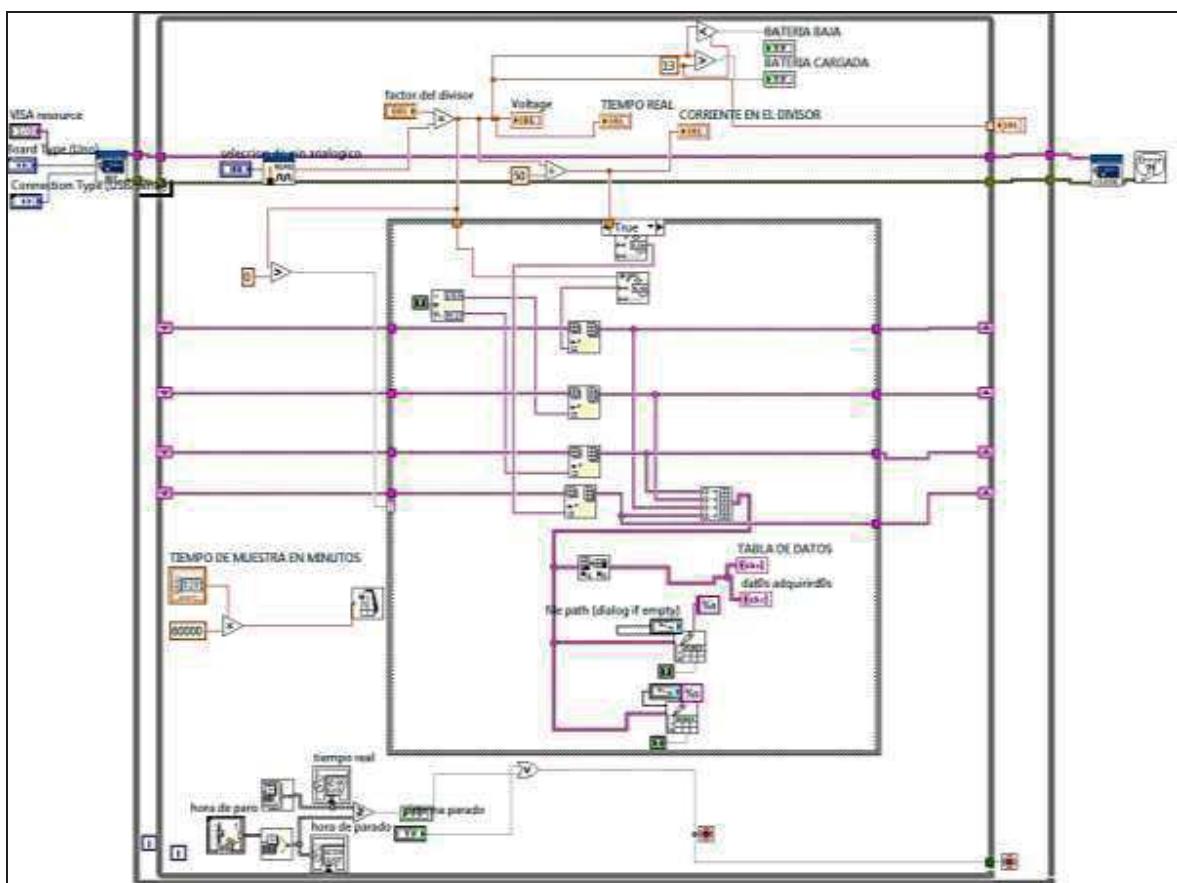


Figura.10 Código completo del programa en LabVIEW.

3. Análisis de Resultados

El sistema completo y terminado que se muestra en la figura 11, está conformado por el sistema mecatrónico de seguimiento solar de dos ejes descrito anteriormente, la interfaz Arduino-LabVIEW, el código de LabVIEW y el panel de control que funciona como medio de interacción con el usuario y medio de visualización de la información obtenida.



Fig.11. Sistema de adquisición de datos completo.

Las pruebas para verificar el correcto funcionamiento del sistema de adquisición de datos se realizaron para distintos ángulos de inclinación, como se observa en las figuras 12 y 13.

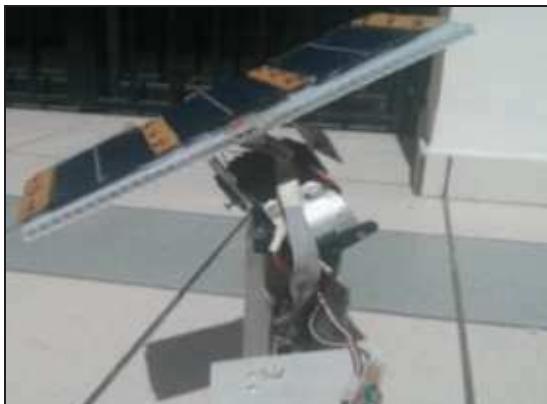


Fig.12 Sistema mecatrónico de 2 grados de libertad en posición inclinada a 45°.



Fig.13. Inclinación en 0° grados o estado de reposo.

Una vez comprobado el correcto funcionamiento del sistema, se procedió a utilizarlo con un sistema de paneles fotovoltaicos de mayores dimensiones que se hallan instalados en la institución. Los resultados que se obtuvieron demostraron el correcto funcionamiento del sistema de adquisición de datos y actualmente permiten monitorear en tiempo real los niveles de radiación captados por este panel solar. Como ejemplo de este proceso de lectura de datos, se presenta en la tabla 1 la tensión medida en función de la radiación incidente el dia 21 de mayo del 2014. Estos datos son graficados y mostrados en la figura 14.

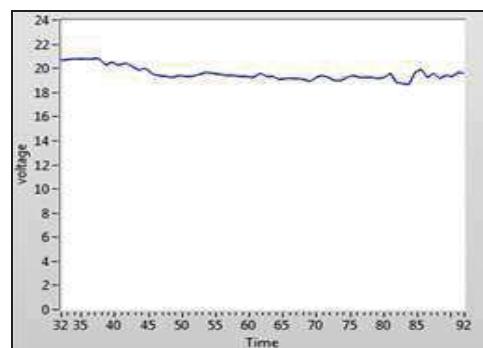


Fig.14. Gráfica de datos correspondientes a la tabla 1 que fueron obtenidos el 21 de mayo del 2014 mediante el sistema de adquisición desarrollado en LabVIEW.

VOLTAJE	FECHA	HORA
17,334240	21/05/2014	9:01
17,357760	21/05/2014	9:05
17,428120	21/05/2014	9:10
17,545920	21/05/2014	9:15
17,663520	21/05/2014	9:20
21,262080	21/05/2014	9:25
20,509440	21/05/2014	9:30
20,462400	21/05/2014	9:35
20,442400	21/05/2014	9:40
20,485920	21/05/2014	9:45
20,500440	21/05/2014	9:50
20,532060	21/05/2014	9:55
20,438180	21/05/2014	10:00
20,580000	21/05/2014	10:05
20,603520	21/05/2014	10:10
20,603520	21/05/2014	10:15
20,650560	21/05/2014	10:20
20,650560	21/05/2014	10:25
20,674080	21/05/2014	10:30
20,674080	21/05/2014	10:35
20,697600	21/05/2014	10:40
20,721120	21/05/2014	10:45
20,768160	21/05/2014	10:50
20,768160	21/05/2014	10:55
20,788160	21/05/2014	11:00
20,791580	21/05/2014	11:05
20,885760	21/05/2014	11:10
20,227200	21/05/2014	11:15
20,556480	21/05/2014	11:20
20,250720	21/05/2014	11:25
20,435360	21/05/2014	11:30
20,203680	21/05/2014	11:35
19,874400	21/05/2014	11:40
20,015520	21/05/2014	11:45
19,356960	21/05/2014	12:55
19,300920	21/05/2014	13:00
19,262880	21/05/2014	13:05
19,508040	21/05/2014	13:10
19,356090	21/05/2014	13:15
19,356060	21/05/2014	13:20
19,121760	21/05/2014	13:25
19,192320	21/05/2014	13:30
19,168800	21/05/2014	13:35
19,192320	21/05/2014	13:40
19,098240	21/05/2014	13:45
18,933000	21/05/2014	13:50
19,239360	21/05/2014	13:55
19,404000	21/05/2014	14:00
19,239360	21/05/2014	14:05

Tabla 1. Voltaje medido en función de la radiación incidente en un panel solar.

4. Conclusiones

El sistema de adquisición de datos que se ha discutido en este artículo, nos permite contar en la actualidad con una herramienta de monitoreo en tiempo real de distintas variables físicas, en particular no interesa utilizarlo para medir niveles de radiación óptica, lumínica y térmica en paneles y colectores solares.

La elaboración de este sistema ha implicado distintos desarrollos los cuales se mencionan a continuación:

1. Se logra realizar un prototipo de un sistema mecatrónico de 2 ejes, capaz de ser manipulado por medio de un dispositivo Android, vía remota de manera manual o automática.
2. Se logra realizar el programa en LabVIEW para adquirir datos del sistema mecatrónico vía remota o por conexión USB.

3. Se demostró el sistema de adquisición en LabVIEW es viable, además de ser gráfico y mostrar valores en tiempo real.

4. En contraste con un sistema de adquisición de datos convencional, este sistema permite hacer un estudio más detallado sobre la captación de energía fotovoltaica además de ser versátil y adaptable a otras variables.

En conclusión, este sistema de monitoreo nos dota de un sistema de adquisición de datos remoto y flexible que nos permitirá monitorear en tiempo real los cambios de los niveles de radiación solar captados por sistemas de captación solar fotovoltaicos y térmicos.

Referencias

- [1] Mendiburu Díaz Henry Antonio, *Instrumentación Virtual Industrial*” INDECOP- Perú, 2006.
- [2] Rodríguez Penin Aquilino, “*Sistemas SCADA*”, Marcombo, S.A 3ra. Edición, 2006.
- [3] Pallas Areny Ramón, “*Sensores y Acondicionadores de señal*”, Ed. Marcombo, 4ta. Edición.
- [4] [\[4\] <http://www.arduino.cc/>](http://www.arduino.cc/)
- [5] Holguín Londoño Germán Andrés y Pérez Londoño Sandra Milena: “*Curso Básico de Labview 6i*”, Universidad Tecnológica de Pereira, 2002.
- [6] Ruiz Gutiérrez José Manuel: “*Utilización de Labview para la Visualización y Control de la Plataforma Open Hardware Arduino*”, Creative Commons Attribution 3.0, 2012.
- [7] Pereda Soto Isidro Elvis “*Celdas Fotovoltaicas en Generación Distribuida*”, Tesis, Pontificia Universidad Autónoma de Chile, 2005
- [8] Lammardo Adriana y Baritto Miguel “*Modelo matemático del comportamiento térmico de un colector solar de placas planas inclinadas para calentamiento de aire*”, Revista de Ingeniería UC, Vol. 7, 2010.

Aprendizaje Basado en Proyectos en el Desarrollo de un Sistema Mecatrónico

Jiménez López Eusebio¹, Hernández Flores María Enedina², Bojórquez Morales Gloria Isabel³, Navarro Fragoso Lilia Beatriz³, Gaytán Martínez Lilia Zulema², Curiel Aguirre Fernando⁴ y Luna Sandoval Gabriel⁴

¹Universidad La Salle Noroeste – CIAAM-UTS-IIIMM, ejimenezl@msn.com

²CIAAM de la Universidad Tecnológica del Sur de Sonora, maenedina@hotmail.com, zgaytan@uts.edu.mx

³Universidad La Salle Noroeste, ibojorquez@ulsa-noroeste.edu.mx, lnavarro@ulsa-noroeste.edu.mx

⁴Universidad Estatal de Sonora, fher.curiel@gmail.com, gabriel.luna@ues.edu.mx

Resumen

En este artículo se presenta la aplicación del Aprendizaje Basado en Proyectos (ABProy) en el desarrollo de un sistema mecatrónico. El objetivo es generar experiencias didácticas en el uso de las metodologías activas, como es el caso del ABProy, para motivar al profesorado de la Universidad La Salle Noroeste, Universidad Tecnológica del Sur de Sonora y la Universidad Estatal de Sonora, a replantear, por medio del Enfoque Basado en Competencias (EBC), su labor docente en las aulas de clase. A pesar de que existen diversas metodologías activas, se consideró el ABProy ya que se adapta de forma natural a las ingenierías. Las materias relacionadas con el desarrollo del proyecto fueron Mecanismos y Seminario de Investigación de las carreras de Diseño Industrial y Mecatrónica de la Universidad La Salle Noroeste. El proyecto desarrollado se refiere al diseño, fabricación, modelación y simulación de un robot cartesiano didáctico. El diseño y la planeación del proyecto resultó ser esencial e importante para la aplicación del ABProy, y sobre todo, fue de gran ayuda para los alumnos el contar con profesores especialistas como asesores durante el desarrollo del prototipo. Las conclusiones de parte de los alumnos sobre el uso del ABProy fueron favorables y contribuyeron a la formación de competencias.

Palabras clave: EBC, Mecatrónica, ABProy, Robótica.

1. Introducción

Las competencias no son un concepto abstracto: se trata de actuaciones que tienen las personas para resolver problemas integrales del contexto, con ética, idoneidad, apropiación del conocimiento y puesta en acción de las habilidades necesarias. Existen como

tales desde el surgimiento del ser humano, porque son parte de la naturaleza humana en el marco de la interacción social y el ambiente ecológico. Las competencias son actuaciones integrales ante actividades y problemas del contexto, con idoneidad y compromiso ético, integrando el saber ser, el saber hacer y el saber conocer en una perspectiva de mejora continua [1].

La introducción de las competencias como objeto de la formación universitaria lleva a una revisión del propio concepto de formación. La formación es algo más que la mera información. Formar no es transmitir. La formación no es sólo acumular conocimientos. Las competencias hablan de conocimiento aplicado. El aprendizaje de competencias supone conocer, comprender y usar pertinenteamente los métodos de enseñanza con participación del alumno, donde la responsabilidad del aprendizaje depende directamente de su actividad, implicación y compromiso. Estos métodos son más formativos que meramente informativos, generan aprendizajes más profundos, significativos y duraderos y facilitan la transferencia a contextos más heterogéneos [2].

Por otro lado, el ABProy es una estrategia integradora que trasciende a la pedagogía activa, promueve pensar y actuar con base en el diseño de un proyecto, elaborando un plan con estrategias definidas, para dar una solución a una interrogante y no tan sólo cumplir objetivos curriculares [3]. El método de proyectos contiene cuatro elementos característicos: Desarrollo de habilidades de razonamiento aplicado a la realidad; la práctica se realiza en función del contenido teórico requerido por ella; se privilegia el saber hacer sobre el aprendizaje de contenidos; el contexto donde se realiza la práctica debe ser natural y el problema por abordar ha de anteponerse a los principios de las disciplinas, nunca al revés [4]. Esta estrategia de aprendizaje permite un

aceramiento entre la realidad laboral y ambiente académico. El ABProy no tiene un método único. Son cuatro las fases relacionadas con el Aprendizaje Basado en Proyectos [5]:

- 1) Información: Los estudiantes recopilan, por diferentes fuentes, informaciones necesarias para la resolución de la tarea planeada.
- 2) Planificación: Elaboración del plan de trabajo, la estructuración del procedimiento metodológico, la planificación de los instrumentos y medios de trabajo, y elección entre las posibles variables o estrategias de solución a seguir.
- 3) Realización: Supone la acción experimental e investigadora, ejercitándose y analizándose la acción creativa, autónoma y responsable.
- 4) Evaluación: Los estudiantes informan de los resultados conseguidos y conjuntamente con el profesor los discuten.

Para impulsar los cambios en el modelo educativo acordes con los enfoques modernos, la Universidad La Salle Noroeste, implementó en el año 2011, un Diplomado en Formación de Competencias del Perfil Docente Lasallista el cual tiene como finalidad capacitar al profesorado en la implementación en el aula de su modelo curricular por competencias, a partir de la puesta en práctica de sus herramientas conceptuales y procedimentales en la planeación, conducción y evaluación de los procesos de enseñanza y de aprendizaje, de acuerdo con el perfil docente Lasallista y los requerimientos curriculares de la materia que imparte [5]. Dentro de los temas tratados en el diplomado destaca la estrategia didáctica relacionada con el Aprendizaje Basado en Proyectos (ABProy). También, la Universidad Tecnológica del Sur de Sonora ha implementado proyectos de capacitación al profesorado en Educación Basada en Competencias, y se han hecho aplicaciones de las metodologías activas en proyectos de docencia [6].

En este artículo se presenta la aplicación del Aprendizaje Basado en Proyectos (ABProy) en dos materias de ingeniería de la Universidad La Salle Noroeste. El producto obtenido es un prototipo de un robot cartesiano. Se describen las tareas y actividades desarrolladas por el profesor y los alumnos durante las fases del ABProy. El objetivo es documentar experiencias derivadas de las aplicaciones de las metodologías activas para motivar a los profesores de las ingenierías a implementar los conocimientos aprendidos durante la capacitación en EBC en los distintos cursos y diplomados que toman en sus

universidades. Este proyecto forma parte de la RED ALFA y del proyecto PROMEP UTSSON-CA-4.

2. Consideraciones sobre la experiencia

Las materias consideradas para el desarrollo del prototipo mecatrónico fueron Mecanismos y Seminario de Investigación. La primera materia pertenece a la carrera de Diseño Industrial y se imparte en el sexto semestre; la segunda pertenece a la carrera de Ingeniería Mecatrónica y se cursa en octavo semestre. El grupo de Mecanismos estaba conformado por tres alumnos (un hombre y dos mujeres) y el grupo de Seminario de Investigación por 19 alumnos (sólo un alumno de este grupo participó en el proyecto cuyo objetivo principal fue el avanzar su tesis). Es importante considerar que: 1) la materia de Mecanismos se refiere al estudio cinemático de los movimientos de cadenas cinemáticas, 2) la materia de Seminario de Investigación trata temas de cómo desarrollar y documentar un proyecto (generalmente se concluye con una tesis), 3) puesto que se trata de un proyecto para la industria, un 40% del conocimiento que se requiere debe ser producto de la investigación y no de los contenidos de las materias, 4) debido a la complejidad del proyecto, se debían planear sesiones extras para detonar el conocimiento previo en los alumnos en temas ya vistos con anterioridad.

3. Aplicación del ABProy

En esta sección se describirán las fases relacionadas con el ABProy en el desarrollo del proyecto.

3.1 Fase 1: Información

Esta fase es de suma importancia en el ABProy y consiste principalmente en la planeación del proyecto y en la búsqueda de información relevante. El profesor líder y los alumnos tienen tareas específicas que realizar. Para el caso del profesor, fueron buscar un proyecto relevante para motivar a los alumnos y una vez definido el mismo, ayudarlos a formular un procedimiento para lograr los objetivos. Se buscó una empresa relacionada con la capacitación en mecatrónica y automatización industrial y ésta propuso el desarrollo de un robot cartesiano el cual cumpliera con las especificaciones generales siguientes:

- 1) Diseño en CAD de las partes y ensamble del robot.
- 2) Desarrollo de un simulador del robot integrando trayectorias rectilíneas con perfiles de trayectoria trapezoidales.

- 3) Uso de técnicas de manufactura aditiva para la fabricación del robot.
- 4) Debe incluir motores paso a paso.
- 5) El control y manejo del robot debe realizarse por medio de un dispositivo móvil (Tablet o celular).

Con el proyecto y las especificaciones dadas por la empresa, el profesor reunió a los alumnos interesados (cuatro alumnos, tres de Diseño Industrial y uno de Ingeniería Mecatrónica). Después de algunas discusiones técnicas y de una búsqueda inicial de información, el profesor propuso la siguiente guía metodológica (aunque esta actividad debe ser de los alumnos, el profesor participó debido a la complejidad del proyecto y porque los resultados debían entregarse a la empresa):

- i. Construir el modelo cinemático del robot cartesiano (usar conocimientos de Cálculo Vectorial y métodos de modelación cinemática de la materia de Mecanismos).
- ii. Generar el modelo de la trayectoria usando interpolación polinomial.
- iii. Integrar los modelos del robot y de la trayectoria.
- iv. Usar un software de cálculo formal para la programación del modelo cinemático (usar conocimientos de la materia de Programación).
- v. Diseñar el robot en un software CAD.
- vi. Construir un simulador del robot para Ipad.
- vii. Fabricar los elementos del robot usando manufactura convencional y aditiva.
- viii. Desarrollar el control del prototipo.
- ix. Efectuar las pruebas necesarias a los prototipos desarrollados.
- x. Entregar el proyecto a la empresa.

Las tareas principales desarrolladas por los alumnos fueron las siguientes:

- Integración de equipos de trabajo. Las funciones del grupo de la materia de Mecanismos fueron: generar el modelo cinemático del robot y la trayectoria, y diseñar y fabricar el prototipo. El alumno de Seminario de Investigación desarrolló el simulador y el control del robot.
- -Recopilación de información inicial. Consultaron tesis, artículos e informes técnicos relacionados con temas de mecanismos, robótica, software y control, con el propósito de elaborar un plan de trabajo para la realización de los proyectos.

3.2 Fase 2: Planificación

La planificación del proyecto es una de las fases más importantes del APBproy. Al igual que la fase anterior, el profesor y los alumnos tienen tareas específicas. Para el profesor fueron las siguientes: 1) Revisar el plan de trabajo, 2) Asesorar sobre la guía metodológica utilizada durante el desarrollo del proyecto, 3) Ubicar asesores internos y externos que ayuden a los alumnos, 4) Gestionar con la empresa solicitante los recursos para la compra de materiales, 5) Señalar y ubicar talleres internos y externos para la fabricación de partes y componentes, 6) Planificar reuniones periódicas con cada equipo de trabajo y con el grupo completo para revisar los avances o aclarar dudas y 7) Diseñar las evaluaciones de los aprendizajes.

Las tareas de los alumnos fueron: 1) Planificar las tareas y actividades técnicas del proyecto, 2) Buscar, cotizar y comprar las partes, componentes y sistemas del prototipo, 3) Evaluar el o los paquetes computacionales que se usarán, 4) Buscar talleres y empresas de apoyo para la fabricación del robot, 5) Localizar asesores internos y externos, 6) Preparar las exposiciones para dar a conocer el proyecto

3.3 Fase 3: Realización

Tarea 1: Desarrollo de un simulador

Una de las primeras tareas que los alumnos realizaron fue el desarrollo de un simulador computacional el cual imita el comportamiento cinemático de un robot cartesiano. Para desarrollar esta actividad fue necesario que el profesor detonara el conocimiento previo al dar asesorías de matemáticas y cinemática, tanto de temas que ya se habían visto en otras materias como nuevos temas. La tarea 1 fue dividida en tres subtareas: 1) Desarrollo del modelo cinemático del robot, 2) Diseño en Solidworks 2014 del prototipo y 3) Construcción del simulador para su manejo en una Tablet. A continuación se describen a detalle:

Desarrollo del modelo cinemático del robot

En esta parte los alumnos de Diseño Industrial analizaron el modelo cinemático considerando dos tipos de análisis: 1) De cuerpo rígido (para modelar los movimientos del robot) y 2) De la partícula (para modelar la trayectoria). El modelo de trayectoria lineal requirió dos análisis: 1) Del lugar geométrico (análisis vectorial del movimiento) y 2) Análisis funcional (modelos de los perfiles de trayectoria trapezoidal y suavización de perfiles usando interpolación polinomial grado 5). Los modelos finales de la trayectoria fueron analizados por un

alumno de mecatrónica el cual los programó en MatLAB.

Diseño en Solidworks del prototipo.

El diseño del robot en un software de dibujo fue desarrollado por el grupo de Diseño Industrial. Se generó un modelo en sólidos del prototipo en Solidworks 2014. Para generar el prototipo virtual fue necesario que los alumnos de Diseño Industrial y de Mecatrónica, seleccionaran y dimensionaran las piezas y los elementos de trabajo. La Figura 1 muestra el prototipo virtual del robot:



Fig.1: Robot dibujado en Solidworks

Construcción del simulador para su manejo en una Tablet.

Esta tarea fue realizada por un alumno de Mecatrónica. Para ello requirió los archivos en Solidworks del robot que previamente habían desarrollado los alumnos de Diseño Industrial. Las tecnologías usadas para el diseño del simulador fueron: archivos de MatLAB (funciones y cálculo de la cinemática del robot), SolidWorks (migración de partes gráficas del robot), Xcode, Migrador SolidWorks-OBJ-VertexData y Open GL ES 2.0. La Figura 2 muestra el simulador del robot.



Fig. 2: Simulador del robot

Tarea 2: Construcción y ensamblaje del robot

La segunda tarea del proyecto fue la construcción del robot. Esta actividad fue realizada principalmente por los alumnos de Diseño Industrial. La tarea se dividió en: 1) Dibujo del robot en

Solidworks 2014, 2) Compra de materiales y dispositivos, 3) Fabricación de partes y componentes y 4) Ensamble del prototipo. Para la fabricación y el ensamblaje del prototipo los alumnos fueron apoyados por profesores expertos en soldadura y maquinado. Algunas piezas del robot fueron fabricadas usando las técnicas de manufactura aditiva (impresión 3D). La Figura 3 muestra el prototipo del robot cartesiano fabricado y ensamblado.



Fig.3: Prototipo fabricado y ensamblado

Tarea 3: Diseño e implementación del control

La tercera tarea relacionada con el prototipo fue el diseño del control. Ésta la realizó un alumno de Mecatrónica. Para mover el robot fueron utilizados dos motores a pasos bipolares de 12 V. Los motores fueron controlados por un microprocesador integrado a una tarjeta comercial arduino UNO. Para manejar el prototipo se realizó una conexión entre el simulador desarrollado en la primera tarea y el prototipo. Para dicha conexión se utilizó un dispositivo Bluetooth 4.0 el cual se conectó a la tarjeta arduino. Para la etapa de potencia se usaron tarjetas comerciales A4988. La Figura 4 muestra los elementos principales del control.



Fig. 4: Elementos principales del control

Tarea 4: Pruebas del prototipo

Una vez fabricado y ensamblado el prototipo, se procedió a probar el control y el simulador. La Figura 5 muestra el prototipo en la fase de pruebas.



Fig. 5. Control desde iPad

3.4 Fase 4: Evaluación

Otra de las fases importantes en el ABProy es la evaluación. Para el caso del proyecto descrito en este artículo, se realizaron dos clases de evaluaciones: 1) se llevó un control de avances y exámenes durante la duración del proyecto y 2) Proyecto terminado en cuanto a funcionalidad y exposiciones. Los siguientes conceptos fueron evaluados:

- Tres exámenes parciales y un examen final.
- Se evaluó en número de consultas hechas a los asesores técnicos.
- Se realizó una presentación final del proyecto en la exposición final de clases.
- Se evaluó la colaboración en equipo tanto de la misma materia como la colaboración de las tres materias.
- Se evaluó la funcionalidad del prototipo ante representantes de la empresa.
- Se entregaron tres documentos científicos (tres artículos).
- Se desarrolló un resumen ejecutivo por parte de los alumnos sobre el proyecto y sobre la metodología activa utilizada.
- La Figura 6 muestra uno de los carteles usados por los alumnos durante la exposición de proyectos de fin de cursos.



Fig. 6. Carteles para la exposición del proyecto

Conclusiones

En este artículo se ha descrito la aplicación del ABProy en el desarrollo de un prototipo mecatrónico. Las principales conclusiones se resumen en los puntos siguientes:

-El desarrollo de sistemas y productos mechatrónicos didácticos exige una multidisciplinariedad de campos de conocimientos y de la aplicación de métodos de aprendizaje diferentes a los convencionales. El prototipo mechatrónico desarrollado requirió conocimientos de las matemáticas, física, computación y electrónica, así como de la integración de tecnologías de control y manufactura convencional y avanzada. Para su desarrollo, fue necesario utilizar la metodología activa ABProy en las clases de Mecanismos y Seminario de Investigación, pues por su complejidad no era posible aplicar los métodos de la enseñanza tradicional. Los resultados de la aplicación del ABProy fueron importantes, pues se generó un producto de alta integración mechatrónica bajo la planeación y supervisión de la metodología activa, se mostró una alta aceptación y adaptación por parte de los alumnos a la metodología y se entendió que el desarrollo de este tipo de proyecto sólo es posible si se trabaja en equipo y se acude con asesores especialistas.

-La metodología del ABProy permite diseñar evaluaciones en donde los exámenes no son el centro ni lo más importante, más bien el producto final, la búsqueda de conocimientos, la asesoría y

la defensa y exposición del proyecto es lo más importante.

-La capacitación del profesorado es crucial e importante para conocer y aplicar el Enfoque por Competencias y las metodologías activas. El desarrollo del proyecto descrito en este artículo ha sido posible por la capacitación continua que reciben los profesores de ingeniería en el diplomado de competencias lasallistas y por los cursos de EBC impartidos en la Universidad Tecnológica del Sur de Sonora.

-En el diseño de las metodologías activas para favorecer la formación de competencias el reto se encuentra en ampliar el repertorio metodológico intentando conocer bien y en profundidad las posibilidades de las diferentes estrategias e ir experimentando su aplicación en la práctica educativa consiguiendo, de este modo, la apropiación y adaptación a nuestras circunstancias y posibilidades de dichos métodos [2].

Agradecimientos

Los autores de este trabajo agradecen a las Universidades y Empresas que integran la RED Interinstitucional ALFA, a la RED de Pyme de Noroeste, a la empresa Spin – OFF Innovación en Ingeniería de Manufactura y Mantenimiento S. de R.L. MI, al Cuerpo Académico CIAAM de la UTS, al Programa del Mejoramiento del Profesorado (PROMEP) bajo el proyecto Desarrollo de un Prototipo de Robot Industrial para Aplicaciones en Celdas de Producción Automatizadas Clave UTSSON-CA-4, por el apoyo brindado para la realización de este trabajo.

Referencias

- [1] Tobón S., Pimienta J., García J. “Secuencias didácticas: aprendizaje y evaluación de

competencias”. Prentice Hall-Pearson. España, Primera edición. 2010.

- [2] Fernández A. “Metodologías activas para la formación de competencias”. *Educatio siglo XXI*, Vol. 24, pp. 35 – 56, 2006.
- [3] Galeana L. “Aprendizaje basado en proyectos”. *Revista Ceupromed. Universidad de Colima*, Revista digital 1/27. 2006. Documento recuperado el 12 de marzo de 2013. URL:<http://ceupromed.ucol.mx/revista/PdfArt/1/27.pdf>.
- [4] De Miguel M. “Modalidades de enseñanza centradas en el desarrollo de competencias. Orientaciones para promover el cambio metodológico en el Espacio Europeo de Educación Superior” (Proyecto EA2005-0118). España: Universidad de Oviedo. 2005
- [5] Jiménez E., Bojórquez G., Navarro L., Amavizca L., López L., Martínez V., López S., Hernández M. “Experiencias de la aplicación del Aprendizaje Basado en Proyectos de la materia de Mecanismos en la ULSA Noroeste”. Eleventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI). Agosto 14 al 16, Cancún, México, 2013.
- [6] López L., Bojórquez A., Ochoa D., Jiménez E., Amavizca L., Hernández M., Luna A. “Experiencias de la implementación de un programa de refuerzo de la competencia basada en estrategias de enseñanzas orientadas a los docentes de la Universidad Tecnológica del Sur de Sonora”. XII International Conference on Engineering and Technology Education. March 11 - 14, Dili, East Timor. 2012.

Diseño Conceptual de un Robot Paralelo Reconfigurable

Sánchez Alonso Róger Ernesto, González Barbosa José Joel y Castillo Castañeda Eduardo

Instituto Politécnico Nacional. Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Unidad Querétaro. Cerro Blanco No. 141, Colinas del Cimatario, Querétaro, Qro, C. P. 76099.

Resumen

En este trabajo se reporta el diseño a nivel de concepto de un robot paralelo reconfigurable. El prototipo tomado como modelo base a reconfigurar es un manipulador paralelo de 6 GDL que se encuentra en proceso de patentamiento ante el Instituto Mexicano de la Propiedad Industrial.

El esquema de reconfiguración seleccionado es el de geometría variable, por lo que primero se identificaron todos los parámetros que definen la geometría del manipulador, para posteriormente, tomando en cuenta una serie de requerimientos técnicos, determinar aquellos a considerar como base para la reconfiguración del robot. Después se tomó como criterio de selección definitivo la generación de espacios de trabajo para establecer la mejor estrategia de reconfiguración. Finalmente se presenta una propuesta de diseño conceptual para el robot manipulador reconfigurable.

Palabras clave: robot paralelo, reconfiguración, geometría variable.

1. Introducción

Actualmente la industria tiene la necesidad de poder ajustar sus líneas de producción para hacer frente a cambios inesperados en las exigencias del mercado. Por lo anterior, la ingeniería robótica tiene el reto de desarrollar manipuladores flexibles que puedan desempeñarse eficientemente en cualquier entorno, situación o condición de trabajo, haciendo especial énfasis en los manipuladores paralelos por sus ventajas competitivas sobre los robots tipo serie.

Una estrategia de reconfigurabilidad es quizás el enfoque más razonable para proveer de flexibilidad a un robot manipulador [1]. En el caso de manipuladores paralelos los principales avances en esta materia pueden agruparse en dos categorías; los

basados en un diseño modular y los basados en un diseño con geometría variable [2].

Un diseño modular consiste en un conjunto de módulos estandarizados, tales como actuadores, articulaciones, eslabones y efectores finales, los cuales pueden ser rápidamente conectados y desconectados para obtener diferentes configuraciones en un robot manipulador, los trabajos presentados en [3-7] constituyen ejemplos de esta estrategia aplicada a robots manipuladores paralelos.

Por otro lado, en un diseño de geometría variable las dimensiones de los parámetros geométricos del manipulador son modificadas para alcanzar nuevas configuraciones. El dodecápodo [8], la plataforma Gough-Stewart reconfigurable planteada en [9], y el robot paralelo presentado en [2], constituyen ejemplos de este enfoque, el cual se encuentra más desarrollado que el de tipo modular ya que presenta la ventaja de una fácil implementación durante la operación del robot.

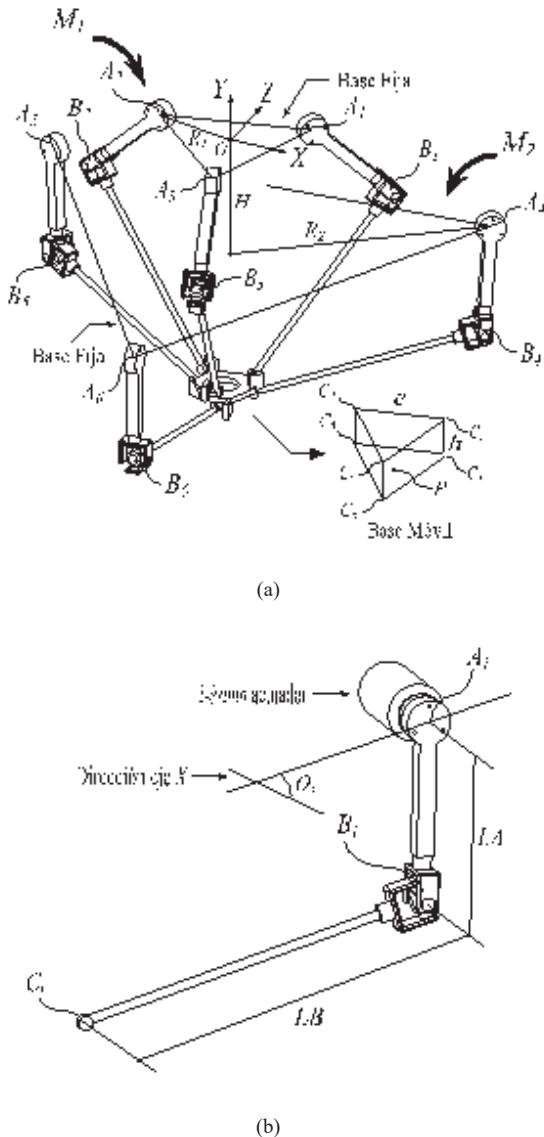
A pesar de los avances obtenidos, en la actualidad el diseño de un manipulador paralelo reconfigurable de gran versatilidad y desempeño sigue siendo un desafío, por lo que la búsqueda de un diseño innovador y de gran potencial resulta en una tarea altamente justificada para la ingeniería robótica.

En este trabajo se presenta el diseño conceptual de un robot paralelo reconfigurable cuyo modelo base es un manipulador de 6 GDL que por sí sólo presenta grandes ventajas cinemáticas y que actualmente se encuentra en proceso de patentamiento ante el IMPI.

2. Descripción geométrica del modelo base

El manipulador tomado como modelo base a reconfigurar (proceso de patentamiento: MX/a/2013/011009) consiste en dos submanipuladores paralelos tipo 3-RUS que comparten

base móvil (Fig. 1a), donde R, U y S representan articulaciones de revoluta, universales y esféricas respectivamente, y el subrayado indica la junta activa. A menos que se especifique de otra manera los subíndices $i = 1, 2, 3$ se refieren a los elementos del sub-manipulador llamado M_1 , mientras los subíndices $i = 4, 5, 6$ se refieren a los elementos del sub-manipulador llamado M_2 .



**Fig. 1. Modelo base a reconfigurar. (a) Vista general.
 (b) i -ésima cadena cinemática.**

La plataforma fija del manipulador es representada por dos triángulos equiláteros paralelos, $A_1A_2A_3$ y $A_4A_5A_6$, separados por una distancia H . El sistema de referencia global del robot está ubicado en el centro O del triángulo $A_1A_2A_3$, con sus ejes X y Z yaciendo en el plano definido por este triángulo, y el

eje Y apuntando hacia arriba. Las juntas de revoluta definidas en los puntos A_i son accionadas por actuadores rotacionales, y determinan los vértices de los triángulos equiláteros $A_1A_2A_3$ y $A_4A_5A_6$, donde R_1 y R_2 representan los circunradios de estos triángulos, y θ_i representa la orientación de cada una de las seis cadenas cinemáticas del robot (Fig. 1b), la cual está medida desde la dirección del eje X hasta A_i , para $i = 1, 2, 3$, y hasta la proyección de A_i en el plano XZ , para $i = 4, 5, 6$.

Por otro lado, los puntos B_i denotan la posición de la articulación universal que conecta el eslabón de longitud LA con el eslabón de longitud LB en la misma cadena cinemática. De forma similar, los puntos C_i denotan la posición de la articulación esférica que conecta la base móvil con el eslabón de longitud LB . Los puntos C_i forman un prisma triangular de lado e y de altura h , donde las bases $C_1C_2C_3$ and $C_4C_5C_6$ son triángulos equiláteros. Finalmente, el punto P es el punto de interés en la base móvil (elemento efector) y está ubicado convenientemente en el centro del triángulo $C_4C_5C_6$.

Los parámetros que definen la geometría del manipulador de 6 GDL y sus valores tomados como referencia se muestran en la Tabla 1.

Tabla 1. Parámetros geométricos de referencia

Parámetro	Valor
R_1	200 mm
R_2	450 mm
H	200 mm
LA	200 mm
LB	400 mm
h	34 mm
e	86.60 mm
θ_1	35°
θ_2	155°
θ_3	275°
θ_4	25°
θ_5	145°
θ_6	265°

3. Determinación de los parámetros geométricos a considerar como base para la reconfiguración

El esquema de reconfiguración seleccionado es el de geometría variable. La razón es que dicho esquema presenta la ventaja de poder implementarse fácilmente durante la operación del robot, es decir, no hay necesidad de parar al robot para reconfigurarlo tal y como sucede en la mayoría de robots modulares.

Una reconfiguración de tipo geométrica implica la variación de uno, algunos o todos los parámetros mostrados en la Tabla 1. En este trabajo se toma la opción de variar solamente un parámetro, ésto para no intensificar el problema de redundancia cinemática que se generará al reconfigurar el robot.

La selección del parámetro considerado como base para la reconfiguración se hace tomando en cuenta aquel que implique un diseño mecánico que cumpla con los siguientes requerimientos técnicos: simplicidad para la construcción y de fácil implementación en el manipulador (Req.1), que no agregue cargas excesivas en las cadenas cinemáticas para no afectar drásticamente la capacidad de carga, rigidez y exactitud del manipulador (Req.2), que no genere interferencias en el espacio de trabajo operable del manipulador (Req.3), que no modifique drásticamente el espacio físico que ocupa el manipulador, pues de otra manera la reconfiguración sería prácticamente igual a hacer un robot de dimensiones superiores (Req.4), que favorezca la reconfiguración durante la operación del robot (Req.5), y que minimice el uso de actuadores adicionales (Req.6).

La evaluación de los parámetros geométricos del robot considerando los requerimientos técnicos antes mencionados se muestra en las Tablas 1, 2 y 3.

Tabla 1. Evaluación de los parámetros R_1 , R_2 y H

Requerimientos	Imp. Relat.	R_1		R_2		H	
		Calif.	Pond.	Calif.	Pond.	Calif.	Pond.
Req.1	0.2	9	1.8	4	0.8	8	1.6
Req.2	0.25	10	2.5	10	2.5	10	2.5
Req.3	0.2	10	2	1	0.2	10	2
Req.4	0.15	9	1.35	5	0.75	9	1.35
Req.5	0.1	9	0.9	6	0.6	9	0.9
Req.6	0.1	9	0.9	6	0.6	8	0.8
Total	1	9.45		5.45		9.15	

Tabla 2. Evaluación de los parámetros LA , LB y h

Requerimientos	Imp. Relat.	LA		LB		h	
		Calif.	Pond.	Calif.	Pond.	Calif.	Pond.
Req.1	0.2	10	2	10	2	2	0.4
Req.2	0.25	4	1	3	0.75	4	1
Req.3	0.2	10	2	10	2	10	2
Req.4	0.15	8	1.2	8	1.2	10	1.5
Req.5	0.1	10	1	10	1	6	0.6
Req.6	0.1	8	0.8	8	0.8	2	0.2
Total	1	8		7.75		5.7	

Tabla 3. Evaluación de los parámetros e y θ_i

Requerimientos	Imp. Relat.	e		θ_i	
		Calif.	Pond.	Calif.	Pond.
Req.1	0.2	2	0.4	2	0.4
Req.2	0.25	4	1	10	2.5
Req.3	0.2	10	2	2	0.4
Req.4	0.15	10	1.5	10	1.5
Req.5	0.1	6	0.6	7	0.7
Req.6	0.1	2	0.2	2	0.2
Total	1	5.7		5.7	

Tal y como se observa en las tablas anteriores, el parámetro geométrico que implica un diseño más ajustado a los requerimientos es R_1 (9.45 pts.), es decir la reconfiguración del robot estaría basada en el redimensionamiento del triángulo $A_1A_2A_3$ de la base fija del mecanismo. Por otro lado, el parámetro H parece también implicar un diseño factible (9.15 pts.), es decir la reconfiguración del robot podría estar también basada en redimensionar la separación entre los triángulos $A_1A_2A_3$ y $A_4A_5A_6$ que definen la base fija del mecanismo.

4. Selección de la estrategia de reconfiguración

En la Fig. 2 se muestra tanto la i como la $i+3$ -ésima cadena cinemática del robot (para $i = 1, 2, 3$), en donde fácilmente se puede intuir las implicaciones geométricas generadas sobre el mecanismo producto de una variación en el parámetro R_1 o H . En el caso de una variación del parámetro R_1 , las coordenadas de las articulaciones activas de revoluta definidas en los puntos A_i se ven afectadas horizontalmente, por otro lado, para una variación del parámetro H , son las coordenadas de las articulaciones activas de revoluta definidas en los puntos A_{i+3} las que se ven afectadas verticalmente.

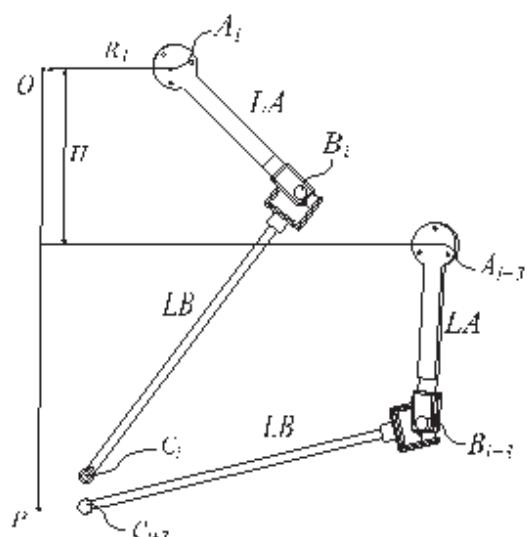


Fig. 2. Representación de la i e $i+3$ -ésima cadena cinemática del robot de paralelo de 6 GDL.

La estrategia de reconfiguración tratada anteriormente hace referencia a la variación de solamente un parámetro, ya sea R_1 o H . Sin embargo en la Fig. 3 se presenta un esquema más complejo que implica la modificación de ambos parámetros mediante la inclusión de solamente un grado de libertad al sistema. Esta estrategia de reconfiguración,

propuesta en un trabajo desarrollado en el Instituto Politécnico Nacional (CICATA-Qro) que se encuentra en proceso de patentamiento ante el IMPI (MX/a/2013/006781), se logra al agregar en las $i = 1, 2, 3$ cadenas cinemáticas pertenecientes al sub-manipulador M_i un eslabón móvil de longitud R_f , el cual será llamado de aquí en adelante eslabón de reconfiguración. Las articulaciones de revoluta que permiten la movilidad del eslabón de reconfiguración están localizadas en el punto F_i a una distancia R_f del sistema de referencia global del robot en la dirección de θ_i .

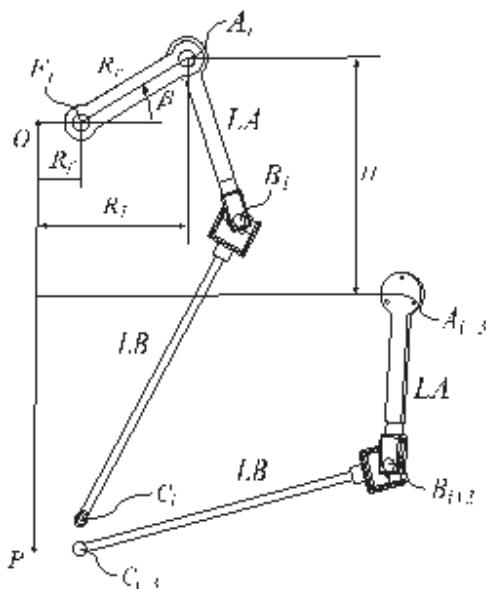


Fig. 3. Estrategia de reconfiguración basada en la variación simultánea de R_I y H .

Como se observa en la Fig. 3, cuando la articulación definida en F_i es accionada, ésta genera un movimiento angular definido por β en el eslabón de reconfiguración, lo que a su vez implica un reposicionamiento de las articulaciones activas definidas en los puntos A_i , que puede ser visto como un redimensionamiento simultáneo de los parámetros R_I y H .

De lo anterior se puede decir que existe tres posibles estrategias de reconfiguración que parecen técnicamente viables; la primera estrategia implica la variación de R_I , la segunda la variación de H , y la tercera la variación simultánea de ambos parámetros mediante el esquema de la Fig. 3.

Para poder decidir qué estrategia conviene implementar se realizó un estudio que permitiera determinar el comportamiento del espacio de trabajo del robot frente a cada una de las estrategias, los resultados se muestran en la Fig. 4.

En la Fig. 4 se puede observar que el espacio de trabajo del robot tiende a disminuir en la medida que aumenta la longitud del parámetro R_I (curva azul delgada), situación un poco distinta en el caso de la variación de H (curva roja delgada) y en el caso de la variación simultánea de R_I y H (curva negra delgada), en donde la relación con el espacio de trabajo es mayoritariamente directa. El análisis anterior permite inferir que la configuración del robot que provee el máximo espacio de trabajo, aproximadamente $5.38 \times 10^7 \text{ mm}^3$, se da bajo la estrategia basada en la variación simultánea de R_I y H , específicamente cuando la posición angular β del eslabón de reconfiguración está entre 50 y 60 grados.

El resultado anterior no es lo suficientemente concluyente como para seleccionar la estrategia de reconfiguración, pues no basta con seleccionar una estrategia que permita obtener el mayor espacio de trabajo dada una configuración, ya que esto en realidad lo que representa es un sub-espacio de trabajo. Lo adecuado es analizar la unión de los sub-espacios de trabajo generados por una estrategia de reconfiguración, ya que en una configuración dada se pueden alcanzar puntos que en otras configuraciones no es posible, de ahí la necesidad de utilizar la unión de los sub-espacios de trabajo como una medida del espacio de trabajo global de un robot reconfigurable.

Tomando en cuenta lo anterior, en la Fig. 4 se muestra el comportamiento del espacio de trabajo global del manipulador con cada estrategia de reconfiguración. En el caso del espacio global generado con la estrategia que implica la variación de R_I , éste permanece prácticamente constante dentro del rango de análisis (curva azul gruesa), presentando su máximo en $5.44 \times 10^7 \text{ mm}^3$ cuando $R_I = 200 \text{ mm}$. En el caso de la estrategia basada en la variación de H , el espacio de trabajo global tiende a aumentar significativamente en la medida que aumenta este parámetro (curva roja gruesa), alcanzando su máximo en $6.59 \times 10^7 \text{ mm}^3$ cuando $H = 350 \text{ mm}$. Finalmente, en el caso de la estrategia de variación simultánea de R_I y H , el espacio de trabajo global también tiende a aumentar en la medida que el eslabón de reconfiguración se mueve de -90 a 90 grados, alcanzando su máximo en $6.79 \times 10^7 \text{ mm}^3$ cuando $\beta = 70$ grados.

Debido a que el máximo espacio de trabajo global se obtiene mediante la estrategia de reconfiguración simultánea de R_I y H , es dicha estrategia la que se selecciona como base para la reconfiguración del robot. En la Fig. 5 se muestra el espacio de trabajo global del robot producido por la estrategia de reconfiguración basada en la variación simultánea de R_I y H .

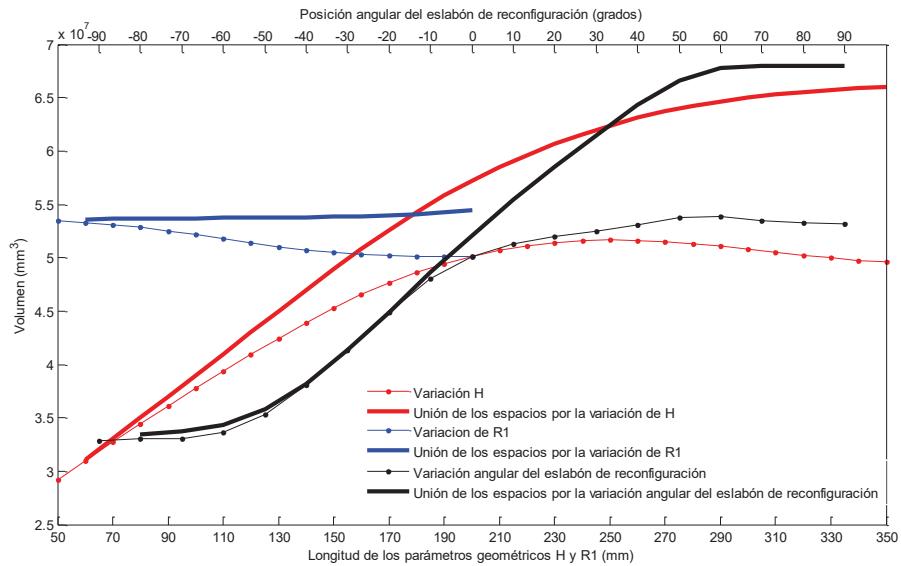


Fig. 4. Comportamiento del espacio de trabajo del robot frente a tres estrategias de reconfiguración.

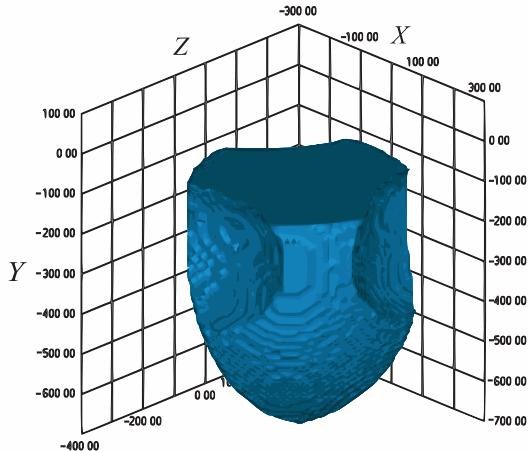


Fig. 5. Visualización 3D del espacio global de trabajo bajo la estrategia de reconfiguración basada en la variación simultánea de R_1 y H .

Para estimar el volumen del espacio de trabajo del robot dada una determinada configuración perteneciente a cada una de las tres estrategias evaluadas, se implementó el método de discretización. En este método un espacio más grande que el espacio de trabajo del robot es discretizado uniformemente en una serie de puntos, de manera que aquellos puntos que satisfagan el modelo inverso de posición del robot son considerados como pertenecientes a su espacio de trabajo. El desarrollo del modelo inverso del robot no es un tópico de interés en este trabajo y por ello no será abordado. Por otro lado, es importante mencionar que para la evaluación del espacio de trabajo generado por la estrategia de reconfiguración seleccionada, se

consideró $R_f = 50$ mm y $R_r = 150$ mm (Fig. 3). Lo anterior con el propósito de que cuando $\beta = 0$ se cumpla que $R_l = 200$ mm, lo que corresponde a la configuración del robot sin reconfigurar (Tabla 1).

5. Propuesta de diseño conceptual

Para que el esquema de reconfiguración seleccionado implique la inclusión de solamente un grado de libertad al sistema mecánico, el posicionamiento angular de los tres eslabones de reconfiguración debe ser idéntico y simultáneo, de esta manera no se intensifica drásticamente el problema de redundancia cinemática. La redundancia cinemática es entendida como el hecho de que el modelo cinemático inverso del manipulador tiene infinitas soluciones para una misma pose.

Existen muchos mecanismos capaces de generar el movimiento deseado en el eslabón de reconfiguración a través de un único grado de libertad. Uno de ellos y quizás el más estudiado y sencillo de implementar, es el mecanismo Manivela-Bielas-Corredera. En la Fig. 6 se presenta una propuesta de diseño conceptual para el robot paralelo reconfigurable. Dicha propuesta está basada en la incorporación de un mecanismo que está compuesto por tres sub-mecanismos tipo Manivela-Bielas-Corredera que son accionados por un único actuador. Es importante mencionar que este mecanismo de reconfiguración es parte de la solicitud de patente (MX/a/2013/006781).

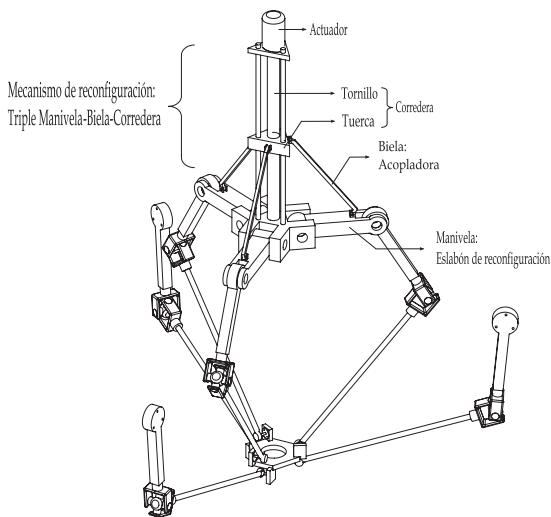


Fig. 6. Propuesta de diseño conceptual para el robot paralelo reconfigurable.

Cuando el actuador ubicado en la parte superior del mecanismo de reconfiguración es accionado en un sentido u otro, se genera el giro de un tornillo que hace que una tuerca se desplace verticalmente hacia arriba o hacia abajo. Esta tuerca está conectada a una barra acopladora, última que está conectada a su vez al eslabón de reconfiguración, de manera que cuando la tuerca sube o baja, el eslabón de reconfiguración realiza el movimiento de balancín deseado.

6. Conclusiones

Se presentó el diseño conceptual de un robot paralelo reconfigurable basado en un esquema de geometría variable. La estrategia de reconfiguración seleccionada se fundamentó en el cumplimiento de una serie de requerimientos técnicos, en su mayoría relacionados con la fácil y adecuada implementación de la estrategia de reconfiguración sobre el prototipo tomado como modelo base, el cual es un manipulador paralelo de 6 GDL que actualmente se encuentra en proceso de patentamiento ante el IMPI, razón por la cual no se entró en muchos detalles de su movilidad y comportamiento cinemático.

Existe mucho trabajo a desarrollar a partir del diseño propuesto: el diseño a detalle, el análisis cinemático y dinámico del prototipo, y el desarrollo de una estrategia de optimización que permita resolver la redundancia cinemática inducida debido a la reconfiguración. Además resulta importante evaluar el efecto positivo que genera la reconfiguración más allá de un incremento en el

espacio de trabajo del robot, por ejemplo estudiar el escape de configuraciones singulares y una potencial mejora en el desempeño cinetostático del robot.

Referencias

- [1] El Maraghy-Hoda A. “*Changeable and Reconfigurable Manufacturing Systems*”, Springer Series in Advanced Manufacturing, Springer, London, 2009.
- [2] Zhang D., Shi Q. “*Novel Design and Analysis of a Reconfigurable Parallel Manipulator Using Variable Geometry Approach*”, *Advances in Intelligent and Soft Computing*, 124: 447-457, 2012.
- [3] Guilin Y., I-Ming C., Wee K., Song H. “*Kinematic Design of Modular Reconfigurable in-Parallel Robots*”, *Autonomous Robots*, 10: 83-89, 2001.
- [4] Brisian C. “*Designing Aspects of a Special Class of Reconfigurable Parallel Robots*”, *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, 101-106, 2007.
- [5] Gogu G. “*Isogilden-TaRb: A Family of up to Five Axes Reconfigurable and Maximally Regular Parallel Kinematic Machines*”, *International Conference on Smart Machining Systems*, Gaithersburg, 2007.
- [6] Haage M., Dressler I., Robertsson A., Nilsson K., Brogardh T., Johanson R. “*Reconfigurable Parallel Kinematic Manipulator for Flexible Manufacturing*”, 13th IFAC Symposium on Information Control Problems in Manufacturing, 145-150, Moscow, Russia, 2009.
- [7] Xi F., Li Y., Wang H. “*Module-Based Method for Design and Analysis of Reconfigurable Parallel Robots*”, *Frontiers of Mechanical Engineering*, 6: 151-159, 2011.
- [8] Bande P., Seibt M., Uhlmann E., Saha S., Rao P. “*Kinematics Analyses of Dodekapod*”, *Mechanism and Machine Theory*, 40: 740-756, 2005.
- [9] Ji Z., Song P. “*Design of a Reconfigurable Platform Manipulator*”. *Journal of Field Robotics*, 15: 341-346, 1998.

Análisis de la Distribución de Temperatura sobre una Barra con Arduino

Aguilar-Loreto Omar¹, Muñoz-Gómez José Antonio¹ y Fernández-Raudales Juan Manuel²

¹ Departamento de Ingenierías CUCSur Universidad de Guadalajara
Av. Independencia Nacional 151, 48900, Autlán de Navarro, Jalisco, México
Tel. 01 (317) 3825010 Ext. 57087

² Flextronics Guadalajara, Carretera A La Base Aérea 5850-4,
La Mora, 45136 Zapopan, Jalisco 01 33 3818 3200

Resumen

En este trabajo se diseña y se construye un sistema electromecánico para el estudio de la ecuación del calor en una barra horizontal. Se utiliza un conjunto de diez sensores distribuidos uniformemente sobre una barra metálica hecha de latón. Cada señal del sensor es adquirida por el microcontrolador Arduino Uno para su posterior análisis. Al mismo tiempo, se resuelve en forma abstracta la ecuación del calor, cuyas gráficas teóricas se comparan con los datos empíricos a fin de corroborar y sustentar el modelo de la ecuación del calor en una barra horizontal metálica.

Palabras clave: Ecuación del Calor, Sensor de Temperatura, Prototipo Medición del Calor.

1. Introducción

Los intentos de construir sistemas mecánicos automatizados tienen una historia fascinante. La piedra angular de mecatrónica es la relación interdisciplinaria en las áreas de la ingeniería mecánica, electricidad, ingeniería electrónica y ciencias computacionales. Esta tecnología ha producido diversos productos nuevos que usamos en nuestra vida diaria [1]-[3].

Uno de los fenómenos que se presenta en la naturaleza y que presenta variadas aplicaciones tanto tecnológicas, como científicas es el estudio de la energía térmica. Desde los primeros tiempos el hombre se ha visto en la necesidad de entender y comprender como se presenta dicho fenómeno del calor.

Si bien es cierto que existen trabajos al respecto [4]-[6], en la presente investigación además de un análisis de propagación del calor, se contempla la creación de un dispositivo electromecánico combinando los conocimientos de mecatrónica, para

obtener resultados óptimos en la medición del calor. Para ello emplearemos diez sensores de temperatura distribuidos uniformemente sobre la superficie de la placa [7,8], las mediciones serán adquiridas mediante la tarjeta Arduino Uno [9]-[12]. Los datos registrados se confrontarán con los valores teóricos obtenidos de la ecuación de calor. Con ello se validará el modelo teórico-práctico de la distribución de calor sobre una barra metálica.

El resto del artículo se organiza de la siguiente forma: en la sección 2 se presenta el modelo matemático para la distribución de temperatura en la barra. El diseño de un dispositivo electromecánico requerido para adquirir los datos sensados dinámicamente en varios puntos a lo largo de la barra se describen en la sección 3. En el apartado 4 los perfiles analíticos de temperatura son validados con los datos obtenidos a partir del prototipo. Finalmente, en la sección 5 se establecen las conclusiones al presente trabajo y se expone el trabajo a futuro.

2. Modelo Matemático

La propagación del calor en un medio se puede llevar a cabo debido principalmente a los procesos de convección y conducción [13]. En el siguiente trabajo suponemos que el proceso de convección del calor es despreciable en comparación con el proceso de conducción. Consideremos una barra de metal con una sección de área A a la cual se encuentra orientada en la dirección x con longitud L donde $L \ll A$. Esta última restricción junto con la orientación de la barra a lo largo del eje x , nos permite considerar el análisis del problema como un modelo matemático unidimensional para la variable espacial. Si consideramos el intercambio de energía térmica para el Sistema barra-medio, supondremos que habrá flujo de calor en la barra, tal flujo está

gobernado por la Ley de Conservación de la Energía en todo momento, por lo tanto podemos expresarlo en la forma

$$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2}, \quad (1)$$

Donde $k=K_0/cp$ se conoce como la difusividad térmica. La ecuación anterior se conoce como la ecuación del calor [13,14]. Una vez que hemos establecido el modelo matemático, es necesario conocer la distribución de temperatura para distintas posiciones en la barra para cualquier tiempo, con ese fin debemos encontrar la función solución $u=u(x,t)$, donde u corresponde a la temperatura en grados Celsius, x equivale a la posición de la barra medida en metros y t corresponde al tiempo medido en segundos.

Utilizamos el método de separación de variables [15], donde se propone una solución de la forma

$$u(x,t) = X(x)T(t) \quad (2)$$

si unimos ambas soluciones en (1) obtenemos [15,16],

$$u(x,t) = a_0 (b_0 \cos(sx) + b_1 \sin(sx)) e^{-ks^2 t} \quad (3)$$

Para las condiciones de frontera e iniciales, el tiempo inicial se considera en $t = 0$. Nuestra hipótesis es que la barra se mantiene aislada del medio ambiente, excepto en los extremos, donde se asume un flujo de calor no trivial hacia el interior de la barra; esto se logra cuando aislamos completamente a la barra excepto en los extremos de la misma que se encuentran en contacto con el medio. A la izquierda en $x = 0$ está en contacto con una fuente de calor, el otro lado en $x = L$ se mantiene aislado lo cual impone un flujo de calor neto cero en la derecha. Si la barra posee una distribución inicial no uniforme de calor la cual depende de la posición, entonces se establece la condición inicial. Suponemos en general que

$$u(x,0) = f(x) \quad (4)$$

donde $f(x)$ es una función que depende de la posición únicamente. Esta última restricción corresponde a la condición inicial del problema.

Asimismo, necesitamos fijar las condiciones de frontera para ambos extremos. Consideramos el caso estacionario, analizamos la temperatura cuando la barra se encuentra completamente aislada sin

cualquier intercambio de calor con el medio ambiente; lo cual implica que

$$\left. \frac{\partial u}{\partial x} \right|_{x=0} = 0 \quad (5)$$

no existe flujo de calor en el extremo izquierdo

$$\left. \frac{\partial u}{\partial x} \right|_{x=L} = 0 \quad (6)$$

tampoco existe flujo de calor en el extremo derecho. Estas restricciones corresponden a las condiciones de frontera para el problema. Tales condiciones se establecen en el diseño experimental en la forma:

- 1) El calor se aplica a la barra en uno de los extremos hasta que se alcanza la temperatura inicial deseada en $x = 0$; y el extremo $x = L$ ya se encuentra aislado.
- 2) Enseguida se apaga la fuente de calor.
- 3) Cuando la fuente de calor se apaga el extremo $x = 0$ se aísla.

Una vez que aplicamos las condiciones (5), (6) a la solución (3) se obtiene (ver [14])

$$u(x,t) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{n\pi x}{L}\right) e^{k\left(\frac{n\pi}{L}\right)^2 t} \quad (7)$$

la cual se conoce como solución general de la ecuación (1). Como paso final para obtener la solución analítica debemos aplicar la condición inicial, que está definida en la siguiente forma; existe una distribución inicial de la temperatura dentro de la barra. En nuestro análisis suponemos que la distribución de temperatura inicial después de que la barra ha sido completamente aislada, corresponde a la función

$$F_0(x) = T_0 e^{-\frac{2}{5}x} + T_L \quad (8)$$

Donde $F_0(x)$ está definida en el intervalo $0 < x < L$; T_0 es la temperatura registrada en el extremo izquierdo de la barra $x = 0$, T_L es la temperatura registrada en el extremo derecho de la barra en $x = L$. La distribución de temperatura inicial gráficamente está dada como

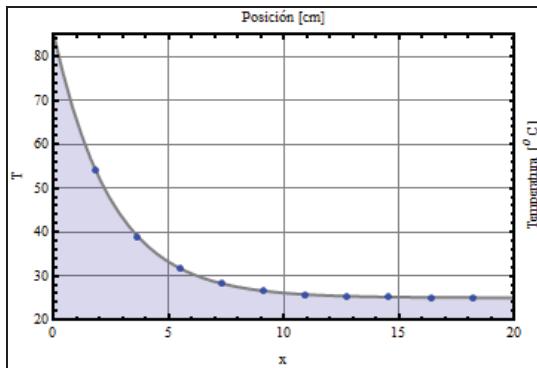


Fig. 1. Distribución de Temperatura Inicial para $F_0(x)$.

Empíricamente cuando consideramos las propiedades particulares de que esta hecho el material, vea la fig. 1. Si aplicamos la condición inicial (8) a (7)

$$u(x,0) = \sum_{n=0}^{\infty} A_n \cos\left(\frac{n\pi x}{L}\right) = F_0(x) \quad (9)$$

Ocurre una expansión para la distribución de temperatura en la forma de una serie de Fourier, como consecuencia debemos calcular los coeficientes A_n necesarios para expresar $F_0(x)$ como una serie de potencias. Después de algunos cálculos, la solución general con condiciones de frontera e inicial es

$$u(x,t) = A_0 + \sum_{n=0}^{\infty} A_n \cos\left(\frac{n\pi x}{L}\right) e^{-k\left(\frac{n\pi}{L}\right)^2 t} \quad (10)$$

$$\text{donde } A_0 = \frac{5}{2} \left(13 - \frac{3}{e^8} \right) \quad A_n = 15 \left(\frac{1 - (-1)^n e^{-8}}{1 + \left(\frac{n\pi}{8} \right)^2} \right) \quad (11)$$

Una vez que hemos establecido el esquema matemático describamos gráficamente este comportamiento.

2.1 Representación Gráfica

Con el fin de tener un mejor entendimiento del modelo se presenta aquí una representación gráfica para la solución mostrada en (10). Con este objetivo consideramos un valor aproximado en la sumatoria con $n = 500$ el cual muestra ser bastante aceptable debido a que para valores mayores la variación en las gráficas es insignificante, además se escogen $T_0 = 60$ °C, $T_L = 25$ °C, $L = 20$ cm cuyos valores de los resultados se mostrarán en la sección siguiente. Consideramos tres gráficas distintas para ilustrar la solución.

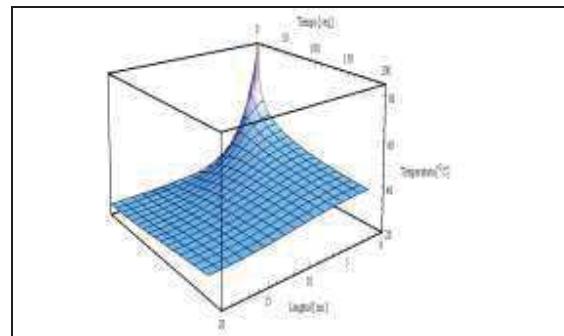


Fig. 2. Representación tridimensional de la evolución del calor en la barra.

La gráfica 2 corresponde a una superficie tridimensional inmersa en un espacio 3D. Los ejes corresponden a la longitud de la barra, la distribución de temperatura y el tiempo medidos en centímetros, grados Celsius y segundos respectivamente. Esta representación nos permite involucrar las tres variables principales simultáneamente, y son las tres variables que aparecen en el modelo de la ecuación el calor, con las restricciones previamente impuestas en (5) y (6).

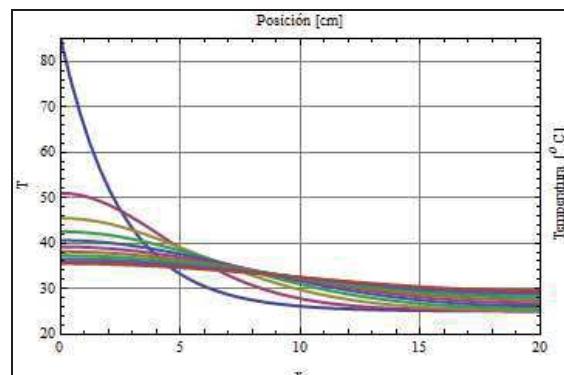


Fig. 3. Modelo de la Ecuación del Calor, Distribución de Temperatura vs. Posición para varios tiempos.

En segundo lugar, la gráfica 3, muestra una representación bidimensional, donde se muestra la temperatura como función de la posición, y superpuestas las gráficas que corresponden a varios tiempos durante la evolución. Cada una de estas curvas que se muestran abajo representan diferentes valores del tiempo.

3. Diseño Experimental

3.1 Consideraciones

Se han tomado en consideración varios aspectos.

Una base de apoyo. Hemos montado unos postes que detienen la barra para que se encuentre aislada de cualquier influencia y desde donde se sujetan los sensores LM35, estos postes tienen un diámetro mínimo para evitar el mayor contacto con la barra y mantenerla aislada.

Circuito electrónico. Consideramos dos tipos de prototipos electrónicos, uno que maneja la alimentación de los sensores y otro para el multiplexor analógico HEF4051BP [17-19], las conexiones son controladas con Arduino desde un núcleo montado en la base. Estos circuitos son instalados en baquelita sobre la base.

Material. Consideramos un prototipo hecho de madera en vez de metal, para evitar posibles errores por la conducción de otros metales y que permitan difusión del calor de la barra. Para las piezas pequeñas usadas en las uniones hemos usado partes de aluminio por su alta maleabilidad.

3.2 Diseño en Rhinoceros

En la figura 4, vista en perspectiva [20,21], mostramos el diseño experimental mecánico como una representación tridimensional. Se aprecian las características del prototipo en conjunto

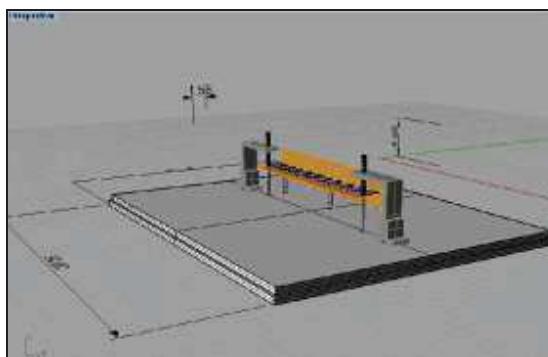


Fig. 4. Diseño en perspectiva.

En la fig. 5 mostramos una imagen real del prototipo, note que los sensores ya han sido instalados en el diseño.



Fig. 5. Imagen del Prototipo Real.

3.3 Validación de Resultados

Las primeras pruebas involucraron mediciones con los sensores LM35, de los cuales se realizaron ajustes sobre todo en el montaje de los mismos sobre la varilla metálica [19]. El proceso en el cual se capturaron los datos para su análisis fue el siguiente:

- 1) Se monta la barra y se sujetta firmemente para comenzar a proporcionar el calor requerido proveniente de una fuente controlada, en este caso la fuente corresponde a un aparato de soldadura el cual proporciona temperaturas mayores a los 100 grados centígrados en la punta del eje metálico.
- 2) Una vez que se ha calentado la barra hasta una temperatura muy por encima de la temperatura ambiente promedio y tomando en cuenta la distribución inicial se colocan los tapones de G10 en los extremos de la barra con el fin de tener un aislamiento completo.
- 3) Se activa una interfaz con Netbeans, como medio de visualización y los datos se envían a un archivo de notas con la información proveniente de Arduino Uno. Básicamente este paso implica la medición directa de la barra y la adquisición de datos que es una de las fases más importantes del experimento.
- 4) Se instaura un procedimiento en Matlab para graficar las mediciones obtenidas en el experimento.
- 5) Finalmente se analizan los resultados a partir de las gráficas obtenidas [20,21].

4. Modelo Ideal vs. Modelo Experimental

4.1 Condición Inicial

La condición inicial en la ecuación (7), se compara con los datos obtenidos en el primer tiempo $t_0 = 0$ segundos, ver gráfica 6, donde se aprecia la similitud entre ambas curvas, la curva que se muestra en color azul corresponde a los datos experimentales, mientras que la curva en color negro corresponde al modelo pautado por la ecuación (7). De esta manera, podemos ver el comportamiento del modelo matemático en comparación con el experimento.

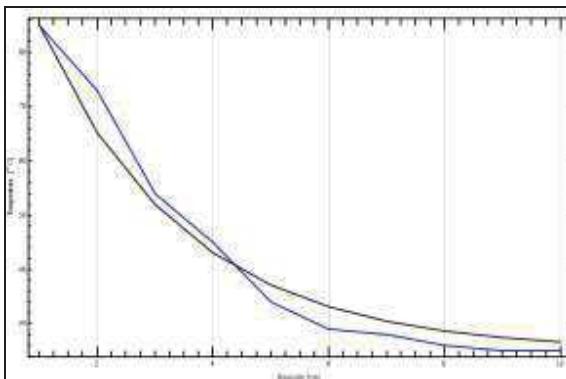


Fig. 6. Condiciones iniciales: ideal vs. real.

Es claro que la diferencia entre ambas es pequeña, por lo que la condición inicial empírica se puede tomar como una buena aproximación al experimento.

4.2 Distribución de Datos Experimental

En este apartado se muestra el resultado crucial y medular del presente trabajo. Los registros de temperatura se recabaron con una duración de 3 minutos. La barra es una barra metálica conductora de Latón de aproximadamente 20 cms. de longitud, 1.0 cms. de ancho y 0.1 cm de espesor. El experimento se desarrolló en un espacio cerrado libre de corrientes de aire sobre una mesa de madera con dimensiones suficientes para colocar el dispositivo experimental.

La gráfica resultante se obtuvo con los datos almacenados de Arduino, dando la gráfica 7, la cual muestra las curvas discretas de 18 mediciones simultáneas de los 10 sensores dispuestos con una distancia de separación de 2 cms. entre cada sensor. Cada una de las mediciones se realizó a intervalos de 10 segundos entre cada una de ellas, la gráfica muestra las distintas curvas en diferentes tonalidades para distinguir la evolución entre ellas, observe que la curva inicial corresponde a la toma de datos inicial (veáse fig. 6) mostrada en color azul, mientras que la curva en rojo más baja corresponde a la condición final para el último tiempo en que se toma la medición.

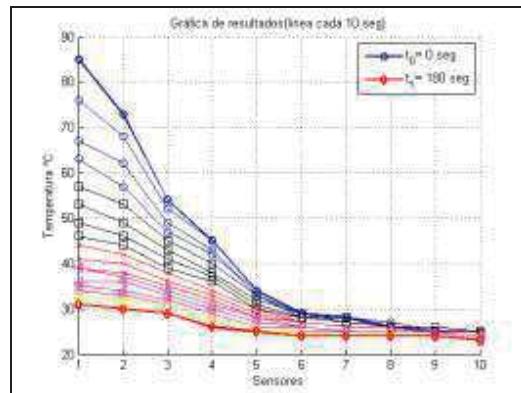


Fig. 7. Mediciones Experimentales para la Distribución del Calor.

En el eje vertical, se grafican los valores de la temperatura. Por ejemplo, se tiene una temperatura máxima de 85° al tiempo $t = 0$ segundos y la temperatura mínima de 23° en el tiempo $t = 180$ segundos para el extremo izquierdo de la barra en $x = 0$, mientras que en el eje de las abscisas encontramos los sensores colocados a 2 cms cada uno, manteniendo la distancia total de la barra de 20 cms. Observe que cada marca circular, cuadrada y triangular en los puntos sobre la gráfica corresponde a un valor de la temperatura en la posición que localiza el sensor a un instante de tiempo respectivamente.

4.3 Modelo vs. Experimento

En este apartado se comparan los resultados obtenidos del análisis de modelo teórico y los resultados obtenidos en la sección anterior. Cabe señalar que la curvas en la fig. 3 se visualizan de una manera continua debido a que el modelo considera al tiempo como continuo sin interrupciones, asimismo nótese que aun cuando las escalas horizontales están en sensores y posición respectivamente se pueden hacer coincidir si se consideran los sensores uniformemente espaciados a intervalos de 2 cms. entre cada uno de ellos, estas diferencias se describen a continuación en la sección siguiente.

4.4 Modelo Corregido

Cabe destacar que el modelo no coincide en su totalidad con los datos obtenidos del experimento. Si bien se han tomado los cuidados y precauciones que exige la técnica, es imposible controlar las variables que influyen en el experimento en su totalidad.

Si consideramos que el entorno que rodea al sistema barra kapton es falso, entonces la diferencia en las gráficas es justificable. Al evaluar los registros de los

sensores se predijo la existencia de una disminución gradual de temperatura; lo cual corresponde a un sistema no completamente aislado.

Teóricamente esto es modelable si en la distribución (ecuación (10)) del modelo se incluye un término de disipación $D(x, t)$ en cuyo caso modifica el modelo en la forma

$$\tilde{u}(x, t) = u(x, t) - D(x, t) \quad (12)$$

donde $u(x, t)$ corresponde a (10), y $\tilde{u}(x, t)$ es la nueva distribución de temperatura con disipación, en la ecuación (12) $D(x, t)$ corresponde a $D(x, t) = a_0 t + 1^\circ C$ con $a_0 = 1^\circ C / 40\text{seg}$.

El resultado de este análisis se ilustra en la gráfica 8, y se confronta con los resultados del experimento expuestos en la gráfica 7.

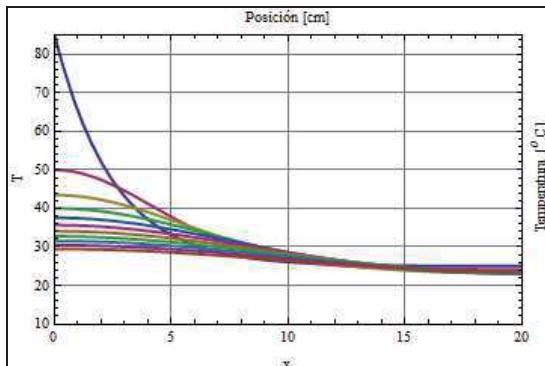


Fig. 8. Modelo de la Ecuación del Calor Modificado.

Es importante señalar que en la función $D(x, t)$ corresponde a un término de disipación, donde se consideran las pérdidas de calor, por un mal aislamiento del conductor del entorno, los sensores no se encuentran en perfecto contacto con la barra, existe una fuga de calor a través de los cables conductores de los sensores, etc. Todas estas pequeñas contribuciones permiten que las mediciones no concuerden con el modelo teórico en el cual se asume un sistema perfectamente aislado del entorno.

4.5 Modelo Experimental Modificado

Aunado a la discusión del apartado anterior y para corroborar aún más la semejanza que existe entre el modelo y el experimento, se expone aquí la modificación que se hace a los datos y se contrasta con el modelo original.

Esta segunda forma consiste en desplazar la temperatura en los datos adquiridos, podemos compensar la disipación que existe si se agregan valores de temperatura a los datos, es decir, en cada

curva que corresponde a una medición de temperatura se agrega calor para compensar dichas pérdidas. A cada una de las curvas experimentales en la fig. 7, se agrega 0.5 grados de corrección de temperatura y se obtiene la forma de la gráfica en la fig. 9. Esta última gráfica se compara directamente con la figura 3.

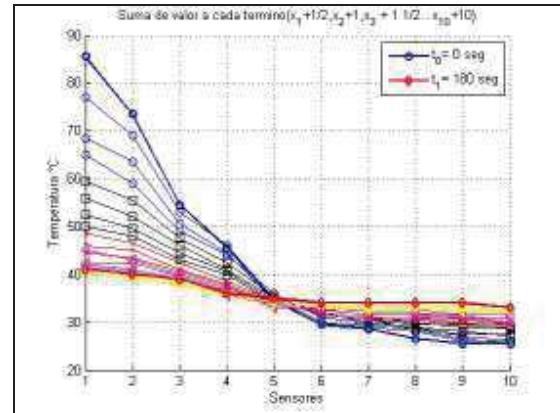


Fig. 9. Modelo corregido con disipación.

De las gráficas 3 y 9, se puede ver claramente que en ambas existe un comportamiento de tendencia hacia el equilibrio en todos los puntos sobre la barra, esto físicamente concuerda con el modelo de propagación del calor en una barra y con la ley de la conservación de la energía. Observe como en (15) el extremo $x = 0$ existe una disminución de temperatura, mientras que en el extremo $x = 20$ cms existe un ligero aumento de la temperatura como lo predice el modelo, hasta alcanzar un valor promedio de temperaturas.

Finalmente para completar el análisis del experimento, vemos el límite hacia el cual tiende la evolución de la temperatura en la barra. La gráfica 10 ha sido tomada a partir de la última curva trazada para el instante $t = 180$ segundos tomada de 7. Observe que la curva tiene a una constante que es la temperatura promedio en la barra, considerando que la barra se encuentra en un estado aislado satisfactorio. Este mismo comportamiento se observa en el modelo bosquejado en la última curva para el mismo tiempo en la figura 10, en el modelo se alcanza la temperatura promedio aproximada de A_0 , dada por la ecuación (14).

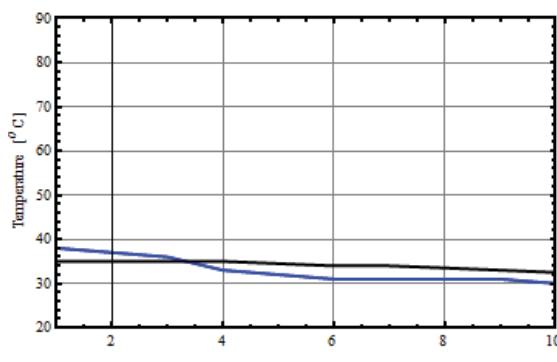


Fig. 10. Medición Temperatura Final.

El comportamiento de tendencia hacia el equilibrio en todos los puntos sobre la barra, físicamente concuerda con el modelo de propagación del calor en una barra y con la ley de la conservación de la energía.

5. Conclusiones

Se ha realizado de manera satisfactoria el diseño y la construcción de un sistema electromecánico para el estudio de la ecuación del calor en una barra. Las mediciones obtenidas a partir del experimento concuerdan con la solución predicha por el modelo matemático de la ecuación del calor.

Se observa que la curva (Fig. 16) tiende a una constante la cual corresponde a la temperatura promedio en la barra, esto es así al considerar a la barra en un estado de aislamiento perfecto. Este es un estado de equilibrio notable sobre la barra completa, la cual corresponde al modelo de propagación de conducción así como a la conservación de la energía. Como trabajo a futuro, hemos considerado dos extensiones al presente estudio: el primero consiste en hacer un diseño incorporando nuevas mejoras con el fin de alcanzar un estado de aislamiento cuasiperfecto, se reconoce la importancia de este hecho para la obtención de los resultados deseados. En segundo lugar, se propone analizar la ecuación del calor para varios tipos de barras con materiales distintos con propiedades distintas.

Referencias

- [1] Bishop R. H. Mechatronic systems, sensors, and actuators., CRC Press. Texas, U.S.A (2008).
- [2] Marro J. Física y Computación. Instituto Carlos I de Física Teórica y Computacional, Facultad de Ciencias, Universidad de Granada, Granada. Revista de Física de la RSEF (2006)
- [3] O'Sullivan D., Igoe T. Physical Computing, Thomson Ed. Boston, USA (2004).
- [4] Golato M. A., Ruiz H., D'Angelo J. F., Aso G. y Paz D. Determinación del coeficiente global de transferencia de calor en secaderos rotativos de cáscara de limón. Rev. Ind. y agrícola de Tucumán, Tucumán, Argentina (2005).
- [5] Patiño L., Espinoza H., Suárez D. Estudio numérico de la convección de calor transitoria en el flujo a través de lechos de empaque, Revista de la Facultad de Ingeniería, Univ. Central de Venezuela. Caracas (2005).
- [6] M. C. R. Bargo et al., Parameter Estimation on a Heat-Conducting Rod: Mathematical Analysis and Numerical Computations. Philippine Journal of Science, Vol. 137, No.1, (2008).
- [7] Zulkifli, N.S.A. Harun, F.K.C. Azahar, N.S. XBee Wireless sensor networks for Heart Rate Monitoring in sport training. Fac. of Electr. Eng. Univ. Teknol. Malaysia, Skudai, Malaysia, (2012)
- [8] National Semiconductor. Datasheet of the LM35 Precision Centigrade Temperature Sensors. November (2000).
- [9] Timmis H. Practical Arduino Engineering. Apress 328 pp. (2011).
- [10] Oxer J., Blemings H., Practical Arduino. Cool Projects for Open Source Hardware Apress. USA (2009).
- [11] McRoberts M., Beginning Arduino Apress. USA (2010).
- [12] Evans B. E. Arduino programming notebook. Creative Commons, San Francisco, California USA, (2007).
- [13] Haberman R., Elementary Applied Partial Differential Equations, U.S.A.: Prentice-Hall Inc., (1987).
- [14] Arfken G. B. and Weber H. J., Mathematical Methods for Physicists, England, Elsevier Inc., (2005).
- [15] Muñoz-Gómez J. A. et al., Ecuaciones Diferenciales Ordinarias, México Universidad de Guadalajara, (2011).
- [16] Zill D. G. Ecuaciones Diferenciales. CENGAGE Learning, México D.F. (2009).
- [17] Atmel Corp. Datasheet of the ATmega48PA, ATmega88PA, ATmega168PA y ATmega328P, 8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash, (2009).
- [18] Arizaga, JA; de la Calleja, J.; Hernández, R.; Bentez, A. Automatic control for laboratory sterilization process based on arduino

- hardware. Proceed. 22nd Int. Conf. on Electrical Communications and Computers (CONIELECOMP) (2012).
- [19] Sarik, J., Kymmissis, I., Lab kits using the Arduino prototyping platform, Frontiers in Education Conference (FIE), 2010 IEEE, pp. T3C-1 - T3C-5, (2010).
- [20] Budynas R. G., Nisbett J. K. Diseño en Ingeniería Mecánica, Ed. McGraw Hill. 1092 pp. 8^a. Ed. (2008)
- [21] Kato Y. Splish: A Visual Programming Environment for Arduino to Accelerate Physical Computing Experiences. Proceed. of The Eight Int. Conference on Creating, Connecting and Collaborating through Computing. La Jolla CA, United States (2010)

Análisis del Comportamiento de un Panel Solar a Través de Scilab

Guerrero Rodríguez Erick Damián, Moreno Castro Luis Fernando, Ramiro Ramiro José y Cruz López Miguel Alberto

Instituto Tecnológico Superior De Zacapoaxtla
División de Ingeniería Mecatrónica
Carretera Acuaco-Zacapoaxtla, Km 8, Col. Totoltepec, Zacapoaxtla, Pue.
Tel: (233) 31 75000.

edgr.dam.11@gmail.com, semirajo@msn.com, migue_cruz15@hotmail.com

Resumen

Se realiza el análisis gráfico del comportamiento de un panel solar, ubicado en la parte superior del edificio de la unidad de prácticas del Instituto Tecnológico Superior de Zacapoaxtla, basándose en Software de código abierto, es decir que no requiere de licencias para su uso. Se realiza la captura de datos que manda el panel solar con ayuda de una placa de desarrollo Arduino, quien será la encargada de recibir y mandar los datos hacia un ordenador (In situ Sistema SCADA), para que posteriormente este los almacene. La parte del análisis gráfico del comportamiento del panel solar, se determina en el Software Scilab, de código abierto, quien es el encargado de tomar el archivo con formato ".txt" creado por la PC y se importa a su entorno de trabajo. De este modo se argumenta el comportamiento del panel a través de una gráfica, y así poder interpretar la comparación entre el comportamiento ideal, el comportamiento real, y el comportamiento adquirido de nuestro panel solar del voltaje en función del clima de la región.

Palabras Clave: Panel Solar, Software Libre, Análisis Grafico, Adquisición de Datos, Sistema SCADA.

1. Introducción

Las energías renovables o alternativas son cordialmente fuentes para conseguir un ahorro energético y con estas obtener la llave para un futuro con energía más limpia, eficaz, seguro, autónomo y amigable con la naturaleza; ya que

En este proyecto se utiliza un panel solar que tiene un voltaje circuito abierto de $V_p = 20$

esto permite contribuir con la reducción de uno de los tantos problemas que hay en el planeta como lo es el calentamiento global.

La fuente más abundante de energía renovable es el sol. La tierra es constantemente abastecida de ENERGIA SOLAR, que es usada por la plantas para crecer y desde hace algún tiempo por las personas para generar electricidad; entre muchos otros usos.

Para lograr convertir la energía solar en energía eléctrica, se ocupan las celdas solares, cuya producción de energía eléctrica depende directamente del nivel de captación de radiación solar que recibe, es por esto que surge la necesidad de encontrar la manera de poder monitorear las variables que recibimos del panel solar y así poder realizar un bosquejo de su comportamiento con respecto al tiempo, y en base a ello, poder compararlo con el clima respectivo de la región y así poder realizar una comparación entre la variable medida y variable real.

Es por ello que para este caso, el análisis gráfico es una manera muy eficiente de comprobar el comportamiento de dicho panel y así poder analizar de manera visual como es su comportamiento y su variación en el tiempo, y así mismo hacer un análisis en base a páginas web de cómo es su respuesta ante la variación climática.

2. Modelo Matemático del Panel Solar.

Volts, para cargar una batería de $V_b = 24$ volts, por lo tanto usaremos un conversor de CD-CD,

o buck converter, que está conformado por un inductor L, y una resistencia RL. El circuito está mostrado en la figura 1 [7].

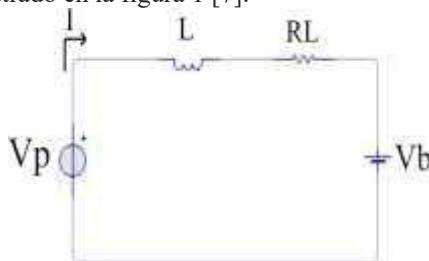


Fig. 1. Circuito de Carga de Batería con el Panel Solar.

En nuestro caso, queremos averiguar la variación de voltaje con respecto del tiempo en que se carga la batería, para esto, hacemos la sumatoria de voltajes.

$$Vp(t) - L \frac{dI(t)}{dt} - i(t) * RL + Vb = 0 \quad (1)$$

Despejando a Vb, para hallar el voltaje en la batería:

$$Vb = Vp(t) - I(t) * R - L \frac{dI(t)}{dt} \quad (2)$$

Con la ecuación 2 ya definida, procedemos a resolvérla, optamos por ocupar la transformada de Laplace para dar solución a esta ecuación:

$$VB(s) = VP(s) - I(s) * R - L * s * I(s) \quad (3)$$

Ajustando y acomodando términos:

$$VB(s) = I(s) * L + \frac{VP(s)}{s+R/L} \quad (4)$$

Y, finalmente dando solución a la ecuación, obtenemos de resultado una función que describe el cambio de voltaje que recibe nuestra batería.

$$Vb = I(t) * L + Vp(t) * e^{-Rt/L} \quad (5)$$

Cabe mencionar que la ecuación 5, es capaz de mostrar los cambios de voltaje que se envían a nuestra batería, es por eso, que tanto el voltaje como la corriente en la ecuación, están con respecto del tiempo, es decir, que varía proporcionalmente a la radiación fotoeléctrica que recibe nuestro Panel.

Para tener un registro de voltajes que se enviaban hacia la batería, se implementó un sistema de control y monitoreo constante de datos, en el que básicamente, una placa captura los datos, y los envía hacia un ordenador, donde posteriormente son analizados gráficamente para poder ser comparados con el cambio climático, y así obtener los valores respectivos de envío hacia la pila, y saber en qué tiempo se cargaría.

3. Adquisición y Procesamiento de Señal.

En primera instancia, se comenzó por hacer el programa para la adquisición de señal, para esto, se utilizó un ordenador, con una placa de desarrollo Arduino utilizada como esclavo para la captura analógica de la señal enviada por el panel solar. La figura 2, muestra el diagrama a bloques del proyecto.



Fig. 2. Diagrama a Bloques del Programa

3.1 Sistema SCADA y elaboración de la HMI.

Para la elaboración del sistema que lograra capturar la señal de salida del Panel solar, se implementó un circuito que redujera el voltaje que nos arrojaba el panel [1], elaborado con un arreglo de resistencias (Figura 3), ya que era

demasiado elevado para la entrada hacia la placa Arduino, quien era la encargada de capturar y enviar los datos hacia nuestro instrumento virtual de medición.

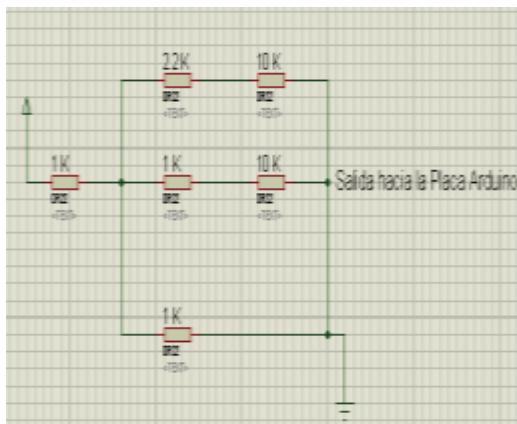


Fig. 3. Circuito de Acoplamiento de Señal

Con los datos ya muestreados analógicamente, La PC los toma [6], y los arregla para que pueda mostrarlos gráfica y numéricamente en pantalla (Figura 4), de manera que los resultados vistos, fueran los mismos que nos proporciona nuestro Panel solar. Pero la PC, además de adquirir y controlar nuestras muestras, se encarga de guardar todos los datos que adquiere de la planta, al mismo tiempo son exportados a un archivo ‘.xlsx’. Cabe mencionar que este programa realizado, es totalmente controlable, y ajustable a partir de su programación [2], ya que se pueden modificar los parámetros de ajuste de entrada y salida de la señal, y también, de establecer fechas y tiempos de inicio y de paro, así como también poder ajustar la gráfica de salida que se muestra.

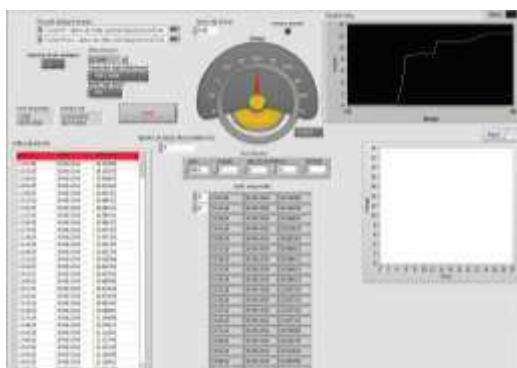


Fig. 4. Pantalla de Adquisición de la PC

4. Exportación e Importación hacia Scilab.

Como ya se dijo antes, la PC es la encargada de adquirir, procesar y controlar los datos, sin embargo, quien se encarga de realizar el análisis correspondiente y mostrar por medio de gráficas dichos resultados, es el software Scilab; un software de Código abierto, muy similar a MatLab, con la misma potencia y las mismas ventajas que MatLab no ofrece, en cambio, Scilab no requiere de una licencia para poder trabajar, ya que al no tener la restricción del código fuente de los autores, hace más fácil y versátil su uso.

Para comprobar su funcionamiento, así como la denominación de software de código abierto y un posible uso futuro, puede descargarse Scilab directamente desde la página del Software: <http://www.scilab.org/>.

La PC, captura, nos arroja y guarda los datos en una hoja de cálculo (.xlsx) con tres columnas o más, en este caso utilizamos: una con la fecha de captura, otra con la hora exacta de captura y la tercera columna que es que nos interesa, nos muestra el valor del voltaje que está emitiendo el Panel solar en tiempo real, dicha hoja de cálculo, se guarda con una nueva extensión (.txt delimitado por tabulación), este nuevo archivo es el que se ocupa para importar datos en la mayoría de Software especializado para el análisis de datos experimentales, y en el caso que respecta a nosotros, ocuparemos Scilab. La figura 5, muestra un ejemplo de las gráficas realizadas con Scilab.

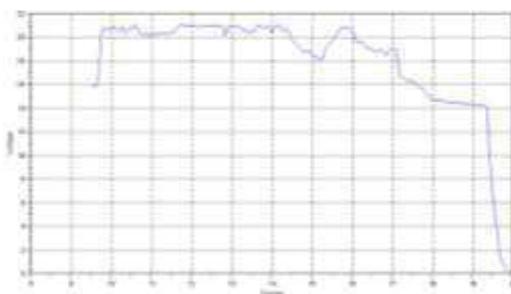
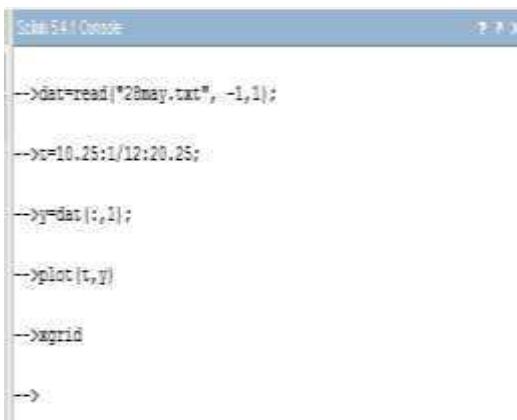


Fig. 5. Grafica hecha con Scilab

Una vez que se tiene este archivo, simplemente se importa al entorno de Scilab mediante comandos, y de la misma manera se

crea una variable que contenga estos datos, ya que se tiene la variable, se crea una nueva donde le indiquemos que columna deseamos ocupar de

este, y así es como tenemos nuestros dos arreglos listos para graficar, uno con las variables, y otro con los tiempos (Figura 6).



```
-->dat=read("28may.txt", -1,1);

-->t=10.25:1/12:20.25;

-->y=dat(:,1);

-->plot(t,y)

-->xgrid

-->
```

Fig. 6. Código para importar y graficar datos en Scilab.

Con lo que respecta a la grafica del clima diario, se creó de igual manera un archivo para exportar y poder tener otros dos arreglos con los cuales graficar la temperatura guardada con respecto al tiempo.

La aplicación ocupada para obtener estos datos climáticos, es una llamada ‘Tiempo’, la cual esta instalada por defecto en el Sistema operativo Windows 8. Esta aplicación obtiene todos estos valores de internet; directamente de la página de BING, la cual es actualizada cada hora.

5. Análisis de Resultados.

De acuerdo con la programación de nuestro instrumento virtual, la placa de desarrollo Arduino, realizaba una lectura de datos cada 5 minutos, que inicia diariamente alrededor de las 7 horas y que termina su muestreo las 20 horas, teniendo a diario una cantidad que oscila entre 110 y 150 muestras de voltajes emitidos por el panel solar. Con lo que su grafica de comportamiento, es bastante fluida, y muestra perfectamente el cambio de voltaje que estuvo recibiendo durante el día.

Sin embargo las graficas que se realizaron donde se grafica el cambio climático con respecto del tiempo, son de cierta manera un tanto menos fluidas, ya que la aplicación ocupada para adquirir los cambios constantes de

temperatura, no es tan constante, es decir que solo nos guarda un muestreo de temperatura cada hora, y al compararlo (Figura 7 y 8), con la que obtuvimos de Scilab, se puede apreciar la relación Voltaje-Temperatura que tiene el panel solar.

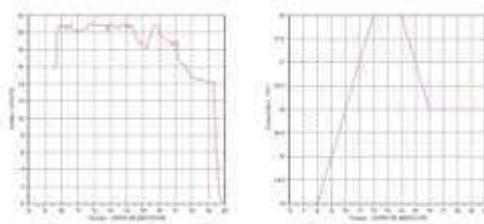


Fig. 7. Comparación de gráficos de Voltaje y de Tiempo en día nublado.

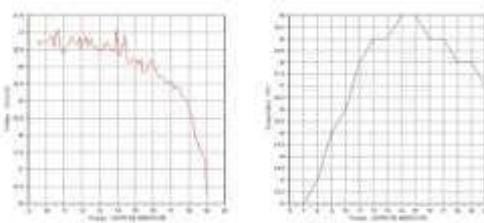


Fig. 8. Comparación de gráficos de Voltaje y de Tiempo en día Soleado

En la tabla 1 y 2, se muestra la comparación entre voltaje y temperatura, donde se puede apreciar la respuesta del panel solar, ante dos situaciones climáticas comunes: un día soleado y uno nublado; y en base a esto, se puede sacar una relación entre el voltaje arrojado por el panel, y la situación climática.

Hora	Temperatura	Voltaje
8 a.m. --	Ascenso de 14 °C a 18 °C	Ascenso de 16 V/DC a 21V/DC
12 p.m. --	Constante de 18 °C	Oscilación entre 20 V/DC y 21 V/DC
2 p.m. --	Descenso de 18 °C a 16 °C	Descenso de 20V/DC a 18V/DC
4 p.m. --	Constante de 16 °C	Descenso de 19V/DC a 0 V/DC
7 p.m. --		

7 p.m.	Oscilación entre 15 °C y 0V/DC	Constante de Adelante 16 °C
---------------	---------------------------------------	------------------------------------

Tabla 1. Relación día Nublado.

Hora	Temperatura	Voltaje
8 a.m. --	Ascenso de 12 °C a 19 °C	Ascenso de 20 V/DC a 21V/DC
12 p.m. -	Ascenso de 19 °C a 20 °C	Oscilación entre 20 V/DC y 21 V/DC
3 p.m. - 7 p.m.	Descenso de 20 °C a 18 °C	Descenso de 20V/DC a 18V/DC
7 p.m. - 8 p.m.	Descenso de 18 °C a 17 °C	Descenso de 18V/DC a 0V/DC
7 p.m. en Adelante	Oscilación entre 17 °C y 16 °C	Constante de 0V/DC

Tabla 2. Relación día Nublado

6. Conclusiones

Como se puede apreciar en las tablas anteriores, existe diferencia entre los dos días, en un día nublado, el panel se mantiene oscilante entre una captura de 14 a 18V/DC, con una temperatura oscilante entre 15 y 18°C; mientras que en un día soleado, se puede percibir un voltaje de 20 y 21 V/DC, a temperaturas de 18 a 20 °C.

Aunque también hay que tomar en cuenta el tiempo, ya que en el transcurso de 12 del mediodía a las 3 de la tarde, es cuando se registra la máxima adquisición de voltaje, aunque no siempre la temperatura más alta. Este parámetro es demasiado importante para nuestra investigación, ya que con esto podemos determinar que la energía que obtenemos no depende de la temperatura, sino de las condiciones del clima, si es que este es soleado, nublado, lluvioso. Para esto debemos analizar más detalladamente para poder dar una explicación certera sobre el comportamiento del voltaje adquirido por la celda, la temperatura y el clima, obteniendo los datos de al menos dos años para realizar una comparación entre estas variables (Dicha investigación se inició en Enero de 2014 y se siguen obteniendo y analizando datos).

Con base a esto, se puede seguir con la siguiente fase del proyecto, que es el control de un seguidor solar, para poder obtener el máximo provecho de un panel solar y de la radiación solar, adquiriendo todo el día un voltaje constante, y tratando de que siempre sea el máximo. Y sobre todo por la región en la que nos encontramos, en la cual el 70% del año se encuentra nublado o con lluvias y con esta investigación podemos determinar si es apto implementar este tipo de energías en nuestro entorno.

Referencias

- [1] Boylestand Robet, L y Nashely, L.: “*Electronica, teoría de circuitos y dispositivos electrónicos*”, Editorial Prentice Hall, 1995.
- [2] Holguín Londoño G. y Pérez Londoño S.: “*Curso Básico de LabView 6i*”, Universidad Tecnológica de Pereira, 2002.
- [3] “*Lenguaje de Programación en Scilab*”, Ceduvirt 2005.
- [4] Pereda Soto I.: “*Celdas Fotovoltaicas en Generación Distribuida*”, Tesis, Pontificia Universidad Autónoma de Chile, 2005
- [5] Lammardo A. y Baritto M.: “*Modelo matemático del comportamiento térmico de un colector solar de placas planas inclinadas para calentamiento de aire*”, Revista de Ingeniería UC, Vol. 7, 2010.
- [6] Ruiz Gutiérrez J.: “*Utilización de LabView para la Visualización y Control de la Plataforma Open Hardware Arduino*”, Creative Commons Attribution 3.0, 2012.
- [7] Dr.Aliaga Zegarra A.: “*Modelamiento y Simulación de un Cargador de Batería para un Seguidor de Máxima Transferencia de Potencia*”, XIX Simposio Peruano de Energía Solar y del Ambiente (XIX- SPES), Puno, 12 - 17.11.2012.

Sistema de Presencia Remota

Charre Ibarra Saida, López Luiz Norberto, Rodríguez García Oscar Alejandro, Palafox Rico Gerardo Javier, Aguilar Castillo Rodolfo Benjamín, Gutiérrez Figueroa Pedro José y Vergara Valora Emmanuel.

Universidad de Colima
Facultad de Ingeniería Electromecánica

1

Resumen

El presente trabajo muestra el diseño mecánico, de control y de comunicación de un prototipo de sistema de presencia remota, siendo parte del desarrollo de un proyecto integrador (PI), como medio de comprobación del uso y aplicación de las nuevas tecnologías y su implementación en nuestra vida cotidiana.

Palabras claves—Telepresencia, teleróbótica, teleoperación, control proporcional, transmisión de movimiento, arduino, app inventor.

1. Introducción

En la Facultad de Ingeniería Electromecánica se desarrollan cada semestre PI donde los estudiantes integran los conocimientos, habilidades, destrezas y aptitudes adquiridas en cada asignatura correspondiente. En el sexto semestre de la carrera de Ingeniero en Mecatrónica, en el periodo Febrero-Agosto 2014 se propuso llevar a cabo el proyecto denominado Sistema de Presencia Remota con el fin de aprender y comprobar el uso y aplicación de las nuevas tecnologías en la vida cotidiana.

Los principios de un sistema de presencia remota, como dispositivo que emula la presencia de un ser humano, iniciaron con el auge de poder realizar una llamada e interactuar con una persona al otro lado de la línea. Los primeros antecedentes de los cuales se tiene conocimiento, sucedieron desde los años sesentas.

Una de las grandes atracciones de la Feria Mundial de New York (New York World's Fair) de 1964, fue el llamado *Picturephone* o *Videophone*, “teléfono de imágenes o videoteléfono”, presentado por AT&T en su pabellón. Con él, las personas podían verse mientras conversaban telefónicamente. En la Feria, las encuestas indicaron que el 60% de la gente opinaba que era “importante” ver a la otra persona con la que hablaba, y consecuentemente AT&T invirtió centenares de millones de dólares en su comercialización. Introdujo el novedoso servicio en Pittsburgh en 1970, y en Chicago en 1971. El *Picturephone* fue finalmente un fracaso; las imágenes no se veían nítidamente y

el servicio era costoso. Posiblemente fue la circunstancia principal que llevó a la no aceptación del invento, la mayoría de las personas no tenían real interés en ver a las personas con las que hablaban. Tal vez no deseaban prestar la atención debida al interlocutor en el otro extremo de la línea y tampoco querían que ello se notara. Además, pareció comprobarse que el video no agregaba mucha información valiosa y útil a la comunicación como para compensar sus desventajas y costo.[1]

Posteriormente, se incrementó la tecnología de video-llamada, dando paso a la telepresencia, el cual en concepto es muy parecido, solo que la comunicación en la telepresencia es mayor que en una videoconferencia, ya que existe interacción en tiempo real.

El término fue acuñado *telepresencia* en un artículo de 1980 por el científico cognitivo estadounidense Marvin Minsky, quien esbozó su visión de una versión adaptada del concepto anterior de teleoperación que se centró en dar a un participante remoto una sensación de estar realmente presente en un lugar diferente. [2]

Así es como nació el sistema de presencia remota o Telebot, por la necesidad de interactuar con otra persona en el otro lado de la línea, pero no solo para platicar, también para realizar actividades u operaciones, tanto del emisor-receptor, como también de forma contraria. En la figura 1 se muestra un ejemplo de un sistema de presencia remota.



Fig. 1. Sistema de presencia remota.

2. Desarrollo

2.1 Metodología

Para cumplir con las especificaciones establecidas para el PI se forman equipos de trabajo entre los estudiantes, donde cada equipo debe resolver el proyecto planteado en el protocolo de investigación que incluye los procedimientos que a continuación se mencionan en la siguiente tabla.

Actividad	Descripción
Planteamiento del problema y definición de objetivos.	Se definen el problema a solucionar y los objetivos a cumplir en función de los lineamientos establecidos previamente por la academia de profesores en el protocolo del PI.
Justificación del proyecto.	En este apartado se presentan las razones por las cuales se realiza el estudio y expresan las motivaciones para desarrollar el proyecto y se responde a la pregunta problema.
Investigación de antecedentes del tema.	Se relacionan los trabajos previos sobre el tema, en determinado periodo de tiempo.
Elaboración del cronograma de trabajo.	Plan de actividades, que muestra en un orden lógico y secuencial la duración del proceso investigativo, en una forma gráfica (Gráfica de Gantt).
Análisis y diseño de componentes mecánicos.	Realizar el diseño estructural y de transmisión con el fin de tener un buen soporte y movimiento motriz.
Diseño e implementación del sistema electrónico.	Diseñar los circuitos de acoplamiento entre la etapa de control y los actuadores.
Elaboración del sistema de control y comunicaciones.	Elaboración de la etapa de control de los servomotores y del sistema de recepción y envío de señales.
Armado del prototipo, prueba y análisis de resultados.	Analís y verificación del comportamiento final del prototipo construido y ensamblado, así como pruebas de comunicación en diferentes espacios.
Análisis de costos	Determinar los requerimientos y costos totales del proyecto, en cada una de las etapas.
Conclusiones	Se describen las principales conclusiones del proyecto desarrollado, derivado del análisis de los resultados.
Reporte final del proyecto y su presentación y defensa oral.	Se realiza el documento que integra el trabajo desarrollado, el cual describe todos los apartados anteriormente mencionados, se lleva a cabo la presentación y defensa oral del prototipo funcionando.

2.2 Sistema Mecánico

Dentro del diseño mecánico está el diseño estructural y de

transmisión, el diseño de la estructura fue tomado de una armadura Warren mostrada en la figura 2. [3]

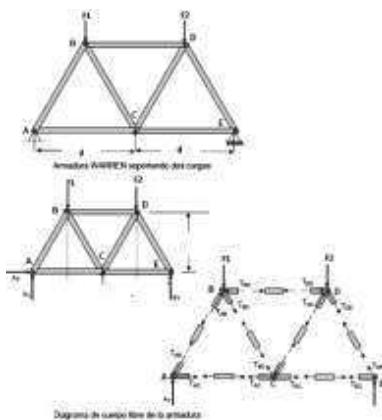


Fig. 2. Armadura Warren.

Como se puede observar en la figura 3, la configuración guarda gran relación con la armadura Warren, este tipo de estructura está diseñada para soportar grandes cargas. [3]



Fig. 3. Sistema de soporte y transmisión.

De igual forma se observa el mecanismo de transmisión en el cual se utilizan dos motores eléctricos para dar movimiento a dos orugas tipo soucy, las cuales presentan una gran ventaja frente a las llantas convencionales, tanto en tracción como en movilidad, ver figura 4. [4]

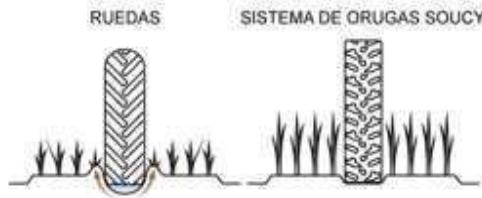


Fig. 4. Ventajas de la oruga soucy.

Para realizar la transmisión tipo oruga se utilizó un sistema de poleas y bandas V, para lo cual se realizaron los cálculos correspondientes de velocidades para transmisiones por correa o banda.

$$\Phi_1 \cdot \eta_1 = \Phi_2 \cdot \eta_2 \quad (1)$$

Donde Φ_1 es el diámetro de la polea motriz y η_1 su velocidad de giro, y Φ_2 , η_2 son el diámetro y a velocidad de la polea conducida.

La relación de transmisión es la relación que existe entre la velocidad de giro del árbol motor y la velocidad del árbol resistente.

$$i = \frac{n_2}{n_1} = \frac{\phi_1}{\phi_2} \quad (2)$$

También se consideró el diseño de transmisión por cadena entre los motores y los ejes motrices de cada par de ruedas, de manera tal que se pudiera aprovechar el torque y reducir la velocidad angular de los motores impulsores.

2.3 Sistema electrónico

Para el diseño electrónico se optó por usar arduino y puentes H para el sentido de giro de los motores, estos puentes H se realizaron con relevadores dados su robustez para soportar grandes corrientes eléctricas. [5] En la figura 5 se muestra un puente H, la cual sirvió para realizar la simulación del circuito.

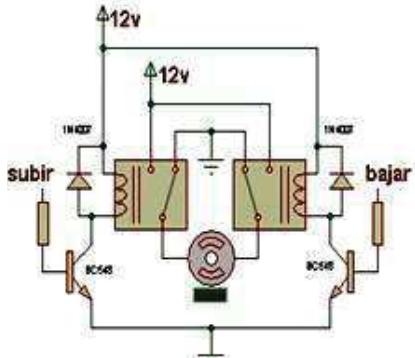


Fig. 5. Puente H.

La simulación del diagrama electrónico se realizó en el programa Proteus Design, para observar el comportamiento y desempeño del circuito antes de su implementación física. La figura 6 representa el circuito realizado, en la cual se tienen dos puentes H para cada motor, ya que cada uno debe realizar el cambio de giro, permitiendo que se mueva en su propio eje para hacer un giro de 360° o girar hacia la izquierda o derecha.

Se observa también la representación de la plataforma de hardware libre arduino uno y el dispositivo bluetooth que permite recibir la información para ejecutar alámbricamente la instrucción a los motores. Las conexiones se indican en cada terminal, de tal forma que el dispositivo bluetooth está conectado en Rx y Tx, en los pines 2 y 3 al arduino y sus pines de tierra y alimentación del dispositivo, las entradas del primer puente H están a los pines 5 y 6, con respecto al segundo puente H sus entradas están a las salidas 9 y 10 del arduino. [6]

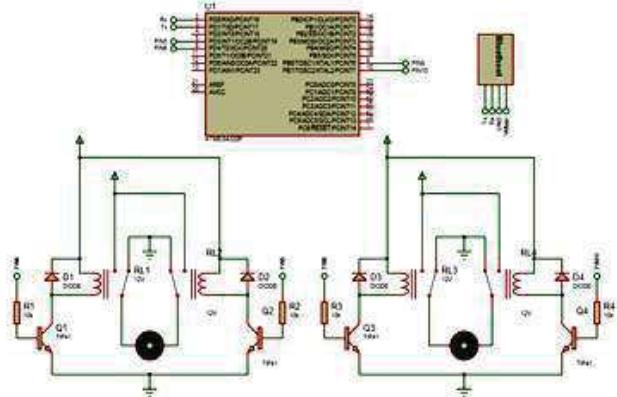


Fig. 6. Diagrama electrónico.

2.4 Sistema de control

El controlador del sistema de presencia remota fue creado en la plataforma Android en un lenguaje de alto nivel. [7]

El controlador funciona de la siguiente manera: un dispositivo Android con el programa de control se conecta a una página web, la cual transfiere los datos enviados por el "control" al dispositivo Android en el Telebot el cual cuenta con el "receptor". Este proceso se ejecuta mediante señal Wi-Fi. Una vez que el "receptor" recibe los datos, los envía mediante señales Bluetooth a un dispositivo Android el cual envía las señales a los motores (puentes H). La figura 7 muestra las etapas y funciones que conforman el sistema.



Fig. 7. Sistema de control.

3 Análisis de resultados

Al realizar las pruebas de cada parte del sistema, se observan diferentes detalles y físicamente el desempeño general. La transmisión de los motores hacia las orugas tipo soucy se efectúa gracias a una cadena acoplada mediante ruedas catarinas en cada uno de los ejes de transmisión, figura 8. [9]



Fig. 8. Sistema de transmisión.

Al activar el motor, la cadena transmite el movimiento hacia el eje el cual rota y a su vez lleva montada una polea, quien transmite el movimiento hacia las bandas para poder trasladar completamente la base. Para lograr la conformación del sistema de traslación con las orugas tipo soucy, se coloca sobre dos ejes horizontales la base completa y a los extremos de estos, las poleas seguidoras del sistema, quienes transmiten el movimiento de la polea impulsora por medio de las bandas, ver figura 9. [4]



Fig. 9. Sistema de traslación.

Para poder tensar el sistema de bandas y cadenas, se implementan mecanismos que faciliten la tensión en las cadenas, se observa en la figura 10, se realizó un pivote de un extremo de la placa que sostiene al motor y del otro extremo una ranura para poder deslizarlo hasta la tensión deseada.



Fig. 10. Tensión de la cadena.

En la tensión de las bandas toda la base donde se encuentran los motores, las horquillas y el eje que transmite el movimiento se levanta con un tornillo, provocando que la banda se tense hasta tener tracción en el suelo, como se observa en la figura 11, el tornillo se aprieta con la tuerca que se encuentra soldada en la base de los motores y se levanta toda la base.



Fig. 11. Tensión de las bandas.

También se realizaron pruebas con la circuitería y el control (arduino y el dispositivo bluetooth), para comprobar su desempeño, figura 12.



Fig. 12. Pruebas arduino y bluetooth.

Se observó un buen desempeño, enviando de forma directa la información del celular al dispositivo y este al arduino, en la figura 13 se puede observar que la programación fue realizada en app inventor para Android .[7], [8].



Fig. 13. Programación.

Una vez realizada la programación para mandar los datos del celular al dispositivo bluetooth, se realizó la parte de envío por medio de internet, creando una página que funciona como servidor, al enviar los datos por internet, el celular los recibe y éste a su vez traduce la información y los envía al dispositivo bluetooth, el arduino los recibe con una instrucción de moverse hacia adelante, izquierda, derecha o hacia atrás, posteriormente el arduino manda la información a los puentes H que mueven a los motores.

Comprobado el funcionamiento de cada componente, se monta todo en la base cuidando la estética, la figura 14 muestra el acomodo de los elementos en el interior de la estructura.



Fig. 14. Control y circuitería.

Finalmente en el centro de la estructura se coloca una placa con un perfil rectangular tubular y en la parte superior se implementa una base metálica para colocar ya sea un celular o una Tablet, en este caso se colocó un celular, mediante el cual se comprobó el correcto funcionamiento del sistema de presencia remota, al interactuar con el dispositivo en distintos espacios de la Facultad y del campus de la Universidad de Colima, figuras 15, 16 y 17.



Fig. 15. Sistema de presencia remota.



Fig. 16. Comprobación de funcionamiento del dispositivo.



Fig. 17. Interacción del dispositivo.

4 Conclusiones

Cumpliendo con los objetivos del proyecto integrador, el prototipo del sistema de presencia remota funciona correctamente conforme a lo establecido. Sin embargo se deben tener en cuenta varios factores que afectan el desempeño del dispositivo, como son el internet, el terreno o superficie donde se encuentre, el medio de control ya sea computadora o tableta y el dispositivo receptor ya sea smartphone o tableta.

Como proyecto futuro se puede rediseñar el sistema de presencia remota para fines más específicos, es decir, implementar innovaciones en el sistema de tracción, para que se desempeñe en áreas o terrenos de difícil acceso, zonas peligrosas, para el hombre, de igual manera se puede pensar para hospitales, teniendo como tarea, alguna intervención médica precisa o simplemente para oficinas o escuelas agregando mayor tecnología para ampliar el desempeño del propio sistema de presencia remota.

5. Referencias

- [1] Q. Digitals, "*Quaderns Digitals*", [En línea]. Available: http://www.quadernsdigitals.net/datos_web/biblioteca/l_746/enLinea/5.htm.
- [2] Minsky M., "*Telepresence*", *OMNI*, june, 45-51, 1980.
- [3] Boylestad R., "*Introducción al análisis de circuitos*", Estado de México: Pearson, 2011.
- [4] McLean W., "*Mecánica Técnica Teoría y problemas resueltos*", Estado de México: McGraw-Hill, 1962.
- [5] Marcus J., "*Guía práctica de circuitos electrónicos*", Estado de México: McGraw-Hill, 2000.
- [6] Norton R., "*Diseño de Maquinaria*", Estado de México: McGraw-Hill interamericana, 1992.
- [7] Parra R., "*Desarrollo fácil y paso a paso de aplicaciones para Android usando MIT App Inventor*", 2013.
- [8] Sierra A., "*Aprendiendo APP Inventor*", 2013.
- [9] Massachusetts Institute of Technology, "*MIT App invento*", Massachusetts Institute of Technology, 2013-2014. [En línea]. Available: <http://appinventor.mit.edu/explore/>. [Último acceso: 07 marzo 2014].
- [10] Quintana E., "*Diseño de transmisión por cadena*", Universidad de antioquia, medellín, 2010.

Procesamiento de Imágenes Aplicado al Control de Robots Manipuladores

Palma Asunción Adilene, Reyes Cortés Fernando, Cid Monjaraz Jaime, Vargas Treviño Aurora y Vergara Limon Sergio.

Benemérita Universidad Autónoma de Puebla-Facultad de Electrónica.
Tel/Fax: +52 222 29 55 00 ext. 7410
PUEBLA, MEXICO

Resumen

En este artículo se aborda el problema de procesamiento de imágenes aplicado al control de robots manipuladores, por tal motivo se elaboran los modelos matemáticos de las configuraciones de los modelos de visión en cámara fija y cámara en mano, para la manipulación de imágenes vistas como señales digitales, y así extraer la información más elemental de ellas, tales como regiones de la imagen. Las aplicaciones del tratamiento de imágenes tratan de dar solución al control de posición en conjunto de un buen modelo matemático de los modelos de visión. Las evaluaciones experimentales realizadas en un robot manipulador de transmisión directa, precisan el comportamiento acertado del control de posición usando la técnica de control servo visual, conocido en la literatura como Visual Servoing.

Palabras clave: Procesamiento de imágenes, visión artificial, control de robots manipuladores, robots experimentales de transmisión directa.

1. Introducción

La Automatización y la Robótica son disciplinas de alto impacto en aplicaciones espaciales, de inteligencia, de manufactura, nucleares, en la medicina, entre otras. De la misma forma el control basado en visión tiene gran número de aplicaciones hoy en día. Visión artificial como un campo de la inteligencia artificial, que mediante las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales. La visión artificial la componen conjuntos de procesos destinados a realizar análisis de imágenes, memorización de la información, procesado e interpretación de resultados.

Las configuraciones que se pueden dar entre el sistema de visión y un robot manipulador son cámara fija y cámara en mano. La configuración cámara fija consiste en colocar la cámara de CCD fija con respecto al sistema de referencia del robot manipulador. Así la cámara captura imágenes del robot y de su medio ambiente. En esta configuración, las imágenes captadas por la cámara son independientes del movimiento del robot manipulador. La figura 1 muestra la configuración cámara fija del sistema de visión junto al robot. La configuración cámara en mano consiste en colocar la cámara CCD en el extremo final del robot manipulador de tal forma que la cámara tendrá el mismo movimiento que tenga el robot manipulador propiciando imágenes de los objetos que se encuentran en el espacio de trabajo del robot manipulador [2]-[7]-[8]-[9].

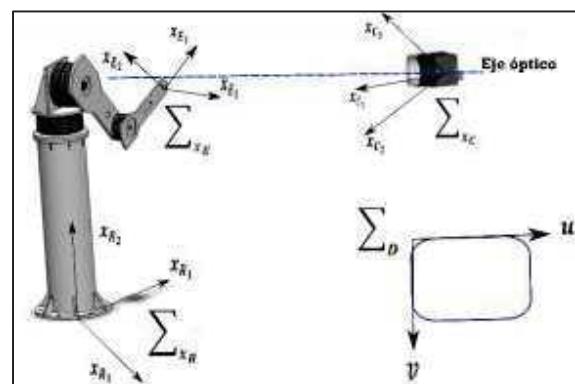


Fig. 1 Modelo de visión en configuración cámara fija

Para la segunda configuración (cámara en mano) consiste en colocar la cámara de video CCD en el extremo final del robot, de tal forma que la cámara tendrá el mismo movimiento que tenga el robot manipulador proporcionando imágenes de los objetos que se encuentran en el espacio de trabajo del robot (ver figura 2). El objetivo de control consiste en

colocar la cámara (extremo final del robot) sobre un objeto deseado.

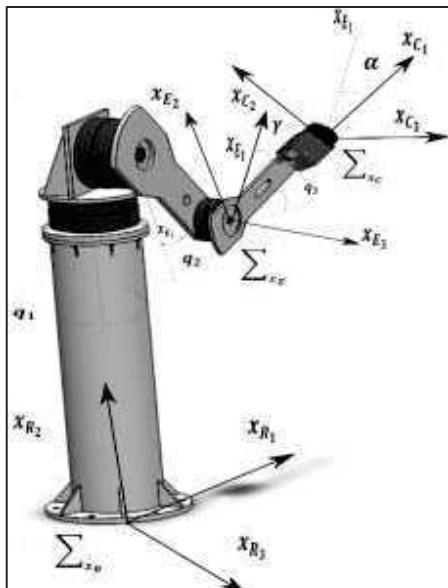


Fig. 2 Modelo de visión en configuración cámara en mano.

2. Visión artificial

La visión por computadora o la visión artificial puede definirse como una técnica de imitar el sentido de visión humano con el fin de extraer información del entorno, algunos investigadores en el área han definido como se muestra a continuación:

- Visión por computadora es una ciencia que desarrolla las bases teóricas y algorítmicas para obtener información sobre el mundo real a partir de una o más imágenes [1].
- La visión artificial o visión por computadora puede ser tratada ya sea como el estudio de transferir a las máquinas el poder de la percepción de la visión humana; también como aquellos métodos que permitan a las máquinas “ver” sólo lo que es necesario para un objetivo específico, es decir, analizar el contenido visual de una escena [2].
- Se define a la visión por computadora, a la disciplina que desarrolla sistemas capaces de interpretar el contenido de escenas naturales [3].
- La visión por computadora, que ha emergido como una disciplina propia basada de manera principal en las matemáticas y ciencias de la computación, consiste en hacer que la computadora vea de manera

artificial su entorno, que involucra parámetros por resolver [4].

El procesamiento de imágenes es el proceso por el cual se toma una imagen en una computadora y se produce una versión modificada de esta. El proceso de análisis de imágenes es del cual a partir de una imagen se obtiene una medición, interpretación o decisión. El esquema típico del análisis de imágenes consiste en cinco etapas:

1. **Adquisición de imagen.** Se obtiene una imagen adecuada del objeto de estudio. Dependiendo de la aplicación la imagen puede ser una fotografía, una radiografía, termografía, entre otras.
2. **Procesamiento.** Con el fin de mejorar la calidad de la imagen obtenida se emplean ciertos filtros digitales que eliminan el ruido en la imagen o bien aumentan el contraste.
3. **Segmentación.** En la que se identifica en una imagen cuáles son los píxeles que pertenecen a un objeto.
4. **Medición.** Es la parte de extracción de características de una imagen, se realiza una medición objetiva de ciertos atributos de interés del objeto de estudio.
5. **Interpretación.** De acuerdo a los valores obtenidos en las mediciones se lleva a cabo una interpretación de la información del objeto de la imagen.

2.1 Sistemas de visión

Los sistemas de visión representan uno de los niveles de complejidad más altos en el procesamiento de imágenes, ya que éstos se pueden adaptar cada uno de los requisitos específicos para diferentes aplicaciones. Las soluciones que ofrecen los sistemas de visión junto con el procesamiento de imágenes junto con algún otro sensor, de ser necesario, son adecuadas para muchos de los requerimientos de la industria, desde la industria automotriz hasta la industria farmacéutica [5]. Los sistemas de visión artificial combinan herramientas de software y hardware para ofrecer sistemas completos de desarrollo para una variedad de tareas de visión artificial.

La calibración de cámaras es un proceso mediante el cual se calculan los parámetros intrínsecos y extrínsecos de la misma a partir de un conjunto de puntos de control, conocidas las coordenadas tridimensionales de esos puntos y midiendo las correspondientes coordenadas de

imagen , en la imagen obtenida con la cámara del sistema de visión [6]. La calibración es el método mediante el cual se estiman los parámetros intrínsecos y extrínsecos de la cámara así como los parámetros del manipulador.

2.2 Descripción de modelos de visión

Para determinar los modelos matemáticos de un robot manipulador consiste en encontrar las matrices proyectivas Euclídeas 3D que hacen la transformación de coordenadas de un sistema de coordenadas relativas al objeto de estudio al sistema de coordenadas del espacio 3D en el que tiene lugar la proyección perspectiva. En el modelo deben incluirse los grados de libertad que posee el manipulador. Comúnmente por cada centro de rotación se establece un sistema de coordenadas. En los modelos debe incluirse factores de cambio de escala, ya que por lo general las unidades en las que trabaja un manipulador son incrementos (grados) y éstos deben convertirse a milímetros o radianes. La conversión de escala se modela en forma lineal.

Para el control de robots manipuladores mediante el empleo de procesar imágenes requiere el conocimiento de los modelos matemáticos que caracterizan el sistema de visión en conjunto con la geometría que involucra el proceso de captación y formación de imágenes por una cámara.

Para el sistema de visión se considera una cámara de video digital con un sensor CCD (charge coupled device) para la formación de imágenes junto con sus elementos de programación y electrónicos necesarios para poder procesar la información visual obtenida de las escenas captadas mediante el uso de una computadora. El modelo de visión se describe por medio de marcos de referencia o sistemas de referencia definidos de tal forma que obedecen a la regla de la mano derecha, lo mismo para los marcos de referencia del sistema robótico.

Los marcos de referencia se ilustran en la figura 1. Se considera $\Sigma_{X_R} = \{X_{R1}, X_{R2}, X_{R3}\}$ un sistema de referencia cartesiano fijo asociado al espacio de trabajo del robot manipulador. El origen del marco de referencia Σ_{X_R} se encuentra localizado en la base del robot manipulador, con sus ejes X_{R1}, X_{R2}, X_{R3} , se seleccionan de manera conveniente. Para este marco coordinado un punto x con respecto a Σ_{X_R} se representa por medio de $x_R = [x_{R1}, x_{R2}, x_{R3}]$.

La cámara de video del sistema de visión tiene asociado un marco de referencia $\Sigma_{X_C} =$

$\{X_{C_1}, X_{C_2}, X_{C_3}\}$, su origen está definido en la intersección del eje óptico con el centro de la lente. El eje X_{C_1}, X_{C_2} definen un plano perpendicular al eje óptico y el eje X_{C_3} , está alineado con el eje óptico de la lente. Un punto x con respecto a Σ_{X_C} se representa por medio de $x_C = [x_{C_1}, x_{C_2}, x_{C_3}]$.

Relativamente la localización entre los marcos de referencia del robot manipulador Σ_{X_R} y el de la cámara CCD Σ_{X_C} está representada por el vector o_R^C , por tal, el plano X_{C_1}, X_{C_2} está localizado a una distancia o_R^C del plano X_{R1}, X_{R2} . Del sensor de la cámara, el plano donde se proyecta la imagen óptica de la escena corresponde al arreglo discreto de elementos fotosensibles del CCD, éste tiene asociado un sistema de referencia de dos dimensiones denotado por $\Sigma_I = \{I_1, I_2\}$ su origen está localizado en el centro geométrico del arreglo del CCD de la cámara. Un punto x_I con respecto a Σ_I se representa por medio de $x_I = [x_{I_1}, x_{I_2}]$, los ejes I_1, I_2 son paralelos y apuntan en la misma dirección de los ejes X_{C_1}, X_{C_2} , respectivamente. El eje X_{C_3} apunta al plano CCD, de manera que un punto x_C frente a la cámara tiene la componente X_{C_3} con signo negativo.

La imagen de la escena captada por el arreglo CCD es digitalizada y transferida hacia la pantalla de un monitor para su muestra en pantalla. Sobre la pantalla del monitor se asocia un marco coordinado de dos dimensiones denotado por $\Sigma_D = \{u, v\}$ cuyo origen se encuentra en la esquina superior de lado izquierdo del monitor con los ejes u, v paralelos a los renglones y columnas de la imagen, respectivamente. El vector u_o, v_o representa el centro de la imagen. Las coordenadas de un punto x sobre la pantalla del monitor de la computadora tiene unidades en pixeles de imagen y es descrito por $x_D = [u, v]$ en vector transpuesto. En la figura 1 se muestra de manera general la ubicación de los marcos de referencia.

El modelo matemático del sistema de visión determinan la proyección de un punto x_R en el espacio de trabajo del robot manipulador sobre un punto en la pantalla del monitor de una computadora x_D denotado por las coordenadas u, v , esto constituye el modelo matemático de visión, el cual está determinado por sus respectivas proyecciones. Las transformaciones de coordenadas cartesianas de la cámara, representa una transformación de cuerpo rígido que se obtiene a partir de la siguiente expresión (1).

$$x_R = O_R^C + R(\alpha, \phi, \theta)_R^C x_C \quad (1)$$

Donde $R(\alpha, \phi, \theta)^C_R$ se denomina matriz de rotación que representa la orientación del sistema de la cámara con respecto al sistema de referencia del robot manipulador, pertenece a la clase de matrices ortogonales que se denomina **SO(3)** [2]. La matriz de rotación considera los ángulos de Euler denotados por α, ϕ, θ , las cuales cumplen con sus respectivas rotaciones en los ejes.

2.3 Pixeles y ángulos

Mediante de un algoritmo capaz de localizar los centroides de los marcadores (círculos negros) ubicados en cada servomotor del robot manipulador (hombro y codo), así también en el extremo final del último eslabón, esto para saber la ubicación exacta de estos elementos y poder diseñar el algoritmo por medio de geometría para el cálculo de los ángulos de cada eslabón, es decir, obtener mediante pixeles el valor de q_1 y q_2 , respectivamente. La figura 3, muestra la posición de los marcadores ubicados físicamente sobre el robot manipulador.



Fig. 3. Marcadores posicionados físicamente en el robot experimental.

Una vez localizados los pixeles en las referencias del robot, mediante un algoritmo de procesamiento de imágenes, el cual tiene como objetivo la localización de los centroides en pixeles de los círculos negros, se tiene la información necesaria para poder determinar geométricamente los valores articulares a los que corresponde esa posición. El análisis geométrico se parte ubicando tres puntos como lo muestra la figura 4. Así calculando los ángulos geométricamente, mediante el uso de conceptos de geometría basado en triángulos

rectángulos y semejanza de triángulos se logran las siguientes ecuaciones para el cálculo de las coordenadas articulares q_1 y q_2 . Las ecuaciones (2) y (8) corresponden a las ecuaciones resultantes de las matemáticas correspondientes para cada coordenada angular.

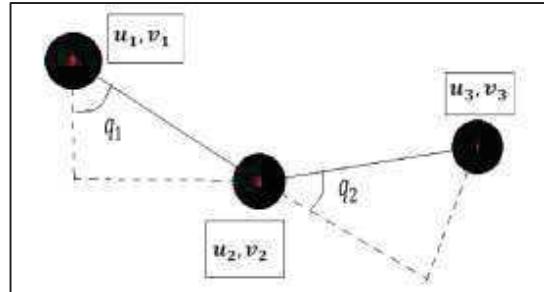


Fig. 4. Ilustración de los marcadores ubicados en el robot manipulador, donde los puntos en rojo determinan las coordenadas en pixeles determinadas por el algoritmo.

Primero, tomando de la figura 4 el triángulo rectángulo tomando como hipotenusa el segmento de línea formado por la coordenada en pixeles u_1 y v_1 unido a u_3 y v_3 , así se obtiene la ecuación (2).

$$q_1 = \tan^{-1} \frac{v_3}{u_3} - \tan^{-1} \frac{l_1 \sin(q_2)}{l_1 + l_2 \cos(q_2)} \quad (2)$$

Para el cálculo de q_2 se propone las ecuaciones (3) a la (7), dando como resultado la ecuación (8).

$$u_3^2 + v_3^2 = (l_1 + l_2 \cos(q_2))^2 + (l_2 \sin(q_2))^2 \quad (3)$$

$$u_3^2 + v_3^2 = l_1^2 + l_2^2 \cos^2(q_2) + 2l_1 l_2 \cos(q_2) + l_2^2 \sin^2(q_2) \quad (4)$$

$$u_3^2 + v_3^2 = l_1^2 + 2l_1 l_2 \cos(q_2) + l_2^2 (\cos^2(q_2) + \sin^2(q_2)) \quad (5)$$

$$u_3^2 + v_3^2 = l_1^2 + l_2^2 + 2l_1 l_2 \cos(q_2) \quad (6)$$

$$\frac{u_3^2 + v_3^2 - l_1^2 - l_2^2}{2l_1 l_2} = \cos(q_2) \quad (7)$$

$$\cos^{-1} \frac{u_3^2 + v_3^2 - l_1^2 - l_2^2}{2l_1 l_2} = q_2 \quad (8)$$

3. Análisis de resultados

Del procesamiento de imágenes se realiza en MATLAB 2012b, el cual cumple con los parámetros de la sección 2 de la captación y manipulación de la información visual. En la figura 5, se muestra el resultado del procesamiento de imágenes aprovechando el toolbox de MATLAB para la captación de las coordenadas en pixeles de la imagen y así obtener el punto coordenado u, v necesario para el modelo matemático de visión.



Fig.5. Proceso de captación para la obtención de la información visual.

Del procesamiento se usa el siguiente código en script de MATLAB:

```
info = imaqhwinfo('winvideo')
info.DeviceInfo
vid=videoinput('winvideo',1,'YUY2_160x120');
src=getselectedsource(vid);
start(vid);
preview(vid);
while(vid.FramesAcquired<100)
I1 = getsnapshot(vid);
Ibinaria=rgb2gray(I1);
x=size(Ibinaria,1);
y=size(Ibinaria,2);
area=0;
c_x=0;
c_y=0;
for i=1:x
for j=1:y
    if Ibinaria(i,j)<=120
Ibinaria(i,j)=0;
        area=area+1;
        c_x=c_x+i;
        c_y=c_y+j;
    else
        Ibinaria(i,j)=255;
    end
end
end
if area ~= 0
```

```
c_x=c_x/area;
c_y=c_y/area;
end
end
```

Para poder verificar que los resultados del procesamiento de imágenes y el modelo obtenido del sistema de visión son correctas se toma un controlador visual para la validación experimental en un robot manipulador de 2 grados de libertad. Para la realización de un controlador con retroalimentación visual, el problema radica en determinar los pares aplicados τ a las articulaciones del robot manipulador para que su extremo final se mueva en forma que la proyección de un objeto estático se encuentre en la posición deseada (u_d, v_d) sobre el plano de la imagen [10]. Hablando de control para la cámara fija consiste en encontrar los pares aplicados de tal forma que la proyección del extremo final alcance la posición deseada sobre el plano de imagen [9]. El error en píxeles de imagen es definido como lo muestra la ecuación (9).

$$\begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} u_d - u \\ v_d - v \end{bmatrix} \quad (9)$$

De la ecuación (2), sacando el límite respecto al tiempo al infinito, tenemos que el vector de error de posición $[\tilde{u}, \tilde{v}]^T$, tiende a cero conforme el tiempo evoluciona, para los propósitos de posición de debe asegurar que el controlador llevará a un error cero. La cámara fue colocada frente al robot manipulador, de manera que coincidió el eje R_3 con el eje C_3 de la cámara web usada para la captación de imágenes. El procesamiento de imágenes para la valoración de las coordenadas del eslabón final fue realizado usando MATLAB donde para el experimento se indicó una posición inicial en pixeles respecto al plano de la cámara de $[u_0, v_0]^T = [123, 192]^T$, es decir, $q_1 = 0^\circ$, $q_2 = -45^\circ$, $q_3 = 120^\circ$ en grados; una posición deseada en pixeles para el experimento fue de $[u_d, v_d]^T = [130, 150]^T$. El error de posición en pixeles está definido por $\tilde{u} = u_d - u$ y $\tilde{v} = v_d - v$. Los resultados experimentales se muestran en las gráficas mostradas en la figura 5 y en la figura 6 donde se aprecian las evoluciones de los parámetros con respecto al tiempo, donde se validan el funcionamiento adecuado del controlador visual. Se muestra en la figura 5 la evolución con respecto al tiempo de los errores en pixeles de cada una de las coordenadas \tilde{u} y \tilde{v} , respectivamente.

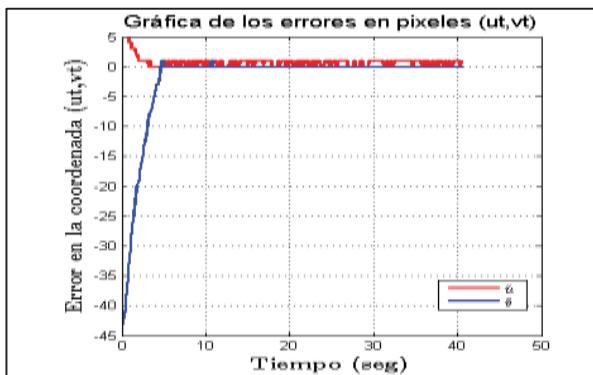


Figura 5. Error de posición \tilde{u} y error de posición \tilde{v} .

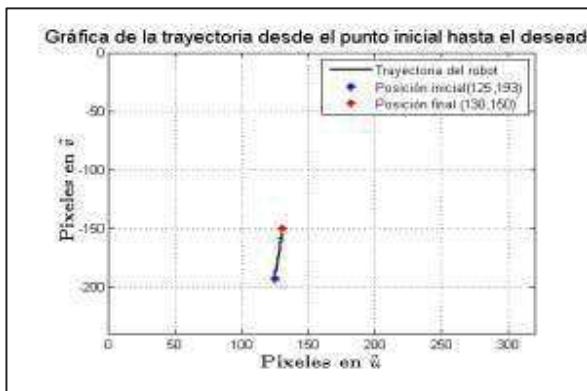


Figura 6. Trayectoria robot sobre el plano de la imagen.

4. Conclusiones

El sistema de visión por computadora al imitar al sentido de visión humano y realizando un procesamiento exacto de imágenes, permite medir sin contacto la posición y orientación del extremo final del robot manipulador. Con los resultados experimentales se valida la aplicación de la visión artificial y el procesamiento de imágenes para una aplicación específica e importante como lo es el control de robots manipuladores, asimismo con los resultados del experimento se hacen válidos y correctos los modelos matemáticos que implica un sistema de visión en sus diferentes configuraciones. De los modelos de visión se enfoca al mapeo de coordenadas cartesianas a coordenadas articulares, para la aplicación a cerrar el lazo de control de los robots manipuladores experimentales con sus respectivas modificaciones para el empleo de las configuraciones cámara fija y cámara en mano. Mediante el uso de herramientas de procesamiento de imágenes se pueden extraer rasgos distintivos de una imagen para poder interpretar y tomar decisiones, tales decisiones no podrían llevar a cabo de manera satisfactoria sino se tuviera la calibración de los sistemas de visión así como los algoritmos que nos determinen la información de coordenadas en pixeles

y en ángulos. La información de visual captada por el sistema de visión es empleada para tener una referencia o un parámetro de decisión para algún proceso o para el uso específico mediante su procesamiento de imágenes.

Referencias

- [1] R. M. Haralick and L. G. Shapiro. “*Computer and robot vision*”, Addison-Wesley Publishing Co., New York, 1992.
- [2] J. Cid & F. Reyes. “*Tesis doctoral: Control de movimiento para un robot de transmisión directa mediante retroalimentación visual*”, UPAEP-México, 2007.
- [3] K. R. Castleman. “*Digital image processing*”, Prentice Hall, Englewood Cliffs, New Jersey, 1996.
- [4] O. Faugeras, Q. T. Luong and T. Papadopoulo. “*The geometry of multiple images: The laws that govern the formation of multiple images of a scenes and some of their applications.*”, The MIT Press, Cambridge, London, 2001.
- [5] E. P. Fuchs. “*Análisis y procesamiento de imágenes*”, http://www.pepperl-fuchs.es/spain/es/classid_1928.htm, España, 2005.
- [6] Y. G. Cid. “*Aplicaciones de la ingeniería electrónica, calibración geométrica de cámaras*”, Escola Politécnica Superior, Departamento de Ciencias Matemáticas e Informática.
- [7] D. Merry. “*Visión por computador*”, Universidad Católica de Chile, Departamento de Ciencias de la Computación, Santiago de Chile, 2004.
- [8] F. Reyes. “*Control de un robot de transmisión directa de dos grados de libertad*”, CICESE-Baja California, México, 1997.
- [9] R. Kelly. “*Robust asymptotical stable visual servoing of planar robots*”, IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 759-766, 1996.
- [10] Hashimoto K., “*Visual Servoing with nonlinear observer*”, Proceedings of the IEEE International Conference on Robotics and Automation, pág.484-489, 1995.

Un Esquema 3D para la Descripción Visual de Gestos Dinámicos

Medrano-Aguilar José Jesús¹, Avilés-Arriaga Héctor Hugo², Gómez-Jáuregui David Antonio³, Herrera-Rivas Hiram⁴ y Nuño-Maganda Marco Aurelio⁵

Universidad Politécnica de Victoria, Parque Científico y Tecnológico de Tamaulipas

Carretera Victoria – Soto la Marina Km 5.5, Cd. Victoria Tamaulipas, México C.P. 87138

¹1229012@upv.edu.mx, ²havilesa@upv.edu.mx, ⁴hherrera@upv.edu.mx, ⁵mnuñom@upv.edu.mx

³Université Paris-Sud / LIMSI-CNRS, Orsay, France

³Gomez-jau@limsi.fr

Resumen— La interacción humano-robot usando gestos es una forma de comunicación natural para instruir robots de servicio. Sin embargo, la naturalidad y efectividad de este tipo de comunicación depende de una adecuada descripción de los gestos. En la literatura reciente es cada vez más común el uso de atributos tridimensionales debido a la actual disponibilidad de sistemas de bajo costo RGB-D. Por lo tanto, en este documento se describe el desarrollo de un sistema de reconocimiento visual de gestos que usa 7 atributos 3D de postura y movimiento capturados con el Microsoft Kinect. Para la localización y seguimiento del usuario se incluye la calibración del Kinect, eliminación del fondo de la imagen, detección del rostro, medidas antropométricas y seguimiento de la mano derecha por color de piel. Se proponen 5 gestos simples que son reconocidos mediante modelos ocultos de Markov. Para los experimentos se creó una base de datos con 2165 gestos obtenidos por una persona. Las pruebas preliminares muestran un 98.3% de clasificación correcta de estos gestos. Los resultados obtenidos sugieren la viabilidad de implementar el sistema visual de reconocimiento de gestos con atributos 3D para instruir a un robot de servicio.

Palabras Clave— Reconocimiento de Gestos, Robots de Servicio, Interacción Humano-Robot.

I. INTRODUCCIÓN

El reconocimiento visual de gestos es una línea de investigación que ha despertado gran interés en los últimos años. En la literatura es bien sabido que el reconocimiento depende de diferentes elementos como son: i) la elección de los atributos para describir los ademanes y ii) el esquema de reconocimiento. Por un lado, gracias a la aparición de videocámaras RGB-D de bajo costo en la literatura se pueden encontrar cada vez más propuestas que utilizan información tridimensional para describir los gestos [1,2]. Por otro lado, los modelos ocultos de Markov son los modelos probabilistas más usados para realizar la clasificación de los gestos.

Por tanto, en este documento se presenta el diseño y desarrollo de un sistema visual de reconocimiento de gestos que utiliza atributos 3D

de un MS Kinect. Esta propuesta es una extensión al trabajo previo descrito en [3] el cual se basa en un sistema monocular RGB y atributos 2D. El reconocimiento de gestos se realiza por medio de modelos ocultos de Markov [4]. Se consideran 5 gestos: a) apuntar a la derecha, b) apuntar a la izquierda, c) saludar, d) acercar y e) detenerse como se puede observar en la Fig. 1.

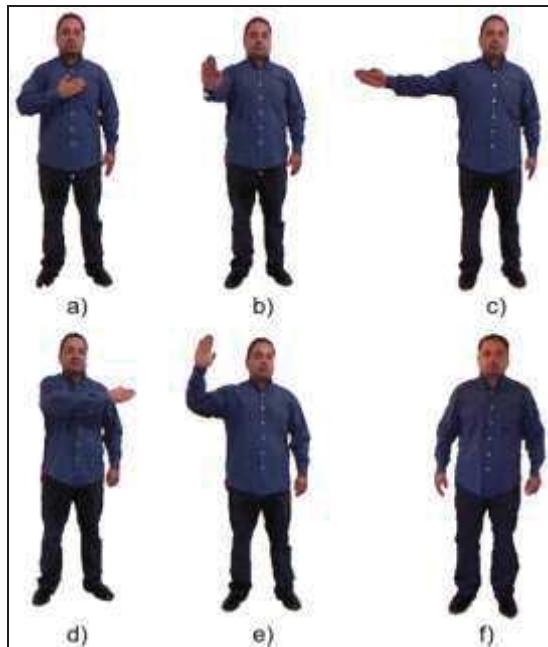


Fig. 1 Gestos propuestos en este trabajo: a) acercar, b) detener, c) derecha, d) izquierda, e) saludar y f) posición de descanso inicial y final de cada gesto.

El sistema incorpora en total 7 atributos de los cuales 3 son de movimiento (desplazamiento) y 4 atributos de postura (posición) de la mano derecha del usuario con respecto a la cabeza y torso. La localización del usuario se realiza por eliminación

del fondo de la imagen, detección del rostro y medidas antropométricas. El seguimiento de la mano derecha del usuario se ejecuta por color de piel. Para la fusión de información de profundidad, de color y reconstrucción de la escena se implementó un sistema de calibración para la cámara RGB-D que en este trabajo corresponde al MS Kinect.

Los gestos descritos anteriormente se eligieron para la instrucción de nuestro robot de servicio SerBot II [5]. Para realizar los experimentos se construyó una base de datos con 2165 gestos ejecutados por una persona. Nuestras pruebas iniciales obtuvieron un 98.3% de clasificación correcta de los gestos. Estos resultados muestran la efectividad del sistema visual de reconocimiento y la viabilidad del sistema para instruir a un robot de servicio.

II. TRABAJOS RELACIONADOS

El uso de dispositivos RGB-D ha aumentado recientemente para el desarrollo de sistemas funcionales de reconocimiento de gestos. En [6] se propone un sistema de reconocimiento de gestos de la mano que con información 3D y adicionalmente lecturas de un acelerómetro que proporciona un seguimiento y pose angular bajo situaciones de oclusión. La nube de puntos de la mano se compara con plantillas para reconocimiento. Palacios et al. [7] proponen reconocer posturas utilizando un cerco convexo para describir la mano. Similar a nuestra propuesta, en [8] se combina información de movimiento y de postura 3D del usuario. Tao et al. [9] describen una combinación de modelos ocultos de Markov multicapa y redes neuronales *backpropagation* para el reconocimiento de gestos continuos usando velocidad angular y aceleración. Estos trabajos utilizan al MS Kinect como dispositivo de entrada. Recientemente, R. Mangera [10] propone el dispositivo Asus Xtion Pro Live para la esqueletización del cuerpo y reconocimiento de 10 posturas tomando como atributos los ángulos de las articulaciones y su posición relativa a la cabeza. El clasificador en este trabajo es *k-means*. Aunque los atributos y métodos de clasificación suelen ser extensiones directas de algoritmos presentes en interfaces de gestos 2D, las propuestas anteriores muestran en general una mejora de la descripción del usuario y un desempeño competitivo en comparación a sus antecesores.

III. METODOLOGÍA

La metodología se divide en 4 partes principales: i) el desarrollo del sistema visual para la captura de gestos, ii) la construcción de la base de datos de gestos, iii) la selección de los atributos para describir los gestos y iv) la definición de los modelos ocultos de Markov para reconocimiento. Cada una de estas etapas se describe a continuación.

3.1. Sistema de análisis visual

Nuestro sistema visual utiliza el MS Kinect para la captura de los gestos. El proceso de análisis visual se divide en tres partes: a) acceso a los datos b) calibración de parámetros intrínsecos y extrínsecos de las cámaras RGB e IR y c) el análisis de las imágenes en la secuencia de video.

a) *Acceso a datos.* El MS Kinect consiste en una cámara de video RGB para luz visible y una cámara IR que detecta patrones de luz infrarroja (emitidos por el mismo dispositivo). Estos sensores producen imágenes bidimensionales en RGB $I_{RGB}(x,y)$ y mapas de profundidad “cruda” $I_{IR}(x,y)$ del ambiente, respectivamente (x, y , indican las coordenadas de cada pixel). Existen diversas librerías como Kinect SDK de Microsoft, OpenNI+SensorKinect o Libfreenect para el acceso a los datos. En este trabajo se utiliza Libfreenect debido a que es una librería de código abierto con ejemplos comprensibles en lenguaje C, con un buen desempeño y no incorpora algoritmos de procesamiento de imágenes, lo cual nos obliga a comprender las técnicas necesarias para implementar funcionalidades disponibles en otras librerías.

b) *Calibración del MS Kinect.* Una vez obtenido el acceso, es necesario calibrar los parámetros intrínsecos de las cámaras y estimar sus coeficientes de distorsión radial y tangencial. Con este propósito se implementó una aplicación basada en OpenCV [11] que requiere múltiples vistas de un tablero de ajedrez y supone un modelo geométrico tipo “pinhole” para la cámara. El tablero es de 4 por 7 esquinas y cuadros de 5cm. La calibración se realiza individualmente para cada cámara y se usan 14 imágenes en cada caso. En particular, en la calibración de la cámara IR se bloqueó el emisor infrarrojo y se usó luz natural para iluminar el tablero [12]. Sin embargo, la imagen del sensor IR suele ser oscura y la detección de algunas esquinas del tablero no es siempre precisa. Nuestra aplicación permite la corrección manual de estos errores al momento de la detección usando un ratón.

La Tabla I muestra un ejemplo de parámetros intrínsecos $\langle f_x, f_y, c_x, c_y \rangle$ obtenidos y redondeados a 3 decimales. Los coeficientes de distorsión $\langle k_1, k_2, p_1, p_2 \rangle$ de la Tabla II se emplean para remover efectos de distorsión en todas las imágenes.

TABLA I. EJEMPLO DE PARÁMETROS INTRÍNSECOS DEL MS KINECT.

	f_x	f_y	c_x	c_y
RGB	5.155e+02	5.161e+02	3.351e+02	2.253e+02
IR	5.968e+02	5.972e+02	3.352e+02	2.267e+02

TABLA II. EJEMPLO DE COEFICIENTES DE DISTORSIÓN DEL MS KINECT.

	k_1	k_2	p_1	p_2
RGB	-1.349e-02	1.310e-01	2.242e-03	-1.837e-05
IR	-1.784e-01	6.239e-01	-5.802e-03	2.001e-03

A partir de los parámetros intrínsecos de la cámara IR y de $I_{IR}(x,y)$ se puede representar la escena como una nube de puntos $P_{IR}(x,y,z) = (X_{xy}, Y_{xy}, Z_{xy})$ en coordenadas “del mundo” y con origen en la cámara IR para la conversión de profundidad cruda a metros. Cada $P_{IR}(x,y,z)$ se obtiene de acuerdo a las siguientes fórmulas:

$$Z_{xyz} = 0.1236 * \tan(d_{xy} / 2842.5 + 1.1863) \quad (1)$$

$$X_{xyz} = \left(\left(x - c^{IR}_x \right) / f^{IR}_x \right) * Z_{xyz} \quad (2)$$

$$Y_{xyz} = \left(\left(y - c^{IR}_y \right) / f^{IR}_y \right) * Z_{xyz} \quad (3)$$

Donde $f^{IR}_x, f^{IR}_y, c^{IR}_x, c^{IR}_y$ son los parámetros intrínsecos del sensor IR. La ecuación 1 fue propuesta en [13] para la conversión de profundidad cruda a metros. En nuestra experiencia los valores válidos de profundidad van de 436 a 1020 (.5m a 5m, aproximadamente).

La nube de puntos es muy útil pero se requiere color para enriquecer la reconstrucción 3D. Libfreenect accede a parámetros de fábrica para el *registro* de imagen (el alineamiento uno-a-uno de los píxeles de profundidad y RGB para combinar 3D y color). No obstante, también reduce el campo visual del dispositivo. De esta forma, se optó por una alineación *vía* software que calcula la rotación R y traslación t entre ambos sensores a partir de imágenes del tablero tomadas por ambas cámaras usando OpenCV [14]. Ejemplos de R y t son:

$$R = \begin{pmatrix} 9.997e-01 & -9.175e-03 & -2.191e-02 \\ 9.078e-03 & 9.999e-01 & -4.495e-03 \\ 2.195e-02 & 4.295e-03 & 9.997e-01 \end{pmatrix} \quad (4)$$

$$t = \begin{pmatrix} 3.553e-02 \\ 6.546e-03 \\ -1.252e-01 \end{pmatrix} \quad (5)$$

Estas matrices representan los parámetros extrínsecos de las cámaras con los cuales se transforma cualquier punto $P_{IR}(x,y,z)$ hacia $P_{RGB}(x,y,z)$, es decir, como si $P_{IR}(x,y,z)$ fuese visto por la cámara RGB en sus propio sistema coordenado de referencia. La siguiente ecuación realiza esta transformación [15]:

$$P_{xyz}^{RGB} = R^T (P_{xyz}^{IR} - t) \quad (6)$$

Una vez calculados los puntos $P_{RGB}(x,y,z)$, éstos son proyectados en el plano de $I_{RGB}(x,y)$ por medio de los parámetros intrínsecos de esta cámara. El resultado son nuevas coordenadas x_{RGB}, y_{RGB} donde puede consultarse en $I_{RGB}(x,y)$ el color correspondiente a $P_{IR}(x,y,z)$. Un ejemplo en OpenGL [16] del resultado de este procedimiento se presenta en la Fig. 2. Este procedimiento y el sistema visual han sido probados satisfactoriamente en particular los modelos del MS Kinect (1414 y 1473).



Fig. 2 Ejemplo de la reconstrucción 3d del ambiente con registro de imagen.

c) *Análisis de imágenes*. Para analizar la escena se realizó la reconstrucción de una tercera imagen bidimensional que contiene el registro de cada pixel de color con información de profundidad $I_{IR}(x,y)$ que a través de las coordenadas xy hace accesible a la información 3D de $P_{IR}(x,y)$. Así, esta nueva imagen permite obtener la forma y color que se usó para la localización y seguimiento del usuario. Posteriormente se procede a la eliminación del fondo utilizando el algoritmo de modelos mixtos gaussianos [17]. Para la eliminación del fondo se utiliza información 3D en lugar de 2D. El Sistema requiere de 15 segundos para el aprendizaje. Una vez eliminado el fondo el usuario puede entrar a la escena dando paso a la detección del rostro por medio de algoritmos de clasificación Haar en cascada [18]. En seguida se estiman las subregiones pertenecientes al torso y a la mano derecha del

usuario en base a la proporción del rostro. Para la localización de la mano se utiliza un algoritmo que trabaja de la siguiente manera: a) por clasificación probabilístico de pixeles de piel y b) por un algoritmo de crecimiento de regiones para la segmentación de la piel. El algoritmo trabaja sobre la subregión de la posición inicial de la mano. De manera similar el seguimiento se hace por medio de una ventana que se actualiza en base al centroide de la mano.

Para la etapa de reconocimiento se utiliza una secuencia de observaciones para cada modelo oculto de Markov calculada por medio del algoritmo hacia-adelante. El cálculo de los parámetros del modelo oculto de Markov se encuentra dado por secuencias de entrenamiento en base al algoritmo Baum-Welch. La consulta de los datos de prueba y entrenamiento se lleva a cabo por medio de una base de datos que almacena los gestos en texto plano. Finalmente se muestra como salida en pantalla el tipo de gesto reconocido. Las partes del sistema propuesto se basan en el diagrama de flujo propuesto por Shrivastava [19]. La Fig. 3 muestra la ejecución del sistema de reconocimiento en el que el usuario ejecuta un gesto. Se indica también la detección del rostro y estimación de subregiones del torso y mano derecha. El rectángulo superior de color rojo indica la detección del rostro del usuario. El recuadro azul claro indica la posición del torso. El recuadro inferior rojo indica la posición inicial de la mano. El recuadro azul marino cercano a la esquina inferior izquierda del cuadro del torso marca la zona inicial de búsqueda de la mano. El recuadro amarillo es la región de búsqueda de la mano en cada imagen. Finalmente, el rectángulo verde es el resultado de la segmentación de la mano. Este último rectángulo (rectángulo verde) sirve para la extracción de la información "cruda" de los gestos que se describe en la siguiente sección.

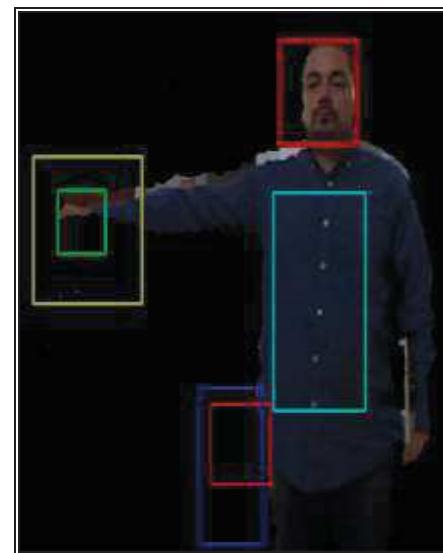


Fig. 3 Ejemplo del sistema visual para el seguimiento de la mano derecha del usuario

3.2. Base de datos

Para este trabajo se generó una base de datos con 5 gestos naturales y un total de 3165 muestras. La ejecución de los gestos fueron realizadas por uno de los autores. Cada muestra está compuesta por el número T de observaciones que componen al gesto (donde $4 < T < 15$) y de las observaciones en secuencia. Cada observación consiste en: a) las coordenadas (x_m, y_m, z_m) de las esquinas superior e inferior del rectángulo que segmenta la mano derecha, b) las coordenadas (x_t, y_t, z_t) de las esquinas superior e inferior del rectángulo que segmenta al torso y c) las coordenadas (x_r, y_r, z_r) del rostro.

Los datos se grabaron en archivos de texto plano. Los gestos comienzan y terminan siempre con el brazo del usuario en una posición de descanso. Nuestro participante ejecutó todos los gestos de frente a la cámara a una distancia de 2 metros aproximadamente. La altura del MS Kinect fue de 1.50 metros en base a la posición cercana a la que se encuentra el MS Kinect en SerBot II. Las observaciones se registraron cada 4 imágenes a una velocidad aproximada de 20 imágenes por segundo.

3.3. Descripción de los atributos

De la información "cruda" descrita en la sección anterior se obtienen 3 atributos de movimiento y 4 de postura para describir a nuestros gestos. Los atributos de movimiento son Δx , Δy y Δz que describen cambios en la posición XYZ de la mano y nos permiten describir el desplazamiento de la mano en el espacio 3D reconstruido de la escena. Cada una de estas variables pueden tomar uno de 3 valores $\{(+), (-), (0)\}$ que

indican incremento, decremento ó no cambio dependiendo de la posición de la mano en la imagen inmediata anterior en la secuencia, respectivamente. Por ejemplo, si el usuario mueve su mano hacia arriba entonces $\Delta y = (+)$; si la mano se mueve hacia abajo, $\Delta y = (-)$ y si no hay movimiento vertical, $\Delta y = (0)$. Los atributos de postura son *forma*, *arriba*, *derecha* y *torso*. Éstos representan la apariencia de la mano y relaciones espaciales entre la mano y otras partes del cuerpo como el rostro y el pecho. La apariencia de la mano es representada por el atributo *forma* y puede tomar 3 valores: (+) si la mano muestra la palma, (-) si la mano está en posición horizontal ó (0) si la mano está inclinada a la derecha o a la izquierda. Esta información es fácilmente deducida por comparación de la longitud de los lados del rectángulo que segmenta a la mano. El atributo *derecha* indica si la mano está a la derecha de la cabeza; *arriba* indica si la mano está arriba de la cabeza y *torso* describe si la mano está sobre el torso del usuario. Las variables de postura son binarias $\{V, F\}$ y toman su valor dependiendo de si la condición correspondiente se satisface o no. El número posible de observaciones diferentes es 648. Así, un gesto es una secuencia de vectores de estos 7 atributos en el siguiente orden: $\langle \Delta x, \Delta y, \Delta z, forma, arriba, derecha, torso \rangle$.

3.4. Modelos Ocultos de Markov para reconocimiento

Para el reconocimiento se utilizan los modelos ocultos de Markov (MOM). Un MOM λ describe la evolución de un proceso a través de un conjunto de estados identificados en cada tiempo t por una variable S_t y un conjunto de transiciones posibles entre ellos. Sin embargo, el estado S_t no es directamente identificable, es decir, está "oculto" y sólo es accesible a través de las observaciones O_t que se obtienen del sistema (en nuestro caso, los valores de atributos descritos en la sección anterior).

Un modelo oculto de Markov sigue el siguiente modelo de probabilidad:

$$P(O|\lambda) = \sum_{\forall S} P(S_1|\lambda) \prod_{t=1}^{T-1} P(S_{t+1}|S_t, \lambda) \prod_{t=1}^{T-1} P(O|S_t, \lambda) \quad (7)$$

donde $P(S_1|\lambda)$ es la distribución de probabilidad inicial de los estados, $P(O_t|S_t, \lambda)$ es la función de probabilidad de las observaciones dados los estados y $P(S_{t+1}|S_t, \lambda)$ es la distribución de probabilidad de las transiciones entre los estados a través del tiempo. La topología "hacia adelante" de 3 estados del modelo utilizado para describir cada gesto se muestra en la Fig. 4. La construcción de un clasificador de gestos requiere construir un MOM λ_i para cada gesto. Así, para el entrenamiento de cada uno de los modelos ocultos de Markov se utiliza una versión logarítmica del algoritmo Baum-Welch. En este trabajo se modificó la implementación de modelos ocultos de Markov de T. Kanugo [20] para múltiples observaciones. Para reconocimiento, un criterio de máxima-verosimilitud se utiliza para seleccionar el modelo λ_i que maximice $P(O | \lambda_i)$.

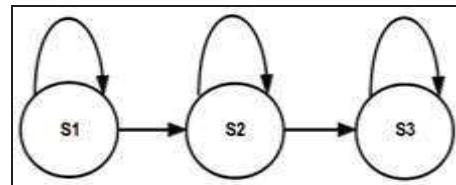


Fig. 4 Topología lineal de 3 estados o clases utilizada en los modelos ocultos de Markov

IV. EXPERIMENTOS Y RESULTADOS

4.1. Condiciones experimentales

Para la experimentación se seleccionaron aleatoriamente 200 muestras de cada tipo de gesto para entrenamiento de los modelos ocultos de Markov como se puede observar en la Tabla III. El resto de las muestras fueron utilizadas para reconocimiento. Este proceso fue realizado una única vez. Sin embargo, a diferencia de otras propuestas similares en las cuales son comunes pruebas de validación cruzada de K repeticiones con 90% de los datos disponibles para entrenamiento, en nuestra experimentación se utilizan más muestras de prueba. Los experimentos fueron ejecutados usando una laptop con un procesador Intel Core 7 y 8Gb de memoria RAM con SO Ubuntu 12.04.

TABLA III. CLASIFICACIÓN DE LOS GESTOS, TOTAL DE MUESTRAS DE CADA GESTO Y CANTIDAD DE MUESTRAS SELECCIONADAS PARA ENTRENAMIENTO Y PARA PRUEBA.

Gesto	Número de muestras		
	Datos de Entrenamiento	Datos de prueba	Total de Muestras
Acercar	200	382	582
Parar	200	441	641
Derecha	200	454	654
Izquierda	200	405	605
Saludar	200	483	683

4.2. Resultados y discusión

Los resultados de reconocimiento se describen en la Tabla IV. El número total de muestras por gesto se muestra en la Tabla III. Los datos de prueba son resultado de la resta de las 200 muestras del total de cada tipo de gesto. Se muestra el total de resultados correctos y de error, así como el porcentaje correcto de reconocimiento obtenido de cada gesto en particular.

TABLA IV. RESULTADOS DE CLASIFICACIÓN DE GESTOS.

Gesto	Datos de prueba	Correctos	Error	Reconocimiento (%)
Acercar	382	377	5	98.69
Parar	441	423	18	95.91
Derecha	454	448	6	98.67
Izquierda	405	405	0	100.00
Saludar	483	475	8	98.34

El promedio ponderado de reconocimiento obtenido es de un 98.3%. Este resultado es muy similar al obtenido en el trabajo de Zhong que obtiene un porcentaje de 96.67% [21]. Los resultados de reconocimiento se pueden observar de forma más detallada en la matriz de confusión de la Tabla V. Como se puede observar el gesto de izquierda no tuvo ningún error de reconocimiento lo cual puede resultar peculiar debido a que en la ejecución del gesto la mano pasa por el recuadro del torso, lo cual se podría pensar que se confunda con el gesto de acercar que no es el caso. Por otra parte el gesto de parar tuvo más errores de reconocimiento con respecto a todos los demás, esto pudiera deberse a los movimientos de trayectorias similares a las de otros gestos. De esta manera, observando estos resultados, en general se puede decir que la clasificación de gestos no presenta una deficiencia considerable. Sin embargo esta investigación nos conduce a la necesidad de encontrar métodos alternativos de relajar el sistema para el reconocimiento de gestos para diferentes usuarios.

TABLA V. MATRIZ DE CONFUSIÓN.

Gesto	Acercar	Parar	Derecha	Izquierda	Saludar	Total
Acercar	377	2	0	3	0	382
Parar	3	423	4	2	9	441
Derecha	0	3	448	1	2	454
Izquierda	0	0	0	405	0	405
Saludar	0	3	5	0	475	483

V. CONCLUSIONES Y TRABAJO FUTURO

En este documento se detalló el procedimiento del desarrollo de un sistema visual de reconocimiento de gestos de un usuario utilizando diversas técnicas de visión por computadora con el apoyo del MS Kinect. Este sistema nos proporciona una base sólida hacia desarrollos e implementaciones más complejas de diversos algoritmos de reconocimiento visual por computadora en ambientes tridimensionales. Los experimentos arrojan resultados con porcentajes de efectividad total de reconocimiento de gestos de un 98.3%, lo cual se considera bueno respecto a otros sistemas similares que se encuentran en la literatura los cuales tienen una efectividad total de reconocimiento de gestos de un 96.67%.

Como trabajo futuro se propone también la realización de un modelo geométrico en 3D de la cabeza, brazo y torso del usuario basado en paralelepípedos, para mejorar la localización. También se propone el uso de filtros de partículas para el seguimiento de la mano, así como la integración de más gestos de reconocimiento y realización de experimentos del sistema con distintos usuarios y la implementación del sistema en un robot de servicio.

Referencias

- [1] Rimkus, K., Bukis, A., Lipnickas, A., & Sinkevicius, S, “3D human hand motion recognition system” In *Human System. Interaction (HSI), 2013 The 6th International Conference on*, Págs 180-183, 2013.
- [2] Sigal, L., Fleet, D. J., Troje, N. F., & Livne, M., “Human attributes from 3d pose tracking” In *Computer Vision–ECCV 2010*, Págs 243-257, 2010.
- [3] Avilés–Arriaga H.H., Sucar–Sucar L.E., Mendoza–Durán C.E., Pineda–Cortés L.A. , “A Comparison of Dynamic Naive Bayesian Classifiers and Hidden Markov Models for Gesture Recognition”, J. Appl. Res. Technol. Vol. 9 No.1, México, Abr, 2011.
- [4] Elmezain, M., Al-Hamadi, A., & Michaelis, B, “Hand trajectory-based gesture spotting and recognition using HMM”, In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, Págs 3577-3580, 2009.
- [5] Camberos A., Tovar C., Medrano J., Arriaga L, Avilés H., “Avances en el Desarrollo y Construcción del Robot de Servicio SerBot II”, *Proceedings of the 12vo. Congreso Nacional de Mecatrónica. Por aparecer*.
- [6] P. Trindade, J. Lobo, J. P. Barreto, Hand gesture recognition using color and depth images enhanced with hand angular pose data, Proc. of the 2012 IEEE International Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI), Págs. 71-76. 2012.
- [7] Palacios J. M., Sagües C., Montijano E. and Llorente S., “Human-Computer Interaction Based on Hand Gestures Using RGB-D Sensors”, Sensors, Volumen 13, Número 9 Págs. 11842-11860. 2013.
- [8] Liang B, Zheng L., Shah S.A.A., Bennamoun M., Boussaid F., El-Sallam A.A., Urschler M., Bornik A., Donoser M., Babahajian P., “Three dimensional motion trail model for gesture recognition”, IEEE International Conference on Computer Vision (ICCV), Págs. 684-69. 2013.
- [9] Tao C., Liu G., “Human Robot Interaction using gesture recognition based on RGB-D Sensor”, Journal of Convergence Information Technology (JCT), Volumen 8, Número 11, Págs. 448-454, 2013.
- [10] Mangera R., “Static gesture recognition using features extracted from skeletal data”, Twenty-Fourth Annual Symposium of the Pattern Recognition Association of South Africa, 2013.
- [11] Bradski G., Kaehler A., Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008.
- [12] ROS.org, Intrinsic calibration of the Kinect cameras. Disponible en:http://wiki.ros.org/openni_launch/Tutorials/IntrinsicCalibration. Última visita: 27 de julio del 2014.
- [13] Magnenat S. Disponible en:
https://groups.google.com/forum/#!topic/openkinect/AxNRhG_TPHg. Última visita: 14 de julio del 2014.
- [14] N. Burrus, Kinect Calibration. Disponible en:
<http://nicolas.burrus.name/index.php/Main/HomePag>. Última visita: 14 de julio del 2014.
- [15] Trindade P., Lobo J., Barreto J. P., “Hand gesture recognition using color and depth images enhanced with hand angular pose data”, Proc. of the 2012 IEEE International Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI), Págs. 71-76. 2012.
- [16] OpenGL. Disponible en: <http://www.opengl.org/>. Última visita: 14 de julio del 2014.
- [17] Yihsin H., Takao N., Toru Y., Eri, s-S., Norio T., “A Hand Gesture Recognition System Based on GMM Method for Human-Robot Interface”, Robot, Vision and Signal Processing (RVSP), 2013 Second International Conference on , Págs 291 - 294 . 2013.
- [18] Nishimura, J. ; Kuroda, T. “Versatile Recognition Using Haar-Like Feature and Cascaded Classifier”, Sensors Journal, IEEE, Págs 942 - 951. 2010.
- [19] Shrivastava R. “A hidden Markov model based dynamic hand gesture recognition system using OpenCV”, Proceedings of the 2013 3rd IEEE International Advance Computing Conference. IACC 2013, Págs. 947 – 950. 2013.
- [20] Kanugo T., “UMDHMM: Hidden Markov Model Toolkit, Extended Finite State Models of Language”, A. Kornai (editor), Cambridge University Press. 1999. Disponible en: <http://www.kanugo.com/software/software.html>. Última visita: 27 de julio del 2014.
- [21] Zhong Y., Yi L., Weidong C., Yang Z., “Dynamic hand gesture recognition using hidden Markov models”, Proceedings of 2012 7th International Conference on Computer Science and Education. ICCSE 2012, Págs 360 – 365. 2012.

Agradecimientos

Este trabajo está financiado parcialmente por el proyecto PROMEP /103.5/12/3620

Diseño de un Sistema de Control para Generación de Energía Eléctrica en Bicicletas Horizontales

Méndez Zapata Elías y Cano Corona Ariana

Universidad Politécnica de Tlaxcala, Programa Educativo de Ingeniería Mecatrónica, Av. Universidad Politécnica No.1 Tepeyanco, Tlaxcala, México C.P. 90180

Resumen

En el presente trabajo de investigación se presenta el diseño e implementación de un sistema de control basado en un microcontrolador de la familia Arduino para el acondicionamiento, procesamiento y control de señales provenientes de un alternador para generar y almacenar energía eléctrica. Aplicando principalmente esta energía a la carga de celulares y tablets así como la generación de estímulos visuales, todo esto justificado con el fin de aprovechar la energía física realizada al momento de hacer ejercicio con una bicicleta horizontal estática. En el desarrollo se muestra que el sistema es de fácil acoplamiento a diferentes aparatos de ejercicio ya que es robusto, funcional y escalable. Finalmente, el sistema de control fue implementado en un prototipo de bicicleta horizontal que también fue diseñada después de un estudio ergonómico y análisis mecánico.

Palabras clave: Control, procesamiento, tarjeta, generación.

1. Introducción

En la actualidad el tema de energías alternativas y renovables para la generación de electricidad está en pleno desarrollo, en México es necesario el trabajo de concientizar el uso de la energía renovable ya que el valor de la electricidad para los usuarios está en función de los costos de producción. La mayoría de los países hoy aún basan su crecimiento económico en los combustibles fósiles tales como el gas y el petróleo entre otras como si fueran inagotables por lo que las transiciones energéticas corresponden a las generaciones presentes, México se encuentra entre estos países y hoy en día no existen productos que comercialicen la iniciativa propuesta en el país.

Se puede sacar provecho del gasto energético como energía renovable, de algunos aparatos de ejercicio, ya que actualmente los aparatos de ejercicio se utilizan para quemar calorías, sin recuperar la energía cinética que a su vez puede alimentar un generador de electricidad.

El proceso para colectar la energía es bastante simple. En el medio del gimnasio existe una “gran pila” que almacena toda la energía conseguida durante el día por todos los ejercicios que realizan las personas que hacen uso del gimnasio de la plaza y que luego se reparte, dependiendo del caso, entre sus instalaciones, luminaria del parque o hacia la comunidad. Esta energía acumulada se produce por dispositivos que transforman la energía cinética (fuerza energética provocada por el movimiento de un objeto) en corriente de luz.

El proceso de conversión de energía mecánica a eléctrica se puede lograr a partir de las revoluciones generadas en el eje de los pedales de una bicicleta horizontal fija, utilizadas por personas que transmitan esas revoluciones para mover un generador eléctrico y producir electricidad [1].

Este proyecto aportará importantes beneficios sociales y culturales. Ya que además de fomentar el deporte y promover la mejora de la salud física, se producirá energía eléctrica con el movimiento mecánico ejercido por los usuarios. Esto traerá como beneficios el aprovechamiento de esta energía en la carga de dispositivos móviles.

El proceso de generación de energía consiste en almacenar esta, durante por el ejercicio realizado y utilizarla en efectos visuales y carga de dispositivos móviles con conexión USB. La energía acumulada se produce por una transmisión por cadenas, que transforma la energía cinética en corriente para utilizarla en las aplicaciones anteriormente mencionadas.

Los alcances que este proyecto tiene pueden replicarse a gran escala proponiendo su uso en lugares públicos concurridos (escuelas, parques, plazas comerciales, estadios, etc.). De igual forma este proyecto aportaría beneficios a las comunidades con poco o nulo acceso a la red eléctrica comercial promoviendo de esta forma un ambiente más sustentable con menor consumo de electricidad y así reducir los gastos de consumo familiar.

2. Desarrollo del Sistema

El desarrollo del proyecto se subdividió en cuatro etapas las cuales se muestran en la figura 1:



Fig. 1 Diagrama secuencial del prototipo

2.1 Diseño ergonómico y mecánico

La mayoría de los aparatos y dispositivos que empleamos en la vida cotidiana requieren energía para funcionar. Esta energía empleada por los dispositivos es provista por alguna fuente: combustibles, electricidad, luz solar, entre otras.

En los distintos sistemas que hemos mencionado, tanto los naturales como los creados por el ser humano, se observan cambios o transformaciones, que serían imposibles sin el aporte de energía. Es por esta razón principalmente que se trabaja sobre un diseño de una bicicleta horizontal para transformar la energía mecánica en energía eléctrica ya que es la más importante desde el punto de vista tecnológico permitiendo tener luz artificial por las noches y permite poner en funcionamiento distintos aparatos de uso doméstico o público, además de la realización de actividades humanas a gran escala, como la industria, el transporte y las comunicaciones.

Durante la fase de investigación de la ergonomía que debía tener la bicicleta horizontal se hicieron pruebas con diferentes personas para medir las distancias ideales que debería tener para que fuera cómoda y pudieran realizar ejercicio por un tiempo considerable y que no corrieran el riesgo de lastimarse [2], los resultados de estas pruebas se observan en la tabla 1.

Nombre	Estatura (m)	Alt. asient o (cm)	Alt. Max peda l (cm)	Alt. peda l (cm)	Dist. Vert. peda l (cm)	Dist. Hor. Pedal . (cm)	Agarr e (cm)
Hombre 1	1.79	50	38	26	24	84	42
Mujer 1	1.73	50	38	26	24	81	47
Mujer 2	1.56	50	38	26	24	63	47
Hombre 2	1.81	50	38	26	24	86	40
Hombre 3	1.70	50	38	26	24	86	42
Hombre 4	1.66	50	38	26	24	80	42
Mujer 3	1.68	50	38	26	24	70	42
PROM.	1.70	50	38	26	24	78.57	43.14

Tomando en cuenta los promedios obtenidos se pudo proponer un diseño funcional y ergonómico del aparato, el cual queda ejemplificado en la figura 2. Este diseño fue elaborado tomando en cuenta parámetros de importancia como lo es: las catarinas, los ejes primarios, bujes, cadenas y chumaceras [3]; también se tomó en cuenta la ubicación donde se montará el sistema eléctrico y el sistema de control para evitar vibraciones mecánicas y ruido magnético así como la ubicación de los elementos de aplicación como el conector USB y los estímulos visuales.

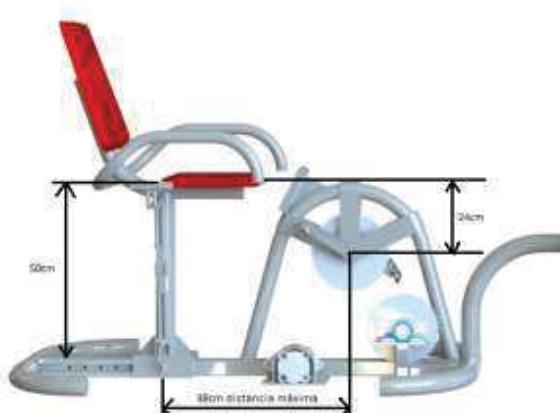


Fig. 2 Diseño funcional de la bicicleta horizontal

2.2 Diseño del sistema eléctrico

El diseño del sistema eléctrico se realizó considerando los tipos de generadores eléctricos que fueron principalmente dos: los alternadores y los dinamos, para definir cuál era el más adecuado, nos basamos en sus características, destacando las del alternador (ver figura 3): mayor gama de velocidad de giro, el sistema de rotor compacto, menor peso y espacio, trabaja en ambos sentidos de giro y tiene una vida útil superior a la del dínamo, eligiendo este para generar la energía eléctrica a partir del movimiento mecánico, el siguiente paso en la fase del diseño fue parametrizarlo y de esta manera diseñar el circuito de acondicionamiento para poder comunicarlo con el sistema de control.



Fig. 3 Alternador

El aspecto más importante de la fase del diseño del sistema de acondicionamiento fue la determinación del punto de donde se iba a tomar la señal de salida del alternador, identificando este punto en el puente de diodos que tiene de manera interna el alternador. Una vez hecho se caracterizó la señal de manera que se identificó haciendo girar el eje del alternador con un motor de velocidad variable y observar la señal en el osciloscopio con magnitud de 0.7V a 5.6V de C.D., señal que será la entrada del sistema de control.

El diagrama de conexión de la señal del alternador se observa en la figura 4, mostrando los elementos principales para su funcionamiento, la batería como fuente de alimentación del alternador y una lámpara la cual indica cuando está funcionando o no el alternador. La conexión entre la batería y el alternador permite que una vez que el usuario comience a pedalear y el alternador genere voltaje, este se retroalimente a la batería de manera que emule el proceso de carga de la batería lo que beneficia al sistema, ya que al contar con esta retroalimentación la batería permanecerá con carga todo el tiempo que esté en funcionamiento el sistema.

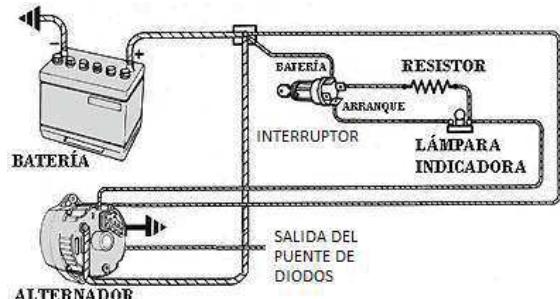


Fig. 4 Diagrama de Conexión

Es conveniente resaltar que se decidió colocar un interruptor como señal de inicio general del sistema ya que si se omitiera se corre el riesgo de que la batería se descargue sin uso del sistema, ya que debido al funcionamiento natural del alternador al no estar trabajando este consume un mínimo de corriente que provoca que un aparato sin funcionar genera que la batería se descargue.

2.3 Diseño del Sistema de Control

La etapa de control se inició con el diseño de una tarjeta, para que una vez acondicionada la señal de salida del alternador, se pudiera utilizar un programa de procesamiento que tomara la señal y la convirtiera en una palabra binaria equivalente al voltaje de entrada, posteriormente el valor binario se comparaba con parámetros definidos inicialmente, para generar una salida que accionaba los elementos de la etapa de potencia y pudiera alimentar tanto la salida de puerto USB, como un sistema de iluminación con efectos visuales.

Para simular el funcionamiento correcto del sistema de control, se empleó Proteus y se comprobó que el circuito de acoplamiento era adecuado así como la etapa de potencia, evitando el daño de elementos digitales a través de las señales analógicas.



Fig. 5 Funcionamiento de los sistemas de iluminación y recarga de dispositivos móviles

El diseño e integración de la placa del circuito de control se realizó con el software Eagle 6.1.0, ya que tiene librerías de componentes electrónicos

amplias y se puede personalizar el diseño de las pistas, posteriormente se exporto el diseño a una impresora especializada en la construcción de PCB (considerando para su elaboración las dimensiones de la bicicleta horizontal), en este diseño se tomó en consideración el ancho de las pistas ya que como la tarjeta tendría los diferentes subsistemas contenidos en ella se tuvo que calcular las corrientes máximas para así evitar que las pistas se botaran por las magnitudes con las que se trabajaran en la tarjeta.

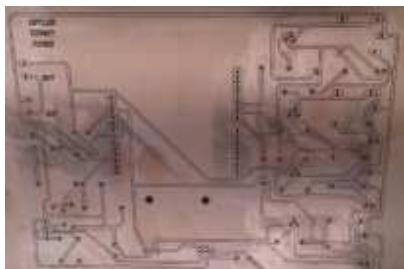


Fig. 6 PCB de la Tarjeta de Control

Para la conexión física de la señal de salida del alternador hacia el microcontrolador se acondiciona la señal, permitiendo que este ejecutara el programa diseñado para encender de forma secuencial tiras de led ultrabrillantes (estímulos visuales), generando motivación al momento de pedalear la bicicleta [4].

El sistema de control también integra una etapa del circuito que permite la carga de dispositivos portátiles a través de conexión USB, este diseño permite que la carga de los dispositivos se realice sin importar la marca, el esquema general de la etapa de control se muestra en la figura 7.

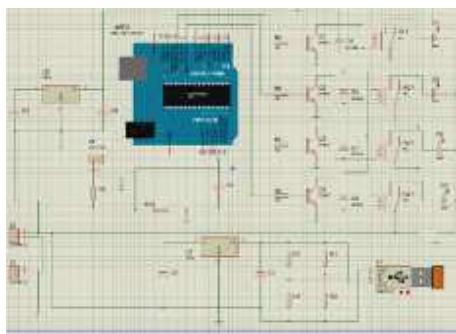


Fig. 7 Circuito de Control

Referente al programa de control se diseñó de manera estuviera monitoreando de manera constante la señal proveniente del alternador que es la que dictamina la señal que estará activando el sistema

para la parte de estímulos visuales. El diagrama de flujo del programa se aprecia en la figura 8.



Fig. 8 Diagrama de Flujo del Programa

2.4 Integración de los Sistemas

El ensamble final de la tarjeta se observa en la figura 9, durante las pruebas se pudo observar la forma en cómo se encienden los leds, siendo esto una manera de ajustar o calibrar los estímulos de manera que se enciendan con un mayor o menor esfuerzo, estos ajustes son de dos maneras, una modificando los parámetros de referencia del programa en el Arduino y otra modificando los elementos activos y pasivos de la etapa de acondicionamiento de la señal proveniente del alternador, esto permite que el sistema sea multifuncional siempre y cuando se ajusten de manera correcta los parámetros.



Fig. 9 Integración de Sistema Mecánico, Eléctrico y Electrónico.

Durante la integración se tuvieron que colocar de manera precisa los subsistemas ya que como las señales que se utilizaron en las simulaciones se ajustaron de manera independiente, al momento de integrar había que tener cuidado con las señales de tierra analógica y tierra digital ya que como todas llegan a la tarjeta de control había que acoplárlas de manera correcta para evitar mal funcionamientos o daños a los elementos electrónicos.

Otro factor que se evitó con un diseño mecánico correcto fue la parte del ruido mecánico y magnético en la etapa de control, ya que al momento de hacer la integración y poner en ejecución el sistema la parte del sistema mecánico genera una vibración natural sobre el sistema y el alternador un campo magnético que ambos repercuten a los sistemas electrónicos pero como el diseño esta eléctricamente aislado y ópticamente acoplado se tiene un funcionamiento adecuado y riesgo de un daño al sistema integral, esto se tiene que tomar en cuenta ya que la idea es que este sistema en el prototipo pueda ser implementado en algún parque y pueda ser operado por cualquier persona sin importar su edad.

3. Análisis de resultados

Debido al funcionamiento natural del alternador, el cual es el de suministrar un voltaje proporcional a la intensidad con la que el rotor gira y así como la de alimentar a la batería para que se mantenga en carga, el interruptor, mencionado en la etapa del sistema eléctrico, fue el elemento que se incorporó para poder mantener este principio en operación ya que si se deja energizado el sistema existe la posibilidad de que se descargue la pila, mismo fenómeno que ocurre en los automóviles, y con este elemento suprimimos este posible escenario; quedando este punto como una mejora en el futuro ya que el interruptor mecánico se puede sustituir por algún otro electrónico lo que brindaría al sistema un punto de accionamiento inteligente.

Con esto se tiene otro punto importante en el prototipo ya que es un sistema que aunque cuenta con batería y parece ser un elemento que implicaría un gasto, el alternador hace que la pila sea recargable y con esto se genera energía propia que alimenta los elementos de la tarjeta, el sistema de carga de celulares y tablets, así como el alternador.

Ya con la colocación de los estímulos visuales habría que verificar que el consumo de corriente de la batería era suficiente para todos los sistemas que integran al mecanismo. Al cuantificar

este dato se revisó que el consumo estuviera por debajo de sus datos técnicos específicos, y los resultados de las pruebas se aprecian en la figura 10.



Fig. 10 Pruebas de los estímulos visuales

En esta etapa de desarrollo se logró que la carga de los dispositivos electrónicos se hiciera sin problemas ya sea con los estímulos visuales funcionando o no, lo que brinda un versatilidad bastante buena al prototipo desarrollado además de que no existe riesgo de dañar al celular o tablet por alguna sobretensión que pudiera existir. Cabe mencionar que debido al estudio que se hizo del protocolo de comunicación USB se puede cargar cualquier dispositivo electrónico Tablet o Smartphone ya que se el sistema de carga cuenta con los niveles de voltaje necesarios para que los dispositivos detecten el sistema de carga.

De acuerdo a las pruebas realizadas, los componentes que se incluyen en la tarjeta electrónica para controlar el tiempo de encendido y apagado de los leds, son dos resistencias de $50\text{ k}\Omega$ de watt junto con un capacitor de $100\text{ }\mu\text{F}$, además de cambiar el rango en la programación, para que el encendido de estos ocurra de manera espaciada y el apagado ocurra de manera rápida. Es decir, el tiempo o velocidad de encendido y apagado en los estímulos visuales se puede ajustar cumpliendo la regla de la constante de tiempo de un circuito RC, la cual se rige por la siguiente expresión matemática:

$$\tau = RC \quad (1)$$

Con esta expresión se pueden tener diferentes escenarios para configurar el encendido de los estímulos visuales lo que permite que el sistema se pueda ajustar a que se enciendan sin mucho esfuerzo

mecánico o bien que se requiera pedalear a la máxima velocidad para poder activar los estímulos en su totalidad.

4. Conclusiones

Una vez concluido el desarrollo del prototipo y en función de los resultados presentados se observa que el diseño fue factible para su fabricación en función de las variables antropométricas consideradas.

De igual forma, el diseño y fabricación del sistema mecánico permitió obtener las revoluciones esperadas para producir cantidades de energía eléctrica necesaria para alimentar los estímulos visuales y realizar la carga de dispositivos portátiles a través de puerto USB de forma correcta.

El sistema de control fue un factor fundamental para determinar el tipo de estímulos visuales que permitieran motivar a los usuarios en el uso del aparato, también es importante mencionar que el prototipo es funcional para producirse a gran escala, ya que el costo de fabricación es considerablemente menor al beneficio que se obtendría por su uso en lugares públicos, donde la población tendría acceso para generar la energía eléctrica necesaria para satisfacer demandas energéticas mayores.

El diseño del sistema puede mejorarse en aspectos de estética y también sobre materiales de menor costo que permitan la penetración del prototipo al mayor número de regiones posibles, incluidas las que tienen poco o nulo acceso de la red eléctrica comercial.

El concepto de este proyecto se aplica actualmente en gimnasios cerrados bajo el nombre de Gimnasios Ecológicos, aquí se almacena la energía en un banco de baterías como centro de energía para su alumbrado interno y/o externo, este proyecto innova bajo un escenario nuevo ya que su impacto inmediato será en parques al aire libre y otra propuesta del mercado donde se puede implementar es como aparato doméstico para que cualquier usuario lo pueda adquirir y lo aplique desde su hogar, esto beneficia varias áreas de interés y que son problemática actual entre nuestra sociedad.

Referencias

- [1] Sánchez A. “Salud y bicicleta”, Centro de salud de la Universidad Alemana del deporte, recuperado de <http://www.granabike.com/consejos/lista/338-montar-en-bici.html>
- [2] Wolfgang L y Joachim V. “Enciclopedia de salud y seguridad en el trabajo. Ergonomía”
- [3] Mott R. “Diseño de elementos de máquinas”. Prentice Hall Pearson. 2006.
- [4] Lajara Vizcaíno J. R. and Pelegrí Sebastiá J., “Sistemas Integrados con Arduino”, 1ra ed., Ed. Alfaomega Marcombo 2013.

Diseño y Análisis Estructural de un Soporte Parcial de Peso para Fisioterapia

Arriaga-Arriaga, Cesar Augusto, Gómez-Pavón Luz del Carmen,
Muñoz-Pacheco Jesús Manuel, Félix-Beltrán Olga Guadalupe y Luis-Ramos Arnulfo

Posgrado en Ingeniería Electrónica, Facultad de Ciencias de la Electrónica,
Benemérita Universidad Autónoma de Puebla
San Claudio y 18 Sur, Edif. 109A-CU, Col. San Manuel, Puebla, Pue.

Resumen

Se presenta el diseño y el estudio estructural de un soporte parcial de peso para aplicaciones en fisioterapia. Por medio del software de diseño SolidWorks se realiza el análisis de materiales así como el diseño de la estructura mecánica del soporte parcial de peso. Debido a que el soporte parcial de peso se utilizará en pacientes con discapacidad motriz, mediante la herramienta computacional SolidWorks Simulation aplicando el método del elemento finito, se lleva a cabo el estudio y análisis de esfuerzos, deformaciones así como desplazamientos de la estructura mecánica diseñada, con base al peso de los pacientes que debe de soportar. Se asegura de esta manera que el soporte cumple con las condiciones de seguridad y confort tanto para el paciente como el fisioterapeuta.

Palabras clave: Soporte mecánico, SolidWorks, fisioterapia, método del elemento finito.

1. Introducción

Con base en la información reportada por el INEGI respecto al Censo 2010 [1], en los últimos años el tema de la discapacidad ha tenido un significativo incremento de atención por parte de los profesionales, y la sociedad en su conjunto. En México, al año 2010 las personas con algún tipo de discapacidad eran 5 millones 739 mil 270, lo que representa el 5.1% de la población total. De las personas con alguna discapacidad, 49% son hombres y 51% mujeres. La limitación de la movilidad es la de mayor frecuencia entre la población del país; alrededor de la mitad de las limitaciones declaradas se refieren a caminar o moverse.

El impacto de la automatización en la medicina ha dado lugar a nuevas tecnologías que han permitido mejorar las técnicas de diagnóstico, tratamiento y prevención de diversas enfermedades. Particularmente, en aplicaciones fisioterapéuticas se ha buscado el desarrollo de sistemas que coadyuven en la mejora del paciente en un plazo de tiempo menor, mediante el uso de sistemas mecánicos automatizados confortables, que sean capaces de llevar a cabo procesos fisioterapéuticos y de seguimiento de evolución del paciente proporcionando además mayor seguridad, confiabilidad y comodidad en el uso de los mismos [2]. Específicamente, respecto a la discapacidad motriz, existen tres componentes esenciales de la locomoción, estas son: la postura, el equilibrio y la coordinación. Diversos trastornos neurológicos o músculo-esqueléticos pueden afectar a uno o más de los componentes antes mencionados, limitando la capacidad de la persona a caminar normalmente. Con base en diversas investigaciones básicas y aplicadas se han desarrollado diversas técnicas para ayudar en el reentrenamiento de las funciones del andar para una amplia gama de alteraciones. A estas técnicas se les llama terapia de marcha por soporte parcial de peso.

En este trabajo se presenta el diseño de un sistema parcial de peso para coadyuvar en la terapia para discapacidad motriz. Se presenta la propuesta de diseño de la estructura mecánica así como el análisis de la misma por medio del método del elemento finito, por medio del software CAD/CAE SolidWorks 2014. Se determina, por medio del estudio y análisis, las dimensiones de la estructura mecánica, los materiales que se deben utilizar para su posterior construcción y el peso a soportar por la estructura.

2. Diseño y análisis estructural utilizando SolidWorks

A continuación se describe la aplicación del método del elemento finito para determinar el material a utilizar en el diseño del soporte parcial de peso, así como el diseño y análisis de esfuerzos, deformación y desplazamientos del mismo.

2.1 Método del elemento finito

El software de diseño SolidWorks utiliza el *método del elemento finito* (FEM, por sus siglas en inglés) para el análisis de estructuras mecánicas. El FEM es una técnica numérica para analizar diseños de ingeniería y está aceptado como el método de análisis estándar, debido a su generalidad y compatibilidad para ser implementado en computadoras. El FEM divide el modelo en numerosas piezas pequeñas de formas simples llamadas elementos, que reemplazan eficazmente un problema complejo por muchos problemas simples que deben ser resueltos de manera simultánea. Particularmente es una de las principales tareas que deben ser realizadas por el diseñador o analista.

Los elementos comparten puntos comunes denominados nodos. El proceso de división del modelo en pequeñas piezas se denomina mallado. El comportamiento de cada elemento es bien conocido bajo todas las situaciones de soporte y carga posibles. El FEM utiliza elementos con formas diferentes y la respuesta en un elemento, en cualquier momento, se interpola desde la respuesta en los nodos del elemento. Cada nodo está descrito en detalle por un cierto número de parámetros, según el tipo de análisis o del elemento utilizado. Para el análisis estructural, la respuesta de un nodo está descrita, por lo general, por tres translaciones y tres rotaciones, denominados grados de libertad (GDL). El análisis que utiliza FEM se denomina *análisis de elementos finitos* (FEA).

El software formula las ecuaciones que rigen el comportamiento de cada elemento teniendo en cuenta su conectividad con los demás elementos. Estas ecuaciones hacen referencia a la respuesta de cargas, restricciones y propiedades del material conocidas. A continuación, el programa organiza las ecuaciones en un conjunto mayor de ecuaciones algebraicas simultáneas y resuelve las desconocidas. En el análisis de esfuerzos, por ejemplo, el solver encuentra los desplazamientos en cada nodo y, posteriormente, el programa calcula las deformaciones unitarias y finalmente los esfuerzos.

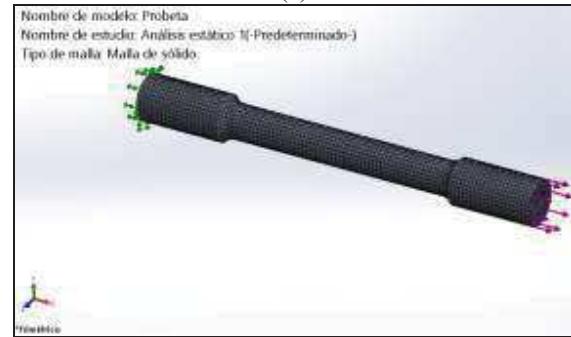
2.2 Análisis de materiales

En esta sección se muestran los resultados de los ensayos de tracción o también conocidos como ASTM E8M (métodos de prueba estándar para tensión en materiales metálicos), denominado así por la American Society for Testing and Materials (ASTM, por sus siglas en inglés) [39], siendo el objetivo del ensayo de tracción determinar aspectos importantes de la resistencia y alargamiento de materiales, que pueden servir para el control de calidad, las especificaciones de los materiales y el cálculo de piezas sometidas a esfuerzos.

Normalmente se deforma una probeta hasta su rotura, con una carga de tracción que aumenta gradualmente y que es aplicada uniaxialmente a lo largo del eje de la probeta. Los ensayos de tracción se realizan en materiales metálicos (probetas de aluminio y de acero); dichas pruebas se realizaron a probetas en forma de halterio como la presentada en la figura 1a); empleando el software SolidWorks mediante el cual es posible llevar a cabo un análisis por elementos finitos. En la figura 1b) se presenta el mallado fino, mediante el cual se tiene un número de total de 73567 Nodos y 48993 Elementos.



(a)



(b)

Figura 1. a) Diseño de probeta en forma de halterio y b) mallado fino aplicado para el análisis de elemento finito.

Para llevar a cabo este ensayo se simula la experimentación, pues se dibuja la probeta en SolidWorks. Posteriormente, mediante un estudio realizado por medio de la herramienta SolidWorks Simulation incluida en el software, se simula la carga de tensión aplicada sobre la probeta y se obtienen como resultado las deformaciones unitarias.

Los ensayos de tracción se realizaron a probetas de los siguientes materiales aplicando una fuerza de 1800N: a) Acero 304; b) Acero 316; c) Aluminio 1060; d) Aluminio 3003. Dichos materiales son ampliamente utilizados en la industria para la construcción de estructuras y soportes mecánicos.

En la figura 2a) se muestran los resultados del análisis de esfuerzo nodal realizado al material, además del punto mínimo y máximo de esfuerzo. Como se puede ver las zonas en color azul y verde son las zonas que están más alejadas del límite elástico y las zonas naranjas y rojas están más cerca de este límite.

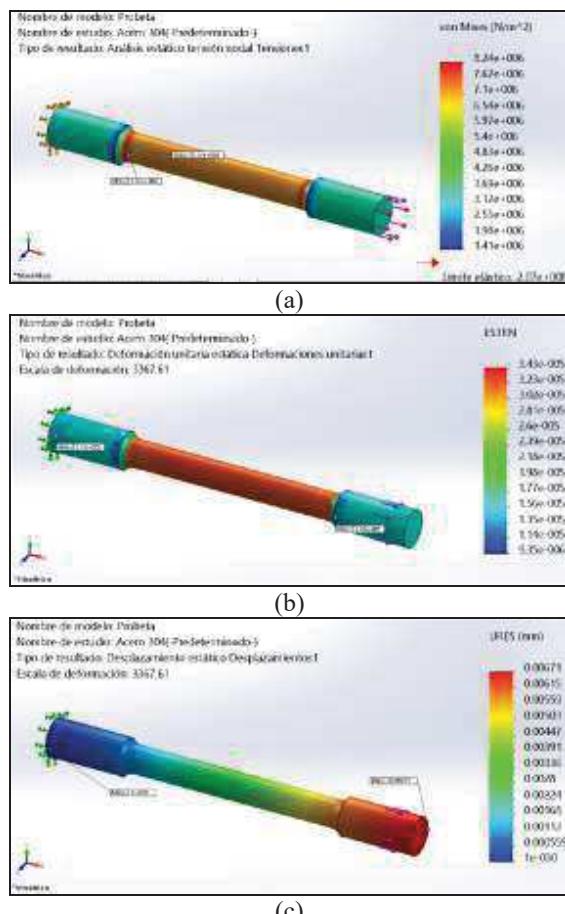


Figura 2. Resultados del análisis de a) esfuerzos, b) deformación y c) desplazamiento, realizado a probetas de acero.

En la fig. 2b) se presentan los resultados del análisis de deformación unitaria y de igual manera se presentan los índices de menor y mayor deformación. Como se puede ver la zonas en color verde son las regiones que presentan menor deformación unitaria, en cambio las zonas en color rojo son las que presentan un mayor índice de deformación unitaria. Finalmente, en la fig. 2c) se muestran el desplazamiento que presentan los materiales al someterlos a una fuerza de 1800N.

En resumen, en la tabla 3.1 se presentan los resultados obtenidos, del reporte que SolidWorks genera mediante la herramienta de SolidWorks Simulation.

Tabla 1: Resultados obtenidos del ensayo de tracción mediante SolidWorks aplicando el método del elemento finito

MATERIAL	Límite elástico (Nm)	Tensión Mínima (Nm)	Tensión Máxima (Nm)
AISI 304	2.06807E+08	1.4148E+06	8.24248E+08
AISI 316	1.37895E+08	1.44454E+06	8.23435E+08
Aleación 1060	2.76742E+07	1.53312E+06	8.20994E+08
Aleación 3003	4.13613E+07	1.53312E+06	8.20994E+08

MATERIAL	Deformación Mínima	Deformación Máxima	Desplazamiento Min. (mm)	Desplazamiento Máx. (mm)
AISI 304	-0.30414E-06	0.40418E-05	0	0.00071137
AISI 316	0.025E-06	0.40410E-05	0	0.00065048
Aleación 1060	0.35621E-05	0.971768E-05	0	0.0184518
Aleación 3003	0.35621E-05	0.971768E-05	0	0.0184818

Como se puede observar en la Tabla 1, el límite elástico de los aceros elegidos para hacer estas pruebas, se encuentra por encima del límite elástico de las aleaciones de aluminio, por un promedio de $137.88 \times 10^6 N/m^2$. Lo anterior, indica que tienen mayor resistencia a la deformación los aceros, lo cual se comprueba en la tabla 1, ya que el aluminio tiene mayor deformación unitaria y presenta mayor desplazamiento, además el acero soporta una mayor tensión, con una diferencia entre ambos materiales de $32490N/m^2$.

Tomando en cuenta estos resultados se propone utilizar un Acero 316 para la construcción del soporte parcial de peso.

2.3 Diseño de la estructura mecánica

El Acero 316 es altamente recomendable para la utilización en procesos alimenticios, industriales o de producción, además contiene altos niveles de Níquel y Molibdeno, para aumentar la durabilidad y la resistencia a la corrosión.

Con base en lo anterior, se propone el diseño de un soporte mecánico parcial de peso como se muestra en la estructura mecánica en la figura 3, el

cual será analizado para garantizar la seguridad del paciente.



Figura 3: Diseño de la estructura mecánica del soporte parcial de peso.

En este caso, las medidas que tiene el soporte son: largo de 1.20m, ancho de 1.50 y de alto 1.90m, es decir, este sería el espacio que ocuparía el soporte.

Además se incluye en el diseño ruedas con freno de seguridad que soporta 75kg. cada una, con lo cual al tener 4 se tiene un soporte para 200kg. sobre las ruedas.

En la figura 4 se presenta el despiece del soporte, el cual está conformado por las siguientes piezas mecánicas: base, acoplador, poste, yugo y agarradera.

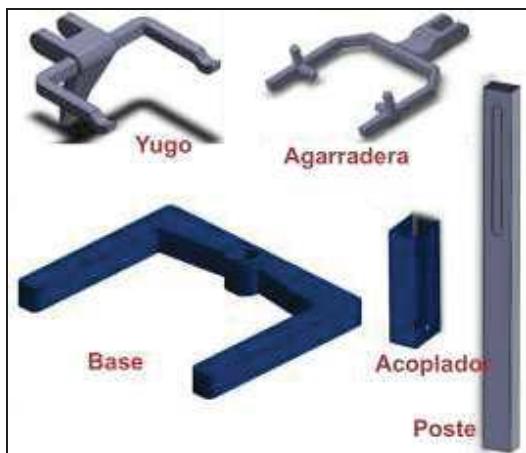


Figura 4: Despiece del soporte parcial de peso diseñado.

En la figura 5, se presenta el análisis de deformación de cada una de las piezas mecánicas del soporte parcial de peso.

Se aplicó la fuerza correspondiente a un peso de hasta 150 kg, esto es, el soporte parcial de peso podrá cargar pacientes de hasta 150 kg, con una estatura de hasta 1.80 m.

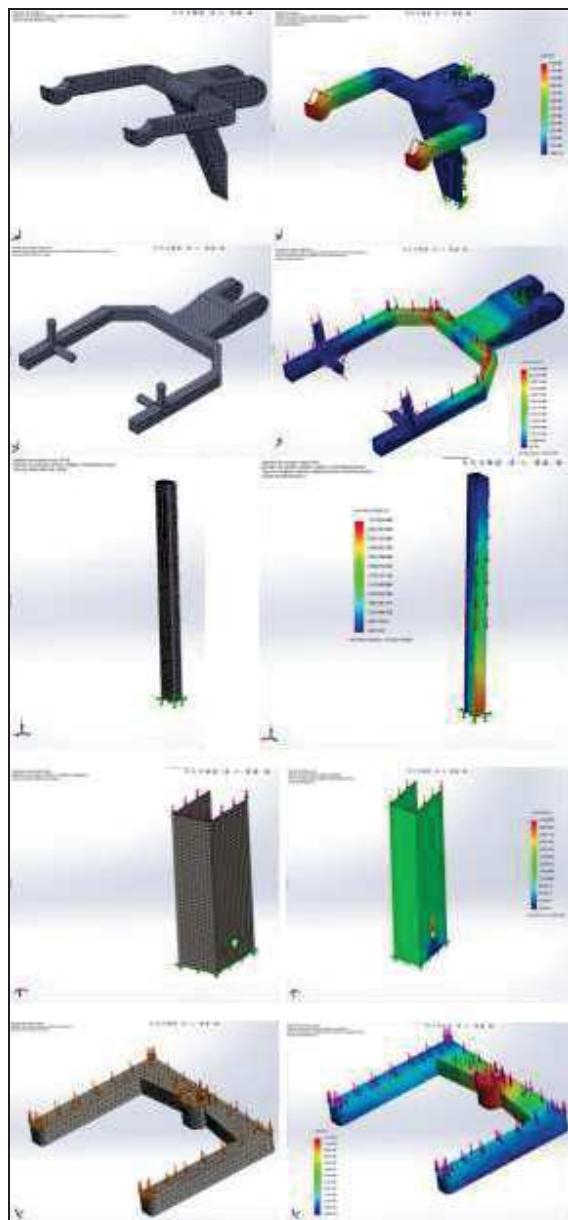


Figura 5: Mallado y análisis de deformación aplicado a cada una de las piezas que conforman el soporte parcial de peso.

En este caso se observa que al ser aplicada la fuerza en cada una de las piezas sufren cierta deformación, siendo las zonas de color rojo y naranja las que tienen mayor deformación en comparación de las zonas de color verde y azul que tienen la menor deformación de toda la pieza.

Sin embargo cabe mencionar que los resultados que se obtuvieron de la simulación presentan que aunque existan zonas de color rojo la tensión máxima que presentan está por debajo del límite elástico del acero a emplear en la construcción, lo que implica un alto porcentaje de confiabilidad en el uso del mismo.

Finalmente, en la figura 6 se presenta el mallado aplicado al soporte parcial de peso completo, en donde se han generado 20130 Elementos y 34598 Nodos para el análisis de la estructura.



Figura 6: Mallado aplicado al soporte parcial de peso.

Una vez aplicado el mallado, mediante el mismo software se obtiene el análisis de esfuerzos a la estructura completa, representado en la figura 7.

A partir del análisis de deformación se observa que el mayor esfuerzo que presenta la estructura compuesta está en el yugo y agarradera ya que son las partes donde se sujetará y se sostendrá el paciente durante el proceso de la terapia, pero de igual manera el máximo esfuerzo presente en las zonas de color rojo y naranja están por debajo del límite elástico del acero 316.

En la figura 8 se muestra la vista frontal, izquierda, superior y trimétrico del diseño del

soporte.

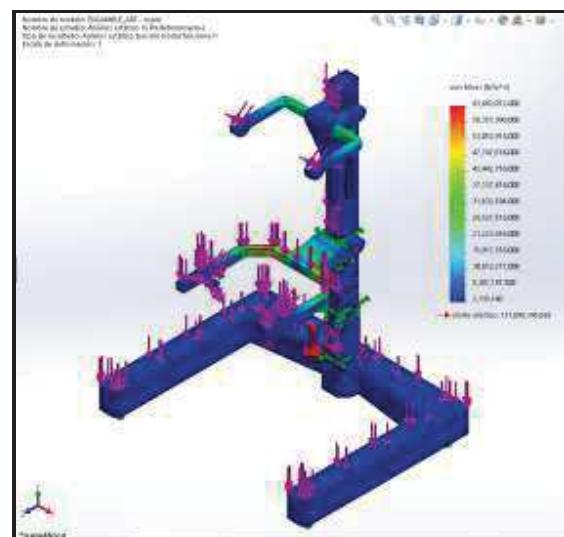


Figura 7: Análisis de deformación obtenido del soporte parcial de peso.

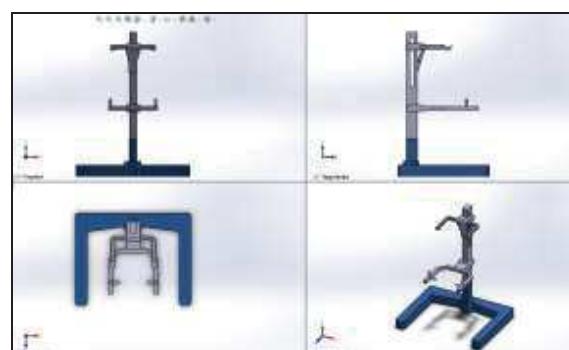


Figura 8: Diferentes vistas del Soporte Mecánico Parcial para Fisioterapia.

3. Conclusiones

Se ha realizado el diseño de la estructura mecánica de un soporte parcial de peso para fisioterapia. A partir del estudio y análisis de esfuerzos, deformaciones y desplazamientos de la estructura mecánica, realizada por medio del software de diseño SolidWorks, aplicando el método del elemento finito, se puede concluir que el soporte diseñado es capaz de soportar un peso de hasta 150kg. Por lo tanto, el soporte parcial de peso puede cumplir con los requerimientos de seguridad, para utilizarse en el área de fisioterapia coadyuvando en los tratamientos de discapacidad motriz con pacientes de hasta 150kg de peso y una estatura de 1.80m.

Referencias

- [1] INEGI.
<http://cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P>, Última Consulta: 04-06-2014.
- [2] Asociación Mexicana de Fisioterapia.
<http://www.amefi.org.mx/>, Última Consulta: 04-06-2014.
- [3] Jacob Fish and Ted Belytschko. A first course in finite elements. John Wiley and Sons, 2007.
- [4] Yu-Qiu Long, Song Cen, and Zhi-Fei Long. Advanced finite element method in structural engineering. Springer, 2009.
- [5] John Edward Akin. Finite Element Analysis Concepts: Via SolidWorks. World Scientific, 2010.
- [6] SolidWorks Simulation y análisis de elemento finito.
<http://www.solidworks.es/sw/products/simulation/finite-element-analysis.htm>, Última Consulta: 04-06-2014.
- [7] Stephen Timoshenko. History of strength of materials: with a brief account of the history of theory of elasticity and theory of structures. Courier Dover Publications, 1983.

Control Háplico con Planificación de Trayectorias Aplicado a Novint Falcon

E. Zepeda-Ruelas, J. Gudiño-Lau, M. Durán-Fonseca, S. Charre-Ibarra y J. Alcalá-Rodríguez

Facultad de Ingeniería Electromecánica.

Universidad de Colima.

Manzanillo, Colima, C.P.28860

E-mail: ezruelas@ucol.mx, jglau@ucol.mx, mduran@ucol.mx, scharre@ucol.mx, janethalcalal@ucol.mx

Resumen

Actualmente existen dispositivos hapticos desde los simples hasta los complejos que brindan retroalimentación de fuerza y gracias a la interacción con ambientes virtuales es posible desarrollar sistemas para la mecanoterapia, rehabilitación de extremidades superiores y aplicaciones médicas.

Este trabajo presenta el seguimiento y planificación de trayectorias en un dispositivo haptico, controlado por un sistema lineal, con la finalidad de apoyar en un futuro en rehabilitación del ser humano.

Palabras clave: Control lineal, planificación de trayectorias, interfaz haptica.

1. Introducción

Recientemente la investigación de la realidad virtual ha sido aplicada en las diferentes áreas médicas. La rehabilitación con nuevas tecnologías, es considerada una de las áreas de investigación más presentes en la actualidad para la rehabilitación del ser humano, una de estas investigaciones, que se ha realizado en Estados Unidos de América en el Departamento de Ingeniería Eléctrica y Computación de la Universidad de Purdue, donde se realizaron prácticas de rehabilitación, en este caso se utilizó el dispositivo PHANTOM 1.0, para escribir caracteres en inglés con un lápiz virtual. [1,2,3,4].

La problemática que se presenta al diseñar estrategias de control no lineal con fines de guiado haptico para

diagnóstico y rehabilitación médica, es dar las soluciones correctas haciendo uso de un método de control que cumpla con un sistema electromecánico, este artículo se basa en teoría de control convencional, además se presenta un control lineal aplicado a un dispositivo haptico, el cual consiste en el seguimiento de trayectorias en coordenadas cartesianas del dispositivo haptico. Este trabajo está organizado de la siguiente forma; la introducción en la sección 1, la sección 2 se hace una descripción de las interfaces hapticas, la sección 3 se describe al guiado haptico, la sección 4 se describe la trayectoria deseada y el controlador empleado, en la sección 5 se muestra la plataforma experimental, la sección 6 se presentan los resultados experimentales y finalmente la sección 7 muestran las conclusiones.

2. Interfaces hapticas

La Fig. 1 muestra el dispositivo haptico Novint Falcon, éste permite la interacción con un ambiente virtual generado por un WIN API. Tiene una velocidad de muestreo de 0.5Khz, esto permite al software tener una buena resolución de donde se encuentra y así brindar al usuario un preciso estímulo kinestésico al momento de interactuar con el mundo virtual, Novint Falcon es un robot tipo delta que consta de 3 actuadores, cuyo sistema de coordenadas está dado por la Fig. 1 y posee 3 grados de libertad. El efecto final intercambiable del dispositivo corresponde a un mango que el usuario puede sujetar con su mano simulando un dedo (un punto) en el entorno virtual, consta de 4 botones para una mayor interacción en el ambiente virtual, en la Fig. 2 muestra el robot sin su cubierta exponiendo el mecanismo y los actuadores del dispositivo haptico[5,6,7].

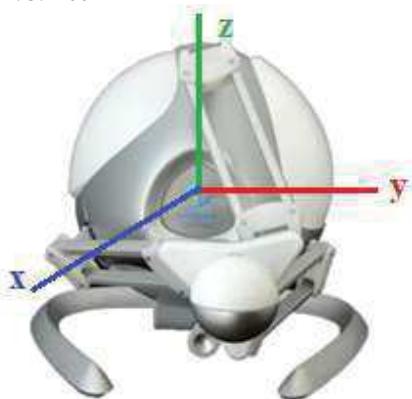


Fig. 1. Dispositivo háptico Novint Falcon con sistema de coordenadas.

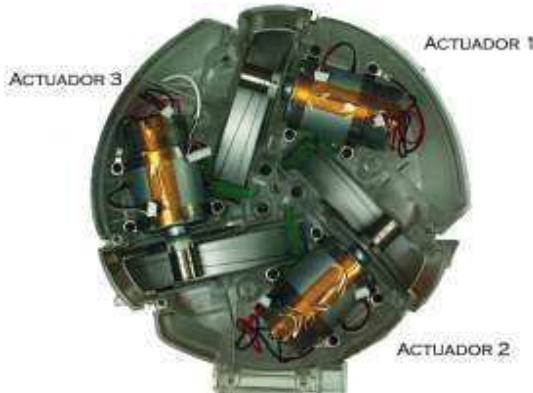


Fig. 2. Actuadores del Novint Falcon

Este dispositivo háptico es comercial, fue desarrollado para realidad virtual en videojuegos por la empresa Novint Technologies Inc., por lo que la cinemática directa, inversa y el modelo dinámico no se encuentran en la literatura[8].

3. Guiado háptico

Los seres humanos ofrecen dos conceptos diferentes de formas de exploración háptica, la activa y la pasiva. La exploración háptica activa es cuando el usuario controla sus propios movimientos. La exploración háptica pasiva es cuando la mano del operador humano es guiada por otra persona. Una de las grandes aplicaciones que existen hoy en día es el guiado háptico para rehabilitación de pacientes con discapacidades motrices o implícitamente para poder diagnosticar el inicio de un padecimiento como dislexia o inmovilidad de un músculo a temprana edad.

4. Planificación de las trayectorias

En la planificación de la trayectoria se emplea para asegurar que la trayectoria deseada sea suave, para ello se consideró una trayectoria con un polinomio de 3 orden es una

alternativa viable debido al peso computacional y la practicidad del mismo como se muestra en la ecuación (1):

$$q_t = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \quad (1)$$

$$t_i < t \leq t_f$$

donde:

q_t = Posición de la trayectoria.

a_0 = posición inicial.

a_1 = aceleración inicial.

a_2 = posición final (variante)

a_3 = velocidad de la posición final (a_2). .

La trayectoria con interpolación cúbica empleada es mostrada en la Fig. 3:

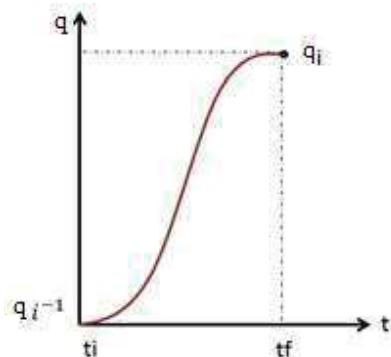


Fig. 3. Trayectoria con interpolación cúbica

Los datos de la figura son:

i^{-1} = posición inicial

q^i = posición deseada.

t_i = tiempo inicial.

t_f = tiempo final de la trayectoria.

Para el diseño de la trayectoria se consideró las siguientes condiciones iniciales para la posición y velocidad:

$$q_0 = a_0; \quad (2)$$

$$q_i = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3; \quad (3)$$

$$0 = a_1; \quad (4)$$

$$0 = a_1 t_f + 2a_2 t_f^2 + 3a_3 t_f^3; \quad (5)$$

Realizando manipulaciones matemáticas se tiene:

$$a_2 = \frac{3a_3 t_f^2}{2t_f}; \quad (6)$$

$$a_3 = -\frac{2q_i - q_0}{t_f^3}; \quad (7)$$

$$a_2 = \frac{3(q_i - q_0)}{t_f^2}; \quad (8)$$

4.1 Diseño de trayectoria propuesta

Se diseñan dos trayectorias una línea recta y una circunferencia en dos periodos de tiempo:

Caso 1 para $t_1 < t < t_2$:

$$q_{t1} = a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3; \quad (9)$$

$$y_{d1}^{i-1} = 0; y_{d1}^i = q_{t1}; \quad (10)$$

$$z_{d1} = 0; \quad (11)$$

Caso 2 para $t_1 < t < t_2$:

$$f_2 = t_2 - t_1; \quad (12)$$

$$q_{t2} = a_0 + a_1 t_{f2} + a_2 t_{f2}^2 + a_3 t_{f2}^3; \quad (13)$$

$$y_{d2}^{i-1} = 0; y_{d2}^i = r \sin(q_{t2}); \quad (14)$$

$$z_{d2}^{i-1} = 0; z_{d2}^i = r \cos(q_{t2}); \quad (15)$$

El robot solo tendrá movimiento en y y z, r es el radio de la circunferencia, para la trayectoria deseada se tiene:

$$r = 1.18 \text{ in (3cm).}$$

Las derivadas de las trayectorias deseadas para el caso 1 son:

$$\dot{x}_{d1} = 0; \quad (16)$$

$$\dot{y}_{d1} = a_1 + 2a_2 t_1 + a_3 t_1^2; \quad (17)$$

$$\dot{z}_{d1} = 0; \quad (18)$$

Para el caso 2:

$$\dot{q}_{t2} = a_1 + 2a_2(t_2 - t_1) + a_3(t_2 - t_1)^2; \quad (19)$$

$$\dot{x}_{d2} = 0; \quad (20)$$

$$\dot{y}_{d2} = \dot{q}_{t2} \frac{\pi i}{180} r \cos(q_{t2}); \quad (21)$$

$$\dot{z}_{d2} = \dot{q}_{t2} \frac{\pi i}{180} r - \sin(q_{t2}); \quad (22)$$

El error de posición cartesiano se define como:

$$\tilde{x} = x_d - x, \quad (23)$$

el error de velocidad cartesiana es definida como:

$$\dot{\tilde{x}} = \dot{x}_d - \dot{x}, \quad (24)$$

y la integral es definida como

$$\int \tilde{x}(dt) = \sum_{i=0}^n \frac{(\tilde{x} + \tilde{x}_{i-1})}{2} T \quad (25)$$

El diagrama de bloque del controlador empleado para el seguimiento de trayectorias cartesianas es mostrado en la Fig. (3) y la ley de control es:

$$f_x = k_p \tilde{x} + k_d \dot{\tilde{x}} + k_i \int \tilde{x} dt \quad (26)$$

$$f_y = k_p \tilde{y} + k_d \dot{\tilde{y}} + k_i \int \tilde{y} dt \quad (27)$$

$$f_z = k_p \tilde{z} + k_d \dot{\tilde{z}} + k_i \int \tilde{z} dt \quad (28)$$

donde las ganancias del controlador son:

$$k_p = 3500 \quad (29)$$

$$k_d = 45 \quad (30)$$

$$k_i = 220 \quad (31)$$

donde las ganancias del controlador son:

$$k_p = 3500 \quad (28)$$

$$k_d = 45 \quad (29)$$

$$k_i = 220 \quad (30)$$

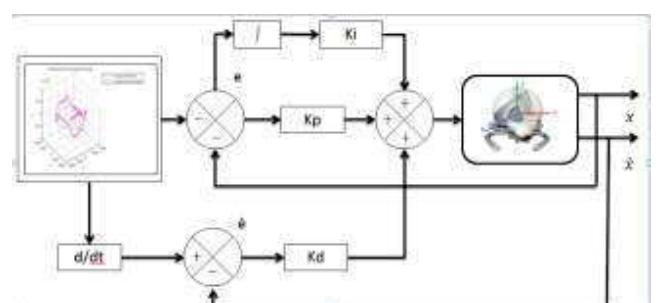


Fig. 3. Diagrama de bloques del controlador.

5. Plataforma experimental

La plataforma experimental consisten en un manipulador paralelo delta NOVINT FALCON (dispositivo haptico) en arquitectura abierta, es decir puede implementarse cualquier algoritmo de control a este dispositivo haptico, por lo que:

- dispone de la interfaz API de aplicación programable (CHAI3D)[8].
- entrada de fuerza y salida de coordenadas cartesianas para el control.
- desarrollo de código en Visual C++

Además:

- Computadora Intel Core i7, 2.67GHz y 6 GB de memoria RAM.

Y software especializado para el desarrollo de los experimentos:

- Windows 7 ultimate.
- Visual studio 2008 C++.
- Matlab R2011a.
- API de CHAI3D

6. Resultados experimentales

Los resultados experimentales consisten en que el dispositivo haptico siga una trayectoria de posición cartesiana y-z, tal como se muestra en la Fig. 4. Es muy similar la trayectoria deseada en color rojo y la trayectoria que describe el Novint Falcon en color azul, por lo que el control PID tiene buenos resultados, ya que el error es pequeño.

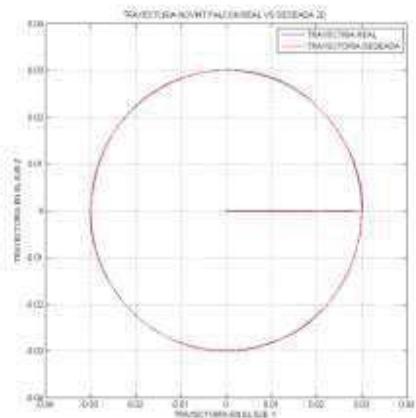


Fig. 4. Grafica de trayectorias.

La Fig. 5 muestra la gráfica en tercera dimensión de la trayectoria deseada (rojo) y la del manipulador Novint Falcon, y se puede observar los buenos resultados experimentales obtenidos.

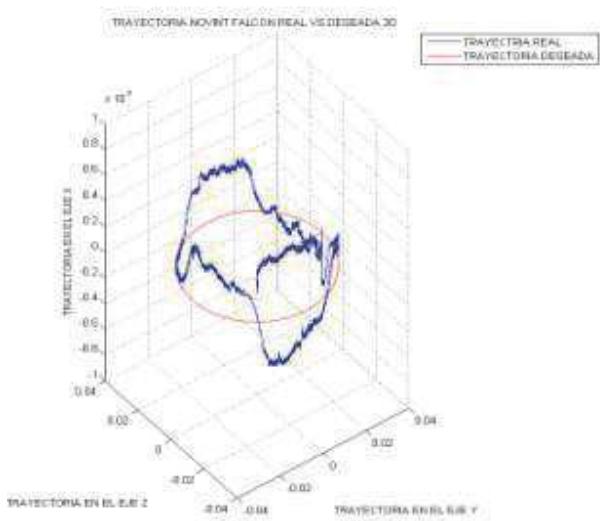


Fig. 5. Grafica de trayectorias en 3d

La Fig. 5 muestra la trayectoria deseada en x en color magenta y la trayectoria que describe el Novint Falcon en color azul, y se observa que no permanece en cero, sin embargo es muy pequeño por lo que se considera cero.

Las Figs. 6,7 y 8 muestra la comparación de la trayectoria deseada (azul) contra la real (amarillo), en cada uno de los ejes x, y y z, respectivamente; y se observa que con casi iguales, por lo que se está haciendo un buen seguimiento de la trayectoria.



Fig. 6. Trayectoria en x real y deseada.

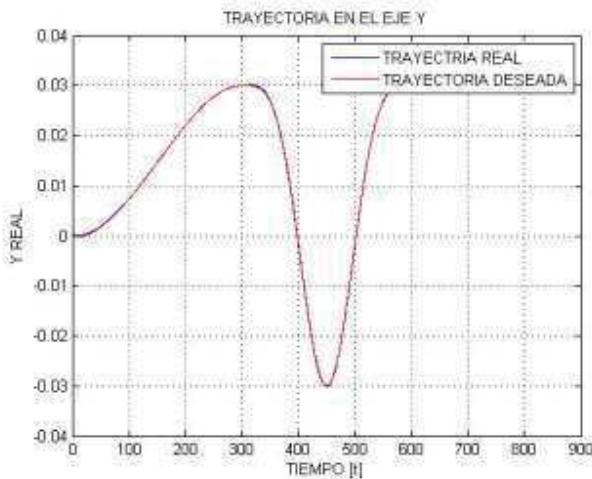


Fig. 7. Trayectoria en y real y deseada.

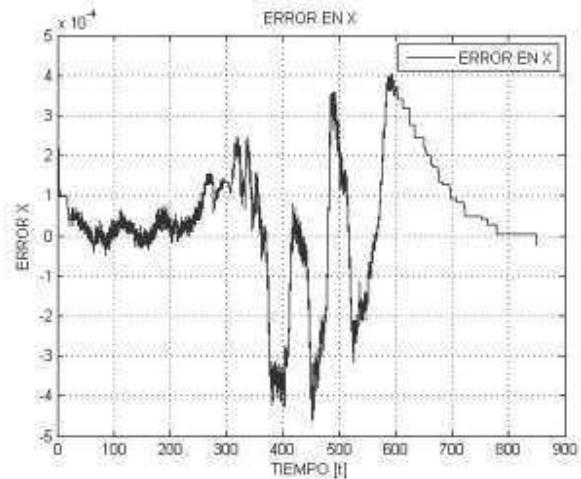


Fig. 9. Error en el eje X.

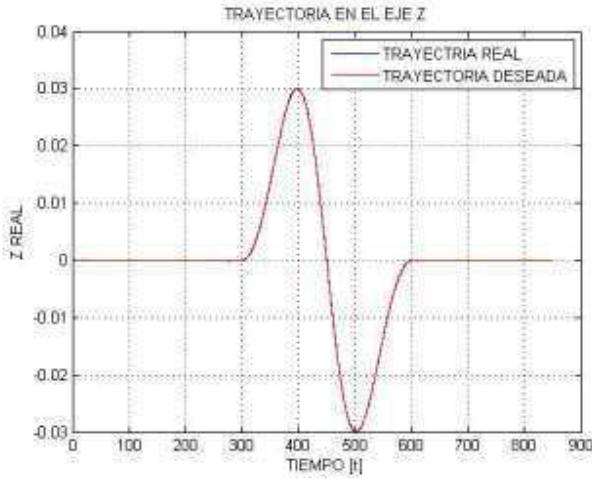


Fig. 8. Trayectoria en z real y deseada.

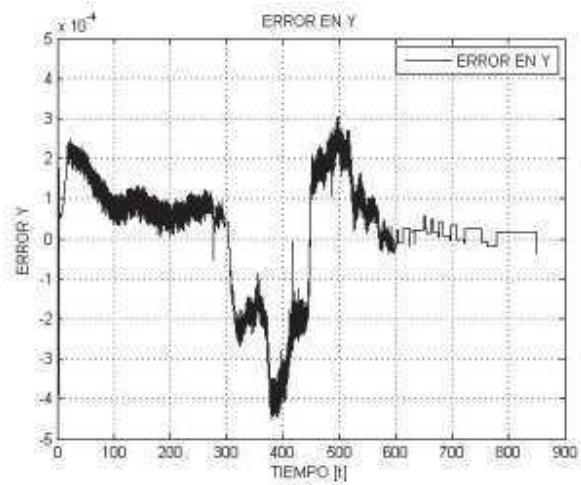


Fig. 10. Error en el eje Y.

Las Figs. 9, 10 y 11 muestran los errores cartesianos, en x, y z respectivamente, como se pueden observar los errores son muy pequeño, por lo que se concluye que se diseñó una buena trayectoria deseada y un buen controlador lineal.

Las Figs. 12, 13 y 14 muestran las fuerzas aplicadas al dispositivo haptico, en x, y y z; como se observa las tres fuerza están acotadas, por lo que el control funciona correctamente, aun siendo lineal. Comentar que este dispositivo haptico Novint Falcon no se tiene en la literatura el modelo matemático, sin embargo se espera en trabajos futuros experimentar con controladores no lineales.

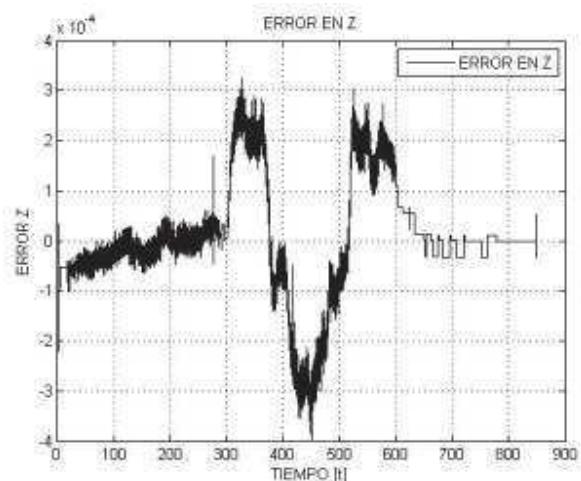


Fig. 11. Error en el eje Z.



Fig. 12. Fuerza aplicada eje x.



Fig. 13. Fuerza aplicada eje y.



Fig. 14. Fuerza aplicada ene eje z

7. Conclusiones

Se presentó una planificación de trayectorias en coordenadas cartesianas y además se implementó un control PID lineal para el seguimiento de las trayectorias. Como se observó tiene un buen desempeño el controlador lineal debido a que se programó una trayectoria suave. Lo que permitió tener buenos resultados experimentales, se espera en un futuro aplicarlo con un control no lineal. También se trabajó en un entorno virtual (Chai 3D) auxiliado de Visual Studio 2008 C++.

Los resultados experimentales son muy buenos lo que permitirá realizar trabajos futuros en teleoperación, diseño haptico y control haptico.

8. Referencias

- [1].- Youn K. Kim and Xiaolic Yang. “Real-Time Performance Analysis of Hand Writing Rehabilitation Excercise in Haptic Virtual Reality”, IEEE, Pags. 1357-1360, 2007.
- [2] Fernando A., Fuertes J., Martínez L, Szabo D. “Análisis de percepción en entornos acústicos envolventes con asistencia haptica para ciegos”, <http://www.haptics-e.org>
- [3] Jarillo A, Hernández I. y Domínguez O. “PHANTOM: una interfaz para retroalimentación kinestésica, entrenamiento y teleoperación con propósitos de diagnóstico y rehabilitación médica”.
- [4] Kevin R. Cleary C. “Closed- loop force for haptics simulation of virtual environments”. Haptics-e.
- [5] Parra, V. y Dominguez, O. “Constrained lagrangian-based force position control for haptics guidance”. Eurohaptics. 2003.
- [6] D.S. Kwon Y.S. Kim J.B. Song. B.H. Ryu. “Testing time domain passivity control of haptics enabled systems”. In Experimental Robotic, Springer Tracts in Advanced Robotics Series. Volume 5. 2003.
- [7] Corno M. and Zefran M. “Haptics Playback: Modeling, controller design and stability analysis”. Robotics: Science and Systems, The MIT Press, 2006.
- [8] Novint. Página de internet <http://home.novint.com/products/novintfalcon.php>. 2009.

Aplicación en Telefonía Móvil para el Control de un Motor Trifásico Mediante Comunicación Bluetooth

Aceves Flores Julio Cesar, Flores Salazar Erick Eduardo y Aguilar Anastacio José Carlos A.

Tecnológico de Estudios Superiores de Ecatepec, División de Ingeniería Mecatrónica e Industrial.
Av. Tecnológico S/n Esq. Av. Carlos Hank González. Col. Valle de Anáhuac, Ecatepec de Morelos.
Estado de México, C.P. 55210. Tel.50002330, mail: jcaa72@yahoo.com.mx, eefz180992@hotmail.com

Resumen

El presente trabajo describe el desarrollo de una aplicación de control para una máquina de corriente alterna trifásica, con la utilidad de las tecnologías de comunicación, así como el software libre para la implementación de una interfaz de usuario para la comunicación y control. La comunicación se realiza vía Bluetooth con un alcance de 30mts en campo abierto. El control del motor trifásico se entiende como manipular la dirección de giro, así como graduar la velocidad angular a voluntad del operador, de manera tal que se tenga un control total del motor sin la necesidad de estar en el lugar donde se encuentre el motor en tiempo real.

La contribución de este trabajo es dar a conocer las bondades que nos ofrecen las tecnologías de la información como es el celular y el software gratuito de Android, para el desarrollo de la aplicación de control del motor en CA trifásico de una manera inalámbrica y en tiempo real.

Palabras clave: Celular, Bluetooth, Arduino, Motor en CA trifásico, programación en App inventor, tiempo real.

1. Introducción

Cada día el desarrollo de aplicaciones con el sistema operativo Android, para tabletas o celulares, está revolucionando tal qué, hoy en día se puede desarrollar sistemas de control vía internet o Bluetooth. En la industria, algunas veces se requiere controlar maquinaria como por ejemplo grúas, elevadores, sistema de bandas, cadenas, mecanismos trasmisores de potencia, etc. Y que estos sistemas tienen como fuente de energía motriz por lo menos un motor de corriente eléctrica, ya sea de alterna o continua.

Existen ambientes de trabajo en la industria que obligan a controlar de manera remota a las máquinas, particularmente a los motores trifásicos de corriente alterna, ya que pueden estar en zonas de alto riesgo para los operadores de las mismas, como por ejemplo: zonas a altas temperaturas (hornos), zonas donde se manipulen gases o polvos tóxicos, excesivo ruido o incluso en ambientes radiactivos, etc.

Se plantea gobernar un motor trifásico por tener mayores aplicaciones en la industria, dado que existen motores que pueden generar y transmitir mayor potencia que los motores monofásicos. A demás tienen la cualidad de cambiar de giro y variar la velocidad en ambos sentidos, otro punto a favor, es que la alimentación trifásica ofrece costos reducidos y mayor eficiencia. [1][2][8].

Con este tipo de aplicaciones, se puede obtener un ahorro en la cuestión del tendido de cables para el control y monitoreo, así como también en la aplicación misma, ya que adquirir un sistema similar en National Instrument con su software LabView y hardware resulta costoso y más aún si es de tipo industrial.

2. Desarrollo del Proyecto

En esta sección se presenta el desarrollo de la interfaz de interacción entre el individuo y la máquina, es una interfaz simple y de fácil entendimiento, para crear dicha aplicación utilizamos App inventor que es una aplicación de Google Labs para crear aplicaciones de software para el sistema operativo Android. De forma visual y a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones en App Inventor permiten cubrir un gran número de necesidades básicas en un dispositivo móvil. App

inventor es un método de programación muy sencillo y de carácter visual, ya que la programación está orientada a objetos. La aplicación creada lleva por nombre **Moto_Master V1.0**, como se puede observar en la figura 1, la creación del ícono en la pantalla del teléfono celular.



Fig. 1. Ícono creado como Moto_Master.

2.1 Código de programación de la interfaz de usuario

App inventor permite una programación a base de bloques como a continuación se describe:

Primero se define la dirección MAC de nuestro módulo HC-05 para que nuestro celular pueda conectarse, como se muestra en la figura 2.



Fig. 2. Dirección Mac del dispositivo.

Lo siguiente es programar la pantalla de inicio de la aplicación, para esto es necesario tomar en cuenta las variables a controlar, como se mencionó en el apartado anterior las variables a controlar son el cambio de giro, variar la velocidad en ambos sentidos y frenar la máquina. Para esto es necesario diseñar una interfaz donde se configura algunos botones que se puedan deshabilitar y algunos otros puedan ser no visibles en la interfaz. En el siguiente bloque se

configuran los botones de arranque a tensión reducida, a plena tensión así como el freno del motor.

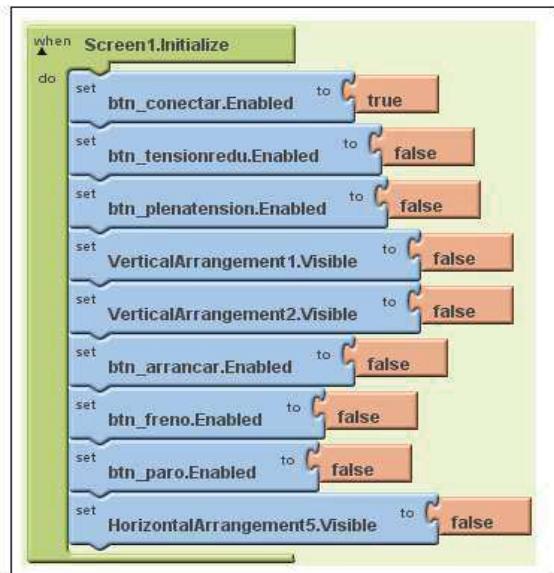


Fig. 3. Programación del menú de inicio.

El diseño de la aplicación termina con los bloques de los botones creados para realizar el control del motor en AC, mostrados en el siguiente bloque.

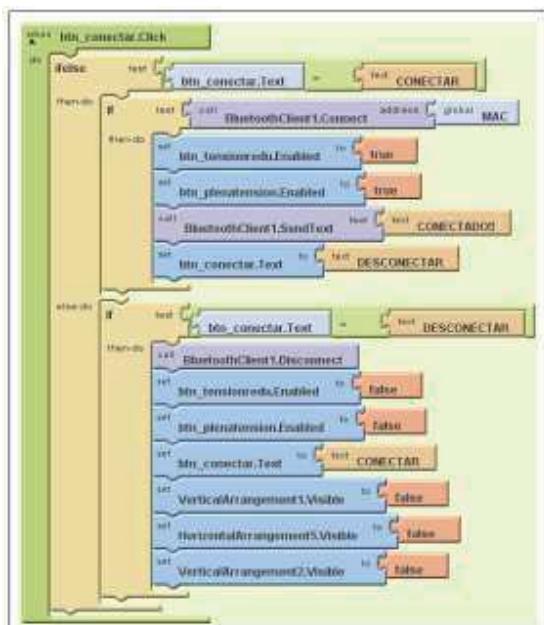


Fig. 4. Programación para la conexión Bluetooth.

La interfaz de usuario concluida se puede apreciar en la figura 5, donde se muestra el logo de la

escuela en la parte superior y los botones creados para realizar el control del motor a partir del teléfono celular vía Bluetooth.



Fig. 5. Interfaz de usuario concluida.

2.2 Programación del microcontrolador

Una vez creada la interfaz de usuario se programa el microcontrolador Arduino para realizar la comunicación entre el teléfono y el módulo Bluetooth. El Bluetooth utilizado es el HC-05 ya que ofrece una mejor relación entre precio y características, es un módulo que se puede configurar como Maestro-Eslavo, es decir además de recibir conexiones desde una PC, Tablet o teléfono inteligente, también es capaz de generar conexiones hacia otros dispositivos Bluetooth.

Para poder establecer una comunicación entre el dispositivo Android y el Bluetooth HC-05, se debe configurar el HC-05 de la siguiente manera:

- La velocidad de transmisión de datos, es de 9600 baudios ya que es la velocidad en que trabaja el Bluetooth del celular.
- El modo de operación del HC-05, ya que puede trabajar como esclavo o maestro.
- Asignación de contraseña del HC-05 para mantener una conexión única.

Para configurar el módulo se utilizan comandos AT, para este caso se diseña un programa donde se realiza la configuración de la velocidad de transmisión, así como las entradas y salidas, con la ayuda de monitor serial enviaremos la configuración. Existen comandos AT que nos permiten configurar el modulo, por ejemplo:

AT+BAUD=9600 Configura la velocidad del módulo a 9600 Baudios (bytes por segundo).

En el programa se configura, la contraseña del dispositivo, la velocidad de transmisión, una de las cualidades que se tienen en este tipo de programación es que la comunicación se realiza en tiempo real, esto significa, que la ejecución de los comandos para manipular al motor eléctrico son casi instantáneos (los tiempos de retardo entre comando y comando son despreciables) [3] [4] [9].

Enseguida se muestra el código para configurar la comunicación entre el celular y el microcontrolador.

```
void setup()
{
    Serial.begin(9600);
    pinMode(13,OUTPUT);
    digitalWrite(13,HIGH);
    delay(10000);
    digitalWrite(13,LOW);

    Serial.print("AT");
    delay(1000);

    Serial.print("AT+NAME");
    Serial.print(NOMBRE);
    delay(1000);

    Serial.print("AT+BAUD");
    Serial.print(BPS);
    delay(1000);

    Serial.print("AT+PIN+");
    Serial.print(PASS);
    delay(1000);
}

void loop()
{
    digitalWrite(13, !digitalRead(13));
    delay(500);
}
```

Fig.6 Configuración del módulo HC-05.

En el programa se realiza la comparación de los datos recibidos a través del puerto serial que envía el celular, lee esos datos y los compara con valores previamente programados, cuando un valor es enviado desde el celular y es igual a una valor programado en Arduino, se ejecuta una acción en particular.

2.3 Circuito de Control

El circuito de control es el dispositivo físico que realiza la interfaz entre el Bluetooth y el motor trifásico, esté se divide en tres etapas las cuales son:

1. Comunicación.
2. Modulación.
3. Potencia.

Comunicación: Consta de la conexión del módulo Bluetooth con el microcontrolador Arduino, para que se realice la comunicación entre el dispositivo móvil y el microcontrolador. Para realizar las pruebas de comunicación se utilizó el software Proteus, dado que no se cuenta con el módulo HC-05 en el simulador, se simula colocando una terminal virtual, donde el pin RX sería el pin RX del módulo, el pin TX sería el TX del módulo, el pin RTS sería el pin VCC del módulo y el pin CTS sería el pin GND del módulo.

En la figura 7, se muestra la conexión del microcontrolador Arduino y el HC-05.

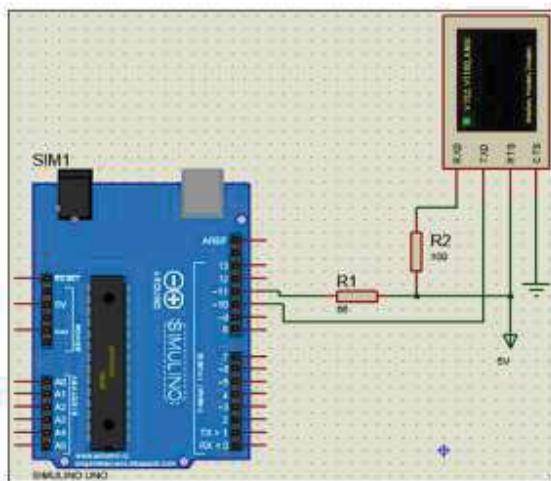


Fig. 7. Circuito de comunicación.

Uno de los pines del puerto del microcontrolador se configura de modo que reciba datos de forma serial proveniente del dispositivo Bluetooth, esto se realiza vía programa, los datos que

son recibidos son aquellos que se asignan desde la aplicación en el celular y son comparados dentro del programa para realizar la instrucción hacia el motor [4][9].

Modulación: El control de la velocidad del motor se realiza con la técnica de modulación de ancho de pulso sinusoidal (SPWM), donde se hace variar el ancho de cada pulso en proporción con la amplitud de una onda sinusoidal evaluada en el centro del mismo. Una de sus ventajas de este método es que el factor de distorsión (DF) y la armónica de orden bajo (LOH), se reducen en forma apreciable.

Las señales de control, se pueden apreciar en la figura 8, donde se generan comparando parte de la onda sinusoidal $v_r(t) = V_r \sin(\omega t)$, como señal de referencia con una onda portadora triangular de frecuencia f_c , mientras que la frecuencia de la señal de referencia f_r , determina la frecuencia f_o de la señal de salida y la amplitud pico A_r , de esta manera se controla el índice de modulación (M), y en consecuencia el voltaje RMS de salida (V_o). [3][6].

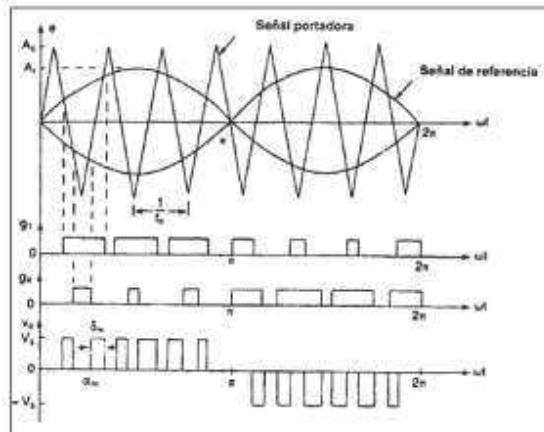


Fig. 8. Modulación de ancho de pulso sinusoidal.

La variación del voltaje de salida se realiza variando el índice de modulación de amplitud (M),

$$M = \frac{A_r}{A_c} \quad (1)$$

Si δ_m es el ancho del m -ésimo pulso, entonces se tiene que el voltaje de salida por cada fase con respecto al punto común: [3]

$$V_o = V_S \left(\sum_{m=1}^{2p} \frac{\delta_m}{\pi} \right)^{1/2} \quad (2)$$

En el microcontrolador se ha desarrollado un programa para generar, la (SPWM) tomando en cuenta la comparación de cada fase con una onda triangular generada internamente con el convertidor analógico digital del mismo microcontrolador, donde se genera y se compara de tal manera que el resultado es la generación de la señal modulada por ancho de pulso en cada fase, la frecuencia de modulación es de 350 a 400 Hz. Se utilizan tres pines del microcontrolador para generar tres salida, estas señales se aplican a un opto-triac, este dispositivo debe de tener la característica de que se apague cuando cruce por cero de forma bilateral con cada fase, para disparar a los triacs de potencia como se puede apreciar en el diagrama de la figura 9.

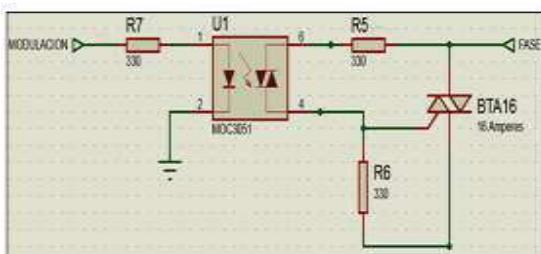


Fig. 9. Circuito de aislamiento entre el microcontrolador y la potencia.

Potencia: el motor se controla con el MOC3051, el cual es el encargado de recibir la señal modulada proveniente del microcontrolador, pines 1 y 2, mientras que por los pines 6 y 4 es la conexión del opto-triac, este es de baja potencia el cual tiene la función de activar el triac de potencia, quienes controlan el motor trifásico, la etapa de potencia se compone por tres triacs de 16 amperes, el circuito utilizado para manipular el motor trifásico se muestra en la figura 10. Se tiene protecciones térmicas que protegen al motor de sobrecarga, los relevadores mostrados en el diagrama marcados con los números del 5 al 10, estos soportan una corriente de 12 Amperes [1][2][3][8].

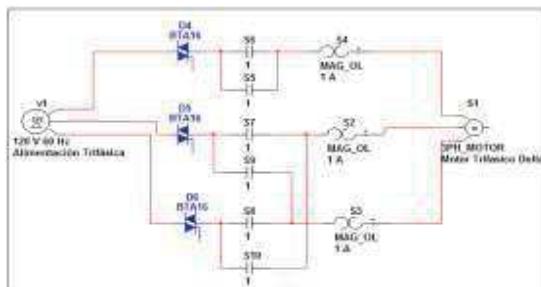


Fig. 10. Circuito de conexión del motor trifásico.

El motor a controlar es el que se muestra en la figura 11, y sus características son:

- Alimentación Trifásica.
- 220/440 volts.
- 2 Amperes a 220Vca, 0.8 a 440Vca.
- 1750 RPM.
- Armazón NEMA 56.
- Peso de 8 kg.
- Aprobado por la Norma Oficial Mexicana.



Fig. 11 Motor trifásico marca siemens con el sistema de freno.

2.4 Freno de motor

El freno se realiza en dos etapas, primeramente cuando se oprime el botón del freno inicia la primera etapa se reduce la velocidad a la mínima frecuencia, la segunda etapa se anula la modulación de ancho de pulso, para que el motor entre en funcionamiento libre, es decir con la inercia que la propia flecha mantiene en tres segundos, al terminar este tiempo se activa un pistón que se conecta a un par de balatas, las cuales ejerce presión sobre un disco colocado en la flecha del motor lo que provoca que se detenga el motor en 3seg, como se observa en la figura 11, se observa el esquema utilizado para el freno del motor, cavé mencionar que el pistón utilizado es de tipo eléctrico a 12volts esté es controlado por medio de un relevador en la tarjeta principal del control [7][8].

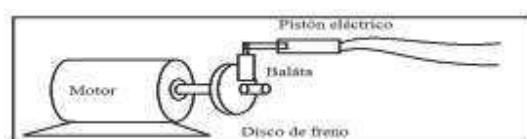


Fig. 12. Freno del motor con disco y pistón.

El circuito completo se muestra en la figura 13, este se compone de la tarjeta Arduino y el circuito del control con relevadores que controlan directamente al motor trifásico y el freno.

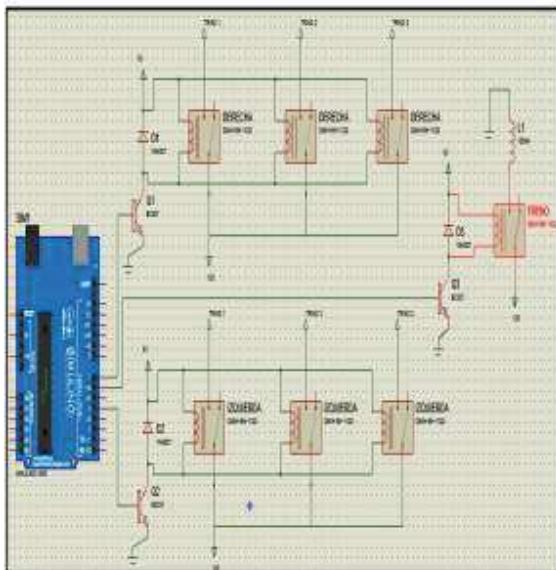


Fig. 13. Circuito completo de control.

3. Análisis de resultados

Los resultados obtenidos de este artículo se muestran en las figuras 11 y 15, las cuales describen los avances de cada uno de los objetivos alcanzados, el resultado principal es la manipulación del motor trifásico a voluntad del operador desde cualquier distancia que no salga del rango de 50mts, a partir del celular con la implementación del software libre como es el sistema operativo Android, sin la implementación de software sofisticado y sin la utilidad de tarjetas costosas o licencias de algún tipo.

En las imágenes se muestra la etapa de potencia por tres triacs junto con los optoacopladores en la figura 10, del lado izquierdo, mientras que por el lado derecho inferior se muestra el Bluetooth, mientras que en el fondo se muestra la etapa de los relevadores que controlan a los contactores, finalmente se muestra la tarjeta del microcontrolador Arduino. El motor de prueba utilizado es un motor de CA trifásico de 1HP.



Fig. 15. Circuito completo en protoboard vista lateral

4. Conclusiones

Las conclusiones del trabajo son específicamente:

1. Se muestra un gran desempeño en el control del motor en ambos sentidos de giro.
2. Se logra comunicar el microcontrolador y el celular vía Bluetooth.
3. La programación por bloques lo hace más compacto y entendible para programar la interface en el celular.
4. Se alcanzó el objetivo deseado.
5. Se está trabajando en la mejoría del sistema en conjunto.

Referencias

- [1] Maloney Timothy J. "Electrónica Industrial Moderna", Prentice Hall, México, 5^a edición 2006.
- [2] Gualda. A.J, Martínez S. Martínez P.M. "Electrónica Industrial", Marcombo, Barcelona España, 2^a edición, 1992.
- [3] Muhammad H. Rashid, "Electrónica de Potencia", Prentice Hall, 3^a edición, 2004.
- [4] Torrente Artero Óscar, "Arduino Curso práctico de formación", Alfaomega, 1^a edición, 2013.
- [6] W. Hart Daniel, "Electrónica de Potencia", Prentice Hall, 2001.
- [7] D. Petruzzella Frank, "Electric Motors Control Systems", Mc Graw Hill, 1^a edición, 2010.
- [8] J. Chapman Stephen, "Máquinas eléctricas", Mc Graw Hill, 5^a edición, 2012.
- [9] <http://arduino.cc/>

Laboratorio Virtual de Escenarios Industriales para el Adiestramiento en el Área de Automatización y Control

Barreiro Fabián A., Barreiro Aldo J. y Ortega-Moody Jorge Alberto

Academic Division of ANT-Automation, Pittsburgh PA, USA. jorgemoody@gmail.com

Resumen

La automatización, es un área que constantemente se actualiza al incorporar nuevos desarrollos tecnológicos. Estos cambios, exigen una constante capacitación que va desde la academia, hasta el ámbito industrial. Una de las limitaciones en la academia, es la falta de infraestructura que permita al estudiante realizar pruebas en un ambiente industrial, ya que el tipo y número de actuadores, sensores y procesos, están limitados a la infraestructura del laboratorio. Por esta razón, se plantea el desarrollo de un laboratorio industrial virtual, en donde los componentes posean características físicas y eléctricas de la industria, ofreciendo un comportamiento similar a los reales. Con este desarrollo, las personas en capacitación, podrán conectar controladores al laboratorio industrial virtual por medio de OPC y programar secuencias complejas de automatización en industrias reales virtualizadas sin la necesidad de comprometer la seguridad del proceso o de las personas involucradas al modificar los parámetros y secuencias de control.

Palabras clave: Automatización, PLC, Virtual Commissioning, Laboratorio virtual.

1. Introducción

La incorporación de nuevas tecnologías y desarrollos tecnológicos en el área de automatización y control, exigen una constante actualización del personal involucrado en este ramo. La base de esta capacitación, inicia en los laboratorios de instituciones educativas y centros de capacitación de personal en temas relacionados a la automatización. En estos laboratorios, se aplican los conocimientos teóricos adquiridos en el salón de clases. Desafortunadamente, los laboratorios de este tipo, se ven restringidos en infraestructura, teniendo como

limitante el uso de solamente algunos actuadores y sensores; y en el mejor de los casos, sets de pruebas en donde se involucran ciertos mecanismos para realizar las prácticas. Otra fuente de entrenamiento son las mismas empresas, las cuales tienen programas de capacitación interna impartido por los expertos en el área. Desafortunadamente en esta capacitación, se limita la interacción con la planta, debido a que en la mayoría de procesos, cualquier error de programación en los controladores, representa pérdidas económicas y en el peor de los casos; desastres que comprometen la seguridad del personal involucrado.

En cualquiera de los dos casos de entrenamiento (instituciones o empresas), el personal involucrado tiene como objetivo aprender y/o actualizarse en alguno o varios de los siguientes tópicos:

- PLCs
 - Lenguaje escalera.
 - Conocer y programar en ambientes de programación comerciales (AB: RsLogix, Siemens: Stp7, etc.).
- Comunicación.
 - OPC
 - Modbus
 - DLLs
 - Protocolos comerciales (DeviceNet, ControlNet, Profibus, etc.).
- SCADAS
 - Comerciales (RSview32, Simatic, etc.)
 - Desarrollar SCADAS mediante lenguajes de alto nivel(C#, VB.NET)

Por otro lado, el software y hardware dedicados a los gráficos 3D y simulaciones, han tenido un desarrollo exponencial debido a su aplicación en los videojuegos [1]. Gracias a este desarrollo, estas herramientas se conjuntan para realizar espacios virtuales en primera y tercera persona enfocados a una aplicación industrial.

En este trabajo, se plantea el desarrollo de laboratorios virtuales (LV) con escenarios industriales para su aplicación en la enseñanza de temas relacionados a la automatización y control. Con este desarrollo, las instituciones de enseñanza tendrán una transferencia de conocimiento sin la necesidad de adquirir hardware y actuadores para su implementación. Además, los participantes podrán automatizar líneas y procesos reales con escenarios que varían la complejidad de secuencias dependiendo del nivel de conocimientos del aprendiz. Otra ventaja de esta herramienta es el poder operar un proceso, eliminando el riesgo económico y de seguridad que representa cuando se alteran parámetros de control.

Este laboratorio puede interactuar con cualquier marca de PLCs que posea comunicación por OPC. Esto ofrece una flexibilidad en la impartición de cursos de control y automatización industrial. Un entorno basado en la simulación de procesos, en donde se representen aspectos físicos y comportamientos dinámicos de los modelos reales, tales como objetos, actuadores, sensores y el entorno [2]. Con este desarrollo, se pretende que las instituciones educativas y centros se adecuen a los diferentes tipos de controladores que existen en el mercado. Los PLCs también pueden utilizar sus I/O reales para que estas interactúen con el proceso virtual que sirve como escenario de entrenamiento. Estas flexibilidades hacen que el proceso de aprendizaje sea muy apegado a la realidad.

2. Metodología

La metodología empleada en el diseño es la que se observa en la Fig. 1, en donde se inicia con una serie especificaciones del proceso a virtualizar. Una vez se obtienen las características físicas del proceso, así como el comportamiento dinámico de cada uno de los elementos como mecanismos, sensores y actuadores; se procede al dibujo CAD de estos. El tercer paso es el de introducir los diseños CAD en un motor de física para poder asignar comportamientos dinámicos a cada elemento. Por último, se realizan pruebas de comportamiento de cada elemento para asegurar que su comportamiento es el que se requiere para que este se comporte lo más apegado a la realidad posible.

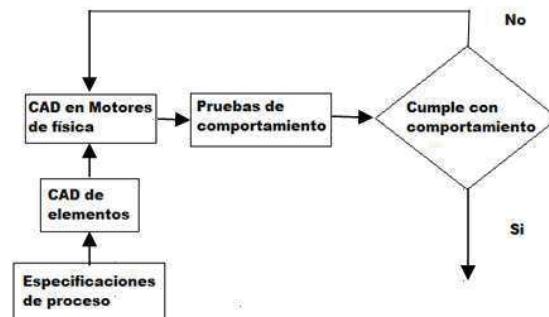


Fig. 1. Metodología.

3. Diseño e implementación

El laboratorio virtual está construido en dos bloques básicos, sensores y actuadores, los cuales envían y reciben señales digitales hacia el PLC. Estas señales son emuladas de los dispositivos virtuales como motores, pistones y HMIs por mencionar algunos. El comportamiento de estos dispositivos depende del grado de sofisticación del modelo, pudiéndose comparar en gran medida al comportamiento de los dispositivos físicos reales.

3.1 Arquitectura del sistema

La arquitectura del laboratorio virtual (Ver Fig. 2), está basada en un servidor autoritario que simula el mundo real y procesa las señales provenientes del PLC y de otros clientes de red [3].

El cliente está clasificado en dos tipos, quienes requieren solamente la simulación de señales (PLCs, HMIs botoneras, alarmas y señalamiento) y quienes requieren tener un entorno visual de la simulación (Persona).

Los clientes envían y reciben señales y comandos hacia el servidor que continuamente actualiza el estatus y resultados de la simulación. Los clientes por sí mismos no pueden hacer cambios directamente en la simulación. En vez de esto, le indica al servidor lo que requiere realizar, y el servidor procesa la petición presentando los resultados al cliente. De esta forma, existe una separación y aislamiento entre lo que el cliente quiere y lo que actualmente sucede en la simulación. Esto permite que el servidor escuche cada petición de cada

cliente antes de tomar la decisión de cómo actualizar el estado de la simulación.

La simulación de la física de los objetos, se realiza mediante un motor de física independiente al servidor, esto ayuda a que el servidor no realice cálculos excesivos y se enfoque a la atención de los clientes.

Una potencial desventaja del uso servidores autoritarios es el tiempo en que los mensajes tardan en viajar por la red. Por ejemplo, si un usuario interactúa con un cuerpo rígido, empujándolo y el mensaje de respuesta del server toma una décima de segundo, entonces esta demora será perceptible por el usuario. Una solución a este problema es la técnica llamada client-side prediction [4]. La aplicación de esta técnica permite al cliente realizar actualizaciones del estado de su versión local de la simulación y luego recibir correcciones del servidor autoritario.

Para lograr un tiempo de muestreo de al menos 20 milisegundos en las señales enviadas por el modelo al PLC se implementaron mecanismos de mensajes independientes para las señales eléctricas del modelo.

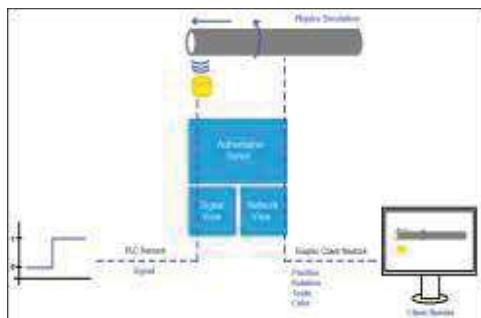


Fig. 2. Mecanismos independientes en el Sistema de mensajes para la red de PLC y los clientes gráficos.

3.2 Implementación

Para la implementación, se seleccionó un escenario de una planta productora de tubos de acero. En este escenario, se implementaron sensores de proximidad, capacitivos e inductivos, asimismo actuadores como motores y pistones. También se modelaron los sistemas mecánicos accionados por los actuadores. Tanto para los actuadores como a los mecanismos, se les agregó sonidos característicos del proceso y de su funcionamiento para tener un ambiente virtual más apegado a la realidad.

En lo que corresponde a Hardware, se utilizaron dos tecnologías para permitir que el LV tuviera escalabilidad para procesar la física de modelos complejos en la simulación y que al mismo tiempo, soportara múltiples clientes en tiempo real.

La primera es un mecanismo que permite la distribución de la simulación y la carga de los clientes en múltiples servidores autoritarios. Si el ambiente industrial puede ser dividido en varios escenarios, el servidor puede ejecutar la simulación de uno o más escenarios y ser cliente para el resto. También ofrece la opción de balancear la carga de los clientes en diferentes servidores ver Fig. 3.

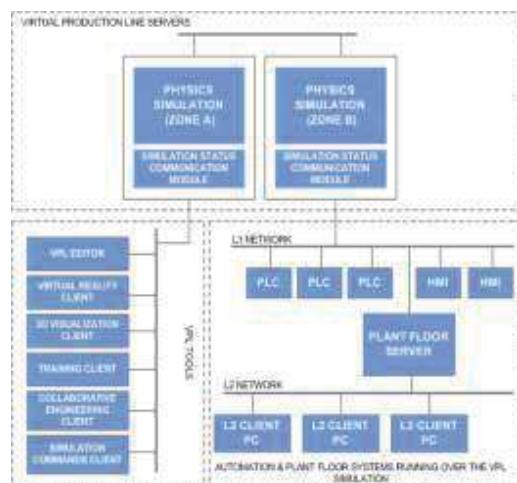


Fig. 3. Distribución de clientes.

La otra tecnología utilizada es un motor de física que se ejecuta en un hardware especializado para esto, se utilizó el NVIDIA® PhysX®. El cual posee procesadores potentes diseñados específicamente para este propósito. Con esta tecnología, se incrementa en alto grado la potencia de procesamiento de la física del entorno virtual y sus componentes (NVIDIA Corporation, 2014).

3.3 Tipos de entrenamiento

Se implementaron dos tipos de entrenamiento, en el cuarto de control y el entrenamiento dentro de la planta. El primero consiste en la recreación y simulación del cuarto de control en donde el operador monitorea la planta por medio de un SCADA y programa los PLCs, teniendo como entorno las pantallas y controles para manipular la operación del proceso. También es posible simular el despliegue de las cámaras de seguridad que monitorean la planta (Fig. 4).



Fig. 4. Cuarto de control virtual.

La segunda forma de entrenamiento es la simulación de estar dentro de la planta. Este modo es utilizado cuando el ingeniero de campo tiene que interactuar con otros objetos que son parte de la escena tales como botoneras e interfaces. En este modo el ingeniero de campo se mueve dentro de la planta en modo de primera persona (Fig. 5 y 6), ofreciendo la perspectiva de estar en el ambiente industrial.



Fig. 5. Perspectiva primera persona.



Fig. 6. Manipulación de objetos en primera persona.

El LV permite el entrenamiento en sus dos modalidades simultáneamente. Un operador puede utilizar el modo de cuarto de operación mientras un ingeniero de campo se entrena en la escena, cabe mencionar que esta interacción en este escenario virtual puede ser accedido por personas con ubicación geográfica diferente por medio de la red. Cuando se utilizan estas dos modalidades, el operador puede visualizar el avatar del ingeniero de campo que está entrenándose y moviéndose dentro de la escena como se muestra en la Fig. 7.



Fig. 7. Vista de clientes gráficos en tercera persona.

4. Pruebas y resultados

El PLC que se utilizó para interactuar con el laboratorio virtual fue un SLC 500 de Allen Bradley, el cual se comunica por medio de OPC. Asimismo, se conectó una botonera de grúa viajera a las I/O del PLC. Esta botonera interactúa con la grúa viajera virtual dentro del escenario, ofreciendo al ingeniero de campo, el entrenamiento de utilizarla con cargas que poseen las características físicas reales.

El modelo gráfico está formado por 3.2 millones de vértices, 2.6 millones de triángulos y 194 texturas de realismo fotográfico. Desde el punto de vista de la simulación física, el modelo cuenta con 340 cuerpos rígidos, 911 puntos de contactos con 283 colisionadores estáticos y 425 dinámicos (Figura 5).

El software del servidor autoritario y los módulos de comunicación se ejecutaron sobre una computadora de 16GB de memoria RAM, con procesador Intel i7-2600K @ 3.40Ghz, Windows 7 64 bits y dos placas GeForce GTX TITAN Black.

Se conectó a la simulación un cliente gráfico de 12 GB de memoria RAM, con procesador Intel Core i7 x880 @ 3.33Ghz, Windows 8.1 x64 y 2 placas gráficas AMD Raedon HD 5700.

Se utilizó una red Gibabit Ethernet para la conexión de los clientes gráficos, servidor autoritario y PLC.

En la corrida, se realizó la simulación del procesamiento de un lote de 150 tubos sobre la línea de teste de calidad simulando un equipo de ultrasonido de tubo completo (UTPA).

La utilización de la CPU/GPU en el servidor autoritario fue constante y menor a 1.5% en toda la utilización. La frecuencia de muestreo de las señales del modelo se mantuvo constante a 0.9 milisegundos (1.111 muestras por segundo).

Con el cliente gráfico tuvo un tiempo de refresco mínimo de 70 fps (frames por segundo) la vista relativa ocupaba toda la línea (3.2 millones de vértices).

La utilización de la red no superó el 5% de su capacidad y los paquetes hacia el servidor OPC se transmitieron en un ratio constante igual al muestreo de señales.

5. Conclusiones

El LV implementado, es una herramienta de entrenamiento que permite capacitar a operadores de planta, ingenieros de campo y estudiantes; en un ambiente virtual utilizando controladores reales conectados a este ambiente. Esta última característica ofrece al estudiante conocer e implementar rutinas de automatización en procesos complejos con la

tranquilidad de no comprometer la seguridad de las instalaciones ni del personal involucrado.

Gracias a la retroalimentación del sistema, el operador puede visualizar los errores en el proceso, dándole la oportunidad de cambiar parámetros y aprender de sus propios errores.

Como trabajo futuro, se diseñarán nuevos escenarios de procesos industriales y se incorporará un casco de realidad virtual para aumentar el grado de realidad de la simulación.

Referencias

- [1] Koutsabasis P., Vosinakis S., Malisova K., & Paparounas N. “*On the Value of Virtual Worlds for Collaborative Design*”, *Design studies*, Elsevier Science, Vol. 33, Num. 4, Pag. 357-390, 2012.
- [2] Bickel D. “*Realtime Simulation and VR-Tools in the Manufacturing Industry*” en *Virtual Reality for Industrial Applications*. Springer, US, Primera edición, Pag. 123-138, 2008.
- [3] Claypool M. and Claypool K. “*Latency can kill: Precision and deadline in online games*”. Proceedings of the first annual ACM SIGMM conference on Multimedia systems, New York, USA, ACM, Pag.215-222, 2010.
- [4] Ximbo J. “*Latency and scalability: a survey of issues and techniques for supporting networked games*”, 13th IEEE International Conference on Networks, Malaysia, IEEE. 2005.

Desarrollo de una Plataforma de Simulación Basada en el Robot Antropomórfico K-680

González-Romero Dulce Isabel, Sánchez-Sánchez Pablo y Castañeda-Camacho Josefina

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica
Puebla, México
dglez.romero@gmail.com, leitable@ece.buap.com y josefinacastaneda@ece.buap.com

Resumen

El objetivo principal del artículo es presentar una plataforma gráfica de simulación que brinde las herramientas necesarias para observar el comportamiento de un robot antes dos diferentes leyes de control. Esta interfaz le permite a los usuarios interactuar con un robot tridimensional basado en el robot K-680 de Steren®, y observar a través de gráficos tridimensionales la respuesta del robot ante cada ley de control. La obtención del modelo dinámico y el análisis de estabilidad se presentan con el debido rigor matemático.

Palabras clave: Control PD, Control PID, Modelo dinámico, Interfaz Gráfica, Simulación.

1. Introducción

El uso de los simuladores en el área de la robótica permite modificar, en base a la observación de los resultados, los parámetros necesarios para evaluar y obtener información que permita su implementación en una planta real. Hoy en día el software usado para diseñar simuladores es variado, y la elección depende de los requerimientos del usuario. Su aplicación se enfoca, en gran medida al uso académico, ya que permiten estudiar varios tipos de respuestas, sin necesidad de adquirir un robot físico. La exactitud de los datos que un simulador entrega depende mucho de la descripción matemática del modelo permitiendo facilitar una aplicación física. Por lo cual el software que se utilice debe de tener las herramientas necesarias para un análisis físico. El uso de softwares de diseño facilitan esta tarea, entregando no solo datos fiables, si no también permitiendo tener una imagen 3D, lo cual proporciona la percepción de movimiento.

El análisis de los datos se realizan en plataformas matemáticas, por la necesidad de

representar de forma gráfica todas las variables, entonces resulta primordial la comunicación entre las dos plataformas, esto para generar un simulador más robusto y completo.

Este artículo está organizado de la siguiente forma: en la segunda sección se aborda la obtención del modelo dinámico. Las estructuras de control se presentan en la tercera sección, mientras que el desarrollo del simulador se describe en la sección 4. Finalmente en la última sección se presentan algunas conclusiones.

2. Modelo dinámico

En la figura N° 1 se presenta el robot, el cual se describirá matemáticamente.



Fig. 1. Robot antropomórfico

Se debe considerar que, el modelo dinámico de un robot de n grados de libertad consiste en una ecuación diferencial ordinaria vectorial, generalmente de segundo orden, en las posiciones articulares o cartesianas. Uno de los procedimientos más empleados, es el basado en las *Ecuaciones de movimiento de Lagrange*.

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} = \tau \quad (1)$$

donde $\mathcal{L}(q, \dot{q})$ es el Lagrangiano, el cual está definido como la energía cinética $\mathcal{K}(q, \dot{q})$ y la potencial $\mathcal{U}(q)$:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q) \quad (2)$$

Al aplicar la ecuación de movimiento de Euler-Lagrange se obtiene como resultado el modelo dinámico del sistema. El modelo dinámico es la representación matemática del sistema que describe su comportamiento a cualquier estímulo, sea interno o externo. El modelo dinámico para un robot de n grados de libertad está definido como [1,2]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (3)$$

donde q, \dot{q}, \ddot{q} son la posición, velocidad y aceleración articular, respectivamente. $M(q)$ es la matriz de masas e inercias, $C(q, \dot{q})$ es la matriz de Coriolis y fuerza centrípeta, $g(q)$ es el par gravitacional y τ es el par aplicado.

3. Control de posición

La obtención del modelo de energía, es de gran relevancia gracias a la información que proporciona, haciendo posible aplicar una ley de control que permita posicionar el robot en las coordenadas articulares deseadas.

Unos de los objetivos principales del simulador es cumplir con el objetivo de control de posición, el cual consiste en determinar una función vectorial τ a través de la aplicación de la ecuación dinámica de un robot de n grados de libertad y dada una posición articular deseada q_d que se supone constante, de forma que las posiciones q asociadas a las coordenadas articulares del robot lleguen asintóticamente a la coordenada deseada [1]:

$$\lim_{t \rightarrow \infty} q(t) = q_d \quad (4)$$

donde $q_d \in \mathbb{R}^n$ es un vector constante.

A continuación se presenta la demostración y la justificación del uso del control PD con compensación de gravedad y del control PID, así como los resultados generados al aplicar cada ley de control en el robot virtual.

3.1 Control PD con compensación de gravedad

La ley de control PD con compensación de gravedad está representada por [1,3]:

$$\tau = K_p \tilde{q} - K_v \dot{q} + g(q) \quad (5)$$

donde K_p y $K_v \in \mathbb{R}^{n \times 1}$ son matrices simétricas definidas positivas que representan la ganancia proporcional y derivativa, respectivamente. La ecuación (5) demuestra la necesidad del conocimiento previo del modelo dinámico del robot, ya que es fundamental el uso del par gravitacional $g(q)$.

Sustituyendo el control PD en la ecuación (3), que representa el modelo de energía, se obtiene:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = K_p \tilde{q} - K_v \dot{q} + g(q) \quad (6)$$

donde el par gravitacional se puede eliminar dada su igualdad. Para saber si la ecuación que describe la ley de control es aceptable, se procede a estudiar la estabilidad de los puntos de equilibrio del sistema. Empleando el teorema de Lyapunov, se considera la siguiente función candidata [1,3]:

$$V(\tilde{q}, \dot{q}) = \mathcal{K}(q, \dot{q}) + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (7)$$

donde la energía cinética $\mathcal{K}(q, \dot{q})$ es definida positiva en \dot{q} y adem\'as K_p es una matriz definida positiva en \tilde{q} , provocando que la función candidata de Lyapunov, también sea definida positiva $V(\tilde{q}, \dot{q}) > 0$. [1,4]

Considerando la ecuación (8), la función candidata, se puede representar como se muestra en la ecuación (9) [1,3]:

$$\mathcal{K}(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (8)$$

$$V(\tilde{q}, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (9)$$

Cuya derivada temporal, ecuación (10), se puede reescribir sustituyendo la variable de estado \dot{q} (ecuación (11)) y realizando el álgebra necesaria, obteniendo la ecuación (12).

$$\dot{V}(\tilde{q}, \dot{q}) = \tilde{q}^T K_p \tilde{q} + \dot{q}^T M(q) \dot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} \quad (10)$$

$$\dot{q} = M(q)^{-1} [K_p \tilde{q} - K_v \dot{q} - C(q, \dot{q}) \dot{q}] \quad (11)$$

$$\dot{V}(\tilde{q}, \dot{q}) = -\dot{q}^T K_v \dot{q} - \frac{1}{2} \dot{q}^T [\dot{M}(q) - 2C(q, \dot{q})] \dot{q} \quad (12)$$

Ahora, teniendo en cuenta la siguiente propiedad:

$$\dot{M}(q) - 2C(q, \dot{q}) = 0 \quad (13)$$

Se obtiene la derivada temporal resultante de la función candidata, presentada a continuación:

$$\dot{V}(\tilde{q}, \dot{q}) = -\dot{q}^T K_v \dot{q} \quad (14)$$

Demostrando así, que la función candidata de Lyapunov analizada cumple con ser definida positiva $V(\tilde{q}, \dot{q}) > 0$, continuamente diferenciable $V(\tilde{q}, \dot{q}) \in C^{\infty}$ y además su derivada temporal es semidefinida negativa $\dot{V}(\tilde{q}, \dot{q}) \leq 0$, por lo tanto se dice que el origen es estable en el sentido de Lyapunov.[1,4,5]

3.2 Control PID

Con el propósito de satisfacer el objetivo de control y más aún para intentar llevar a cero el error de posición sin necesidad de proponer ganancias relativamente grandes, como en el control PD, se ha añadido una componente integral, justificado el uso del control Proporcional-Integral-Derivativo (PID). La ley de control PID puede expresarse de la siguiente manera [1]:

$$\tau = K_p \tilde{q} - K_v \dot{q} + K_i \int_0^t \tilde{q}(\sigma) d\sigma \quad (15)$$

Ya que q_d es constante, se establece la existencia y unicidad del punto de equilibrio para la ecuación de malla cerrada [1]:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = K_p \tilde{q} - K_v \dot{q} + K_i \xi \quad (16)$$

donde $\dot{\xi} = \tilde{q}$. El punto de equilibrio del sistema está localizado en [1]:

$$\begin{bmatrix} \xi \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} K_i^{-1}g(q_d) \\ 0 \\ 0 \end{bmatrix} \quad (17)$$

Para facilitar el estudio de la estabilidad en el sentido de Lyapunov, es necesario hacer una serie de cambios de variable, el primer cambio de variable reubicará el punto de equilibrio colocándolo en el origen, esto es [1]:

$$z = \xi - K_i^{-1}g(q_d) \quad (18)$$

Por lo tanto la ecuación de malla cerrada representada en términos del vector de estados y considerando el anterior cambio de variable, se presenta como:

$$\frac{d}{dt} \begin{bmatrix} z \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \tilde{q} \\ -\dot{q} \\ M(q)^{-1} \left[K_p \tilde{q} - K_v \dot{q} + K_i z + g(q_d) - C(q, \dot{q})\dot{q} - g(q) \right] \end{bmatrix} \quad (19)$$

Se puede observar que la ecuación anterior es autónoma y su único equilibrio se encuentra en el origen. Prosiguiendo con el análisis de estabilidad se

propone el siguiente cambio de variable, considerando $\alpha > 0$ [1]:

$$\begin{bmatrix} \omega \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \alpha I & I & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} z \\ \tilde{q} \\ \dot{q} \end{bmatrix} \quad (20)$$

Con lo que se puede presentar la nueva forma de la ecuación de malla cerrada:

$$\frac{d}{dt} \begin{bmatrix} \omega \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \alpha \tilde{q} - \dot{q} \\ -\dot{q} \\ M(q)^{-1} \left[\left(K_p - \frac{1}{2}K_i \right) \tilde{q} - K_v \dot{q} + \frac{1}{2}K_i \omega + g(q_d) - C(q, \dot{q})\dot{q} - g(q) \right] \end{bmatrix} \quad (21)$$

Para aplicar el teorema de estabilidad de Lyapunov, se propone la siguiente función candidata, la cual debe cumplir con ser definida positiva [1]:

$$V(\tilde{q}, \dot{q}, \omega) = \frac{1}{2} \begin{bmatrix} \omega \\ \tilde{q} \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} \frac{1}{\alpha K_i} & 0 & 0 \\ 0 & \alpha K_i & -\alpha M(q) \\ 0 & -\alpha M(q) & M(q) \end{bmatrix} \begin{bmatrix} \omega \\ \tilde{q} \\ \dot{q} \end{bmatrix} + \frac{1}{2} \tilde{q}^T \left(K_p - \frac{1}{\alpha} K_i \right) \tilde{q} + U(q_d - \tilde{q}) - U(q_d) + \tilde{q}^T g(q_d) \quad (22)$$

donde:

$$\frac{1}{2} \tilde{q}^T \left(K_p - \frac{1}{\alpha} K_i \right) \tilde{q} + U(q_d - \tilde{q}) - U(q_d) + \tilde{q}^T g(q_d) \quad (23)$$

es definida positiva en \tilde{q} sí [1]:

$$\lambda_{\min} \left\{ K_p - \frac{1}{\alpha K_i} \right\} > k_g \text{ con } k_g \geq \left\| \frac{\partial g(q)}{\partial q} \right\| \quad (24)$$

Para el caso del primer término de la función candidata [1]:

$$\frac{1}{2} \begin{bmatrix} \omega \\ \tilde{q} \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} \frac{1}{\alpha K_i} & 0 & 0 \\ 0 & \alpha K_i & -\alpha M(q) \\ 0 & -\alpha M(q) & M(q) \end{bmatrix} \begin{bmatrix} \omega \\ \tilde{q} \\ \dot{q} \end{bmatrix} \quad (25)$$

Si α es condicionada a cumplir la desigualdad denotada en la ecuación (26), entonces $V(\tilde{q}, \dot{q}, \omega)$ es definida positiva en forma global y radialmente no acotada [1].

$$\frac{\lambda_{\min}\{M(q)\}\lambda_{\min}\{K_v\}}{\lambda_{\max}^2\{M(q)\}} \quad (26)$$

La derivada temporal de la función candidata de Lyapunov se presenta en la ecuación (27), la cual cumple con ser semidefinida negativa y por lo tanto el punto de equilibrio del sistema es estable según el criterio de estabilidad de Lyapunov. [1, 3, 4]

$$\dot{V}(\tilde{q}, \dot{q}, \omega) = \dot{q}^T [K_v - \alpha M(q)] \dot{q} - \tilde{q}^T [\alpha K_p - K_i] \tilde{q} - \alpha \tilde{q}^T C(q, \dot{q})^T \dot{q} - \alpha \tilde{q}^T [g(q_d) - g(q)] \quad (27)$$

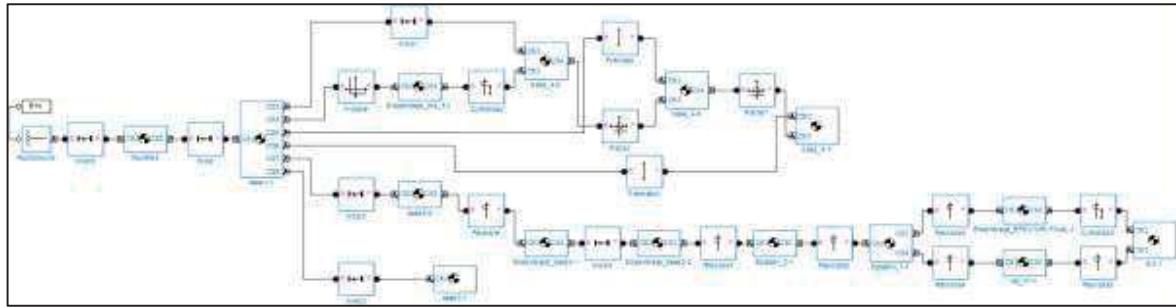


Fig. 2. Diagrama a bloques del robot antropomórfico

3.3 Sintonización de ganancias

Para el caso del control PD se propone la siguiente matriz que contiene las ganancias proporcionales para el sistema:

$$K_p = \begin{bmatrix} 6.8 & 0 & 0 & 0 \\ 0 & 4.5 & 0 & 0 \\ 0 & 0 & 2.7 & 0 \\ 0 & 0 & 0 & 8.5 \end{bmatrix} \quad (28)$$

Para hacer la sintonización y definir la matriz K_v se utiliza la ecuación (29)

$$K_v = 2\sqrt{K_p} \quad (29)$$

Entonces, se tiene:

$$K_v = \begin{bmatrix} 5.21 & 0 & 0 & 0 \\ 0 & 4.24 & 0 & 0 \\ 0 & 0 & 3.28 & 0 \\ 0 & 0 & 0 & 1.84 \end{bmatrix} \quad (30)$$

Las matrices anteriores son definidas positivas según el criterio de Sylvester.

Para facilitar la definición de las matrices K_p , K_v y K_i en el control PID, se debe considerar las ecuaciones (31) y (32). [1]

$$\lambda_{min}\{K_p\} > k_g \quad (31)$$

$$\frac{\lambda_{min}\{M(q)\}\lambda_{min}\{K_v\}}{\lambda_{Max}^2\{M(q)\}} > \alpha > \frac{\lambda_{Max}\{K_i\}}{\lambda_{min}\{K_p\}-k_g} \quad (32)$$

A partir de las ecuaciones anteriores se puede obtener un proceso de sintonización en términos de los valores propios de las matrices de ganancia, que se denota de la siguiente manera [1]:

$$\begin{aligned} \lambda_{Max}\{K_i\} &\geq \lambda_{min}\{K_i\} > 0 \\ \lambda_{Max}\{K_p\} &\geq \lambda_{min}\{K_p\} > k_g \\ \lambda_{Max}\{K_v\} &\geq \lambda_{min}\{K_v\} > \frac{\lambda_{Max}\{K_i\}}{\lambda_{min}\{K_p\}-k_g} \cdot \frac{\lambda_{Max}^2\{M(q)\}}{\lambda_{min}\{M(q)\}} \end{aligned} \quad (33)$$

Con lo cual, se proponen las siguientes matrices K_p , K_v y K_i , las cuales son definidas positivas según el criterio de Sylvester:

$$K_p = \begin{bmatrix} 4.9 & 0 & 0 & 0 \\ 0 & 3.7 & 0 & 0 \\ 0 & 0 & 2.4 & 0 \\ 0 & 0 & 0 & 1.6 \end{bmatrix} \quad (34)$$

$$K_v = \begin{bmatrix} 4.42 & 0 & 0 & 0 \\ 0 & 3.84 & 0 & 0 \\ 0 & 0 & 3.9 & 0 \\ 0 & 0 & 0 & 2.52 \end{bmatrix} \quad (35)$$

$$K_i = \begin{bmatrix} 1.3 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0.16 \end{bmatrix} \quad (36)$$

4. Simulador

Para implementar este simulador y proponer las dos leyes de control, además de su descripción matemática, es necesario conocer los parámetros físicos (longitudes, masas e inercia) de cada uno de los elementos del robot. Para lo cual se hace uso de un software de diseño y de herramientas de análisis mecánico.

4.2 Diseño CAD

El diseño del sistema mecánico es parte fundamental no sólo para el modelado matemático, sino también para el armado de la estructura física. El diseño del robot antropomórfico es realizado en SolidWorks con la finalidad de obtener las características físicas del sistema y facilitar la comunicación con MATLAB, la figura N° 1 muestra el diseño final.

El diagrama a bloques correspondiente al modelo tridimensional se muestra en la figura N° 2. Los parámetros físicos del robot, se obtienen a través de un análisis de elemento finito en SolidWorks, los datos más relevantes se presentan en la tabla N° 1. Para aplicar las estructuras de control considerando las matrices propuestas en la sección anterior, se

utiliza las herramientas proporcionadas por Simulink, siendo necesario obtener la posición, velocidad y aceleración de cada uno de los motores representados por un bloque rotacional, y a su vez aplicar el torque requerido para realizar el movimiento deseado.

Eslabón	Masa(kg)	Inercia($kg \cdot m^2$)	Posición C_m (cm)
Base	0,203155	$I_x = 2,376$ $I_y = 6,923$ $I_z = 6,241$	$x = -2,23$ $y = 1,13$ $z = -0,01$
Primer eslabón	0,02433	$I_x = 0,108$ $I_y = 0,383$ $I_z = 0,459$	$x = 0,19$ $y = -2,03$ $z = -0,01$
Segundo eslabón	0,04339	$I_x = 0,137$ $I_y = 0,890$ $I_z = 0,829$	$x = 0,72$ $y = 0,24$ $z = -0,03$
Efecto final	0,02698	$I_x = 0,169$ $I_y = 0,243$ $I_z = 0,123$	$x = 0,00$ $y = -1,34$ $z = -0,90$

Tabla 1. Parámetros físicos del robot

El simulador proporciona las gráficas relacionadas a la posición, al error de posición y al torque requerido para cada una de las uniones angulares, además de una visualización del robot virtual en tiempo real. Para mostrar su funcionamiento se proponen las siguientes posiciones angulares:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 90^\circ \\ 25^\circ \\ -45^\circ \\ 55 \end{bmatrix} \quad (37)$$

Obteniendo la posición final, mostrada en la figura N° 3.



Fig. 3. Posición final del robot

Matlab permite generar herramientas de visualización, modificar velocidad y generar un archivo tipo .AVI, entre otras cosas.

A continuación se presentan las gráficas del comportamiento del sistema obtenidas del simulador al aplicar la ley de control PD con compensación de gravedad, las figura N° 4, 5 y 6 muestran la posición, error de posición y torque de las tres primeras articulaciones del robot, respectivamente.

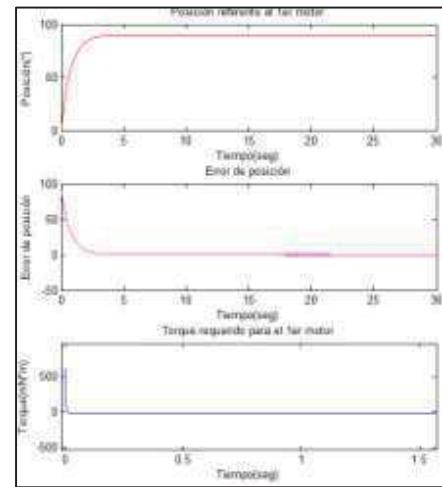


Fig. 4. Posición, error de posición y torque (base)

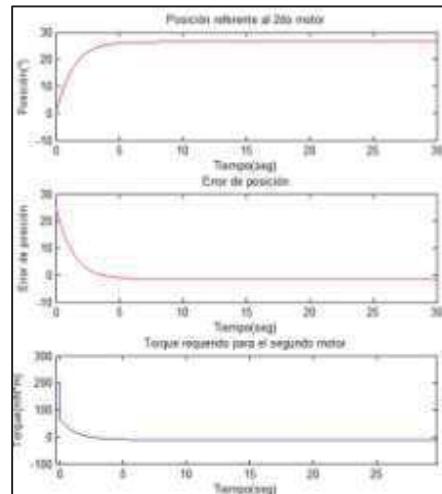


Fig. 5. Posición, error de posición y torque (eslabón 1)

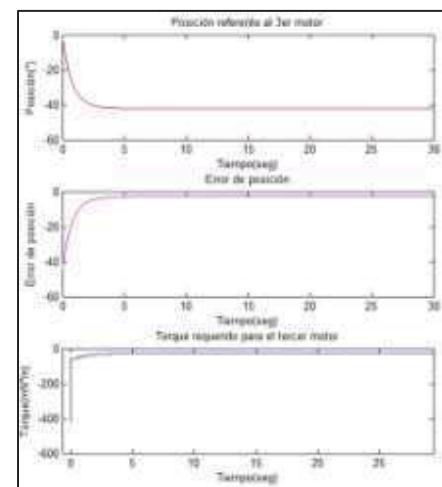


Fig. 6. Posición, error de posición y torque (eslabón 2)

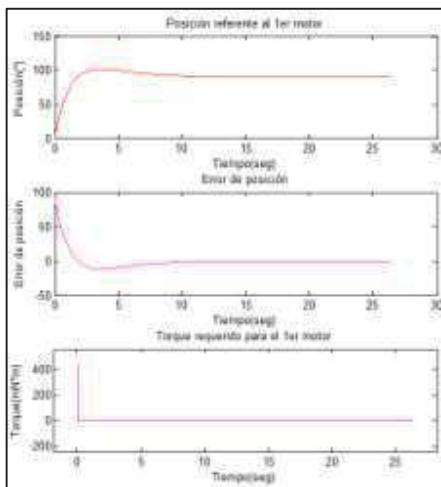


Fig. 7. Posición, error de posición y torque (base)

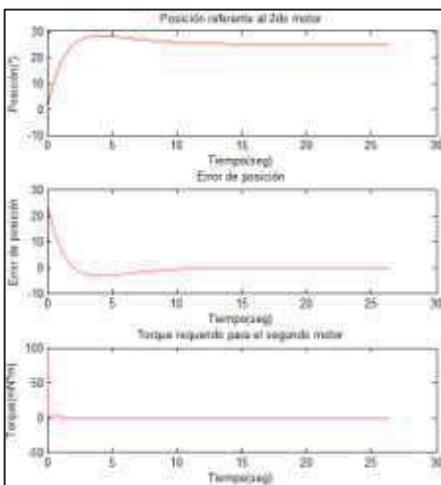


Fig. 8. Posición, error de posición y torque (eslabón 1)

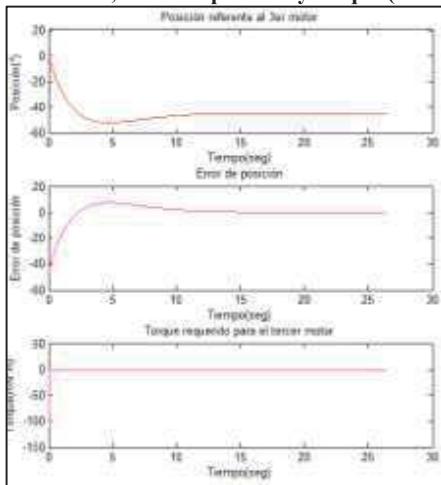


Fig. 9. Posición, error de posición y torque (eslabón 2)

Los resultados obtenidos de la posición, error de posición y torque con respecto a las tres articulaciones al aplicar el control PID se presentan en las figuras N° 7, 8 y 9.

5. Conclusiones

El uso de SolidWorks para la creación de elementos tridimensionales usados en Matlab para realizar la visualización gráfica del robot proporciona un ambiente realista, los datos necesarios para corroborar la similitud del modelo con la planta real, permitiendo modificar a conciencia cada una de las piezas y ensambles asegurando la exactitud del diseño. La comunicación que se establece con Matlab permite la manipulación de todos los parámetros físicos obtenidos en SolidWorks.

El simulador desarrollado entrega la información necesaria para el estudio del sistema permitiendo realizar el proceso de sintonización de ganancias ayudando a diferenciar de manera gráfica la respuesta ante una ecuación de control específica.

Considerando que adquirir un robot, modificarlo e implementar cualquier tipo de control requiere de tiempo y recursos económicos, se hace indispensable una simulación previa del sistema y del control que se desea implementar de forma física, ya que pese a que no es seguro la similitud exacta entre la respuesta del simulador y el sistema físico, si asegura que la respuesta real sea muy cercana a la esperada, esto gracias al análisis matemático que se requiere para diseñar un simulador.

Referencias

- [1] Kelly R. y Santibañez V. “Control de Movimiento de Robots Manipuladores”, Pearson-Prentice Hall, Madrid, 3era Ed. 2003
- [2] Barrientos A., Peñin F., Balaguer C. y Santoja R. “Fundamentos de Robótica”, McGraw Hill, México, 3era Ed. 2003
- [3] Khalil H. “Nonlinear Systems”, Prentice Hall, Estados Unidos, 3era Ed., 1996.
- [4] Spong M. y Vidyasagar M “Robot Dynamics and Control”, Jhon Wiley and Sons, Estados Unidos, 2da Ed., 1989
- [5] Slotine J. y Li W “Applied Nonlinear Control”, Prentice Hall, Estados Unidos, 3era Ed. ,1996

Implementación de un Sistema Neurodifuso en la Optimización del Modelo de Control para el Ambiente de una Incubadora Avícola

Conde Camacho Julián, Sánchez Cuapio Iván Jesús y Atecpanecatl Gutiérrez Jonathan

Departamento de Mecatrónica. Universidad Tecnológica de Tlaxcala. El Carmen Xalpatlahuaya, Huamantla Tlaxcala (ivan.sanchez@uttlaxcala.edu.mx, julian.conde@uttlaxcala.edu.mx, jonathan@uttlaxcala.edu.mx)

Resumen

El uso de sistemas inteligentes para control de ambiente en recintos cerrados ha logrado generar muchos modelos que resuelven el problema matemático resultante al modelar su comportamiento, permitiendo disminuir el margen de inestabilidad pero eso conlleva a que la tarea de control indique una implementación más compleja, pero con un desempeño mejorado.

El empleo de inteligencia artificial, para establecer un sistema de control a una incubadora avícola, permite realizar una comparativa entre modelos de control clásico y el sistema neurodifuso propuesto, basado en un modelo predictivo ANFIS, dotando al sistema de adaptabilidad con respecto a los parámetros internos de la incubadora y al cambiante clima que se genera en su exterior, logrando tener un sistema estable que administre a los componentes que regulan dichos parámetros y que garanticen el desarrollo del producto.

Con un modelo predictivo se indagan los distintos modos en que se mejora el desempeño de un proceso, siendo clave en los propósitos del desarrollo del prototipo; buscando reducir el consumo de energía requerido para lograr el desarrollo del producto de manera correcta y con un desempeño de mayor gama en comparación a las demás técnicas de control.

Palabras clave: Ambiente Controlado Inteligencia Artificial, ANFIS, Predictivo, Incubadora

1. Introducción

Existen actualmente diversas estructuras que permiten realizar el trabajo de una incubadora

avícola, bajo dos perspectivas, dispositivos que logran desarrollar el producto en grandes masas con un propósito comercial y a la par, modelos que establecen la misma tarea pero con un enfoque doméstico considerando cantidades menores. Para el primer caso se puede considerar dentro de las características, sistemas de control con una confiabilidad buena en el desarrollo de su trabajo pero un costo de adquisición relativamente alto. En el segundo caso se establecen modelos con confiabilidades muy bajas en la obtención del producto, de costos muy bajos, generalmente con falta de técnicas de control y el uso de los dispositivos indicados para la manipulación de los parámetros requeridos.

El establecer un prototipo doméstico que garantice un desarrollo del producto con gran índice de confiabilidad y que no genere costos adicionales, esclarece una meta en el diseño de la incubadora que sea aprovechada por cierta sección de la población. La confiabilidad en la obtención del producto, se procura cuando al sistema se le agrega inteligencia de adaptación a los parámetros externos, de tal modo que se refleje en la disminución de tiempo de trabajo de los dispositivos que regulan las variables de la incubadora y generen un bajo consumo de energía.

Los sistemas Neuro-Difusos son el resultado de la simbiosis de dos tecnologías aplicadas al control con grandes resultados y que han postrado el diseño de modelos de sistemas inteligentes mejorando el desempeño de procesos con cierto grado de imprecisión e incertidumbre, estos son los sistemas difusos y los sistemas neuronales [1].

Los modelos neurodifusos usan lógica que se comprende de reglas que combinan datos numéricos y lingüísticos, usan las condiciones del idioma natural, las cuales no son precisas y por lo tanto no

pueden representarse en la teoría normal. Los sistemas Difusos y las Redes Neuronales son estimados modelos numéricos y sistemas dinámicos, comparten la capacidad de mejorar la inteligencia del sistema trabajando en un ambiente un poco indeterminado y generando ejemplos de funciones para tomarlas como memorias asociadas.

Ambos tienen la ventaja de la estimación estadística y en la aproximación del control adaptable, generan una función sin necesidad de una descripción matemática donde la salida de la funcionalidad depende de la entrada, finalmente se puede decir, que aprenden de ejemplos o situaciones numéricas [2].

Esta característica de adaptabilidad permite estimar un rumbo distinto con el uso de esta tecnología, si se puede mejorar el desempeño de todo un sistema de control es posible que esta mejora se vea reflejada en los parámetros que permite la funcionalidad de los procesos.

2. Controladores Neurodifusos

Para lograr que un dispositivo adopte las características de control inteligente se deben marcar como indicio el contenido de las dos ramas que conforman al sistema propuesto.

2.1 Control difuso

La lógica difusa puede ser una herramienta útil al momento de hacer asociaciones del lenguaje natural con la inteligencia humana, de tal modo que algunos términos del comportamiento sean más descriptivos bajo el lenguaje cotidiano, a diferencia de las matemáticas puras; por ello su representación atribuye a construir modelos complejos bajo una perspectiva amplia y común [3].

La lógica difusa puede ser comprendida como un sistema matemático que considera funciones no lineales, basándose en los parámetros de entrada y obteniendo salidas específicas, retomando el concepto del aproximamiento. Las operaciones realizadas para lograr esta aproximación se desarrollan con el trabajo de los conjuntos difusos y su integración a partir de un sistema de inferencia, que logra cumplir su tarea con la estipulación de reglas de la forma SI-ENTONCES, convirtiendo este lenguaje en el reflejo de los sucesos predeterminados y su acción correspondiente.

2.2 Control Neuronal

Si hay que referir otro modo de dotar de inteligencia a un sistema, se debe considerar el funcionamiento del cerebro humano y la forma en que procesa la información. La principal funcionalidad se refiere al trabajo a partir de la idea de cajas negras, que no mantienen conocimiento previo de un hecho o situación pero que cuenta con datos de los cuales logra conjugar el concepto de aprendizaje [4].

Las neuronas son la conjunción de operaciones matemáticas que permiten dotar de inteligencia a los sistemas artificiales y dar la capacidad de recibir, procesar y transmitir señales de respuesta a diversos eventos. Para asemejar aún más esta técnica a la estructura natural, se debe mantener el enfoque de imitación teniendo una conjunción de varias de estas estructuras logrando una red neuronal, que debe contener las propiedades de redundancia y ser tolerantes a fallas o acciones no deseadas [4].

2.3 Sistemas de control neurodifuso

Los sistemas neurodifusos son un camino alterno para explotar las características que pueden encontrarse en la unión de las redes neuronales y la lógica difusa, aprovechando los méritos y eludiendo los defectos que cada una de las técnicas presentan por separado.

La estructura Neuro-Difusa permite el uso de métodos cualitativos y cuantitativos en la construcción de modelos, en la etapa de aprendizaje, de funcionamiento y de retroalimentación del conocimiento, permitiendo una apertura completa de su desempeño. Los métodos cuantitativos se observan en el aprendizaje que los sistemas neurodifusos pueden realizar a partir de datos del ambiente y en su forma matemática de funcionamiento, mientras que los métodos cualitativos se reflejan en el sentido lingüístico que tiene la base de conocimiento de los sistemas difusos, permitiendo la introducción de conocimiento previo al sistema y la retroalimentación del estado final del modelo, en forma de reglas expresadas lingüísticamente desde el sistema [5].

El concepto neto de neurodifuso se puede tratar a partir de las siguientes ideas, que son comunes a ambas tecnologías: La transparencia propia de los sistemas difusos que mejora la compresión de la red neuronal y la capacidad de aprender y adaptarse de las redes neuronales que aporta un mecanismo de sintonización automática, y adaptabilidad al sistema difuso.

Esta unión resulta especialmente interesante en aplicaciones de control, debido a que tanto las redes neuronales, como los sistemas de inferencia difusas tienen la capacidad de estimar el valor de la función que caracteriza su relación de entrada con la salida, sin necesidad de disponer de un modelo o descripción analítica de la misma. Su estimación es realizada partir de información adquirida por ejemplos proporcionados por el diseñador que esencialmente son numéricos. Ambos sistemas admiten un soporte matemático riguroso y una caracterización algorítmica que favorece el uso de herramientas de cálculo para su evaluación, y que resulta atractiva para su implementación mediante circuitos electrónicos, debido a su marcado carácter de procesamiento paralelo.

lingüística, determinar las reglas que constituyen la base de reglas, tomando en cuenta su número y enunciado en términos de antecedente y consecuente, seleccionar una máquina de inferencia, y finalmente seleccionar un método de defuzzificación.

Para el desarrollo de estos controladores se establece como prueba de desempeño a un sistema de simulación y las técnicas de aprendizaje de las redes neuronales son empleadas para encontrar y ajustar los parámetros de la estructura de estos sistemas y cumplir de manera óptima el objetivo de control.

La arquitectura establecida para este controlador se define a partir del diagrama a bloques mostrado en la figura N° 1.

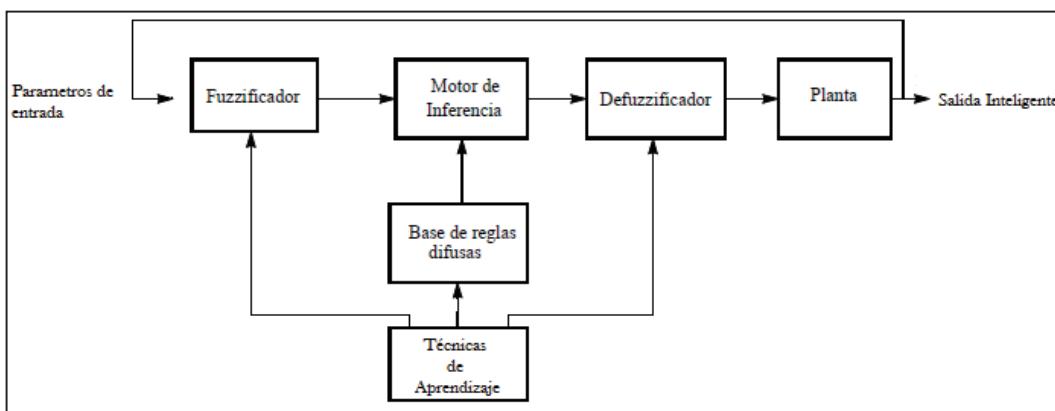


Fig. 1. Arquitectura de un controlador Neurodifuso.

3. Diseño del controlador

Los controladores neurodifusos se caracterizan por mantener los rasgos fundamentales y los componentes básicos de los sistemas de inferencia difusa, mientras que aprovechan la capacidad de aprendizaje de las redes neuronales para dotar al conjunto de la flexibilidad y la capacidad de adaptación necesaria que permite automatizar su proceso de diseño, o adaptar su comportamiento frente a cambios en el entorno [6].

Su diseño supone determinar su estructura, ajustar los parámetros que caracterizan a los elementos de esa estructura y en su caso, seleccionar una estrategia de aprendizaje.

La conformación del sistema consiste en resolver cuestiones como determinar el número de variables de entrada-salida y el número y distribución de sus etiquetas lingüísticas, determinar el tipo de las funciones de pertenencia asociadas a cada etiqueta

El esquema anterior muestra uno de los modelos probados para la tarea de control que permite los resultados deseados dentro el control del ambiente en un contenedor cerrado, de tal modo que mantiene la flexibilidad de adaptarse a diferentes técnicas de aprendizaje.

La tarea de aprendizaje y su estructura dentro de la arquitectura completa se puede integrar sobre distintos tipos de sistemas de inferencia difusa establecido la técnica de Takagi-Sugeno, que es representada por:

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{a_i} \right)^2 \right]^{\frac{1}{b_i}}} \quad (1)$$

Para una función de inferencia de gauss. La etapa de aprendizaje está establecida bajo la arquitectura ANFIS que presenta 5 capas en su estructura como se muestra en la figura N° 2.

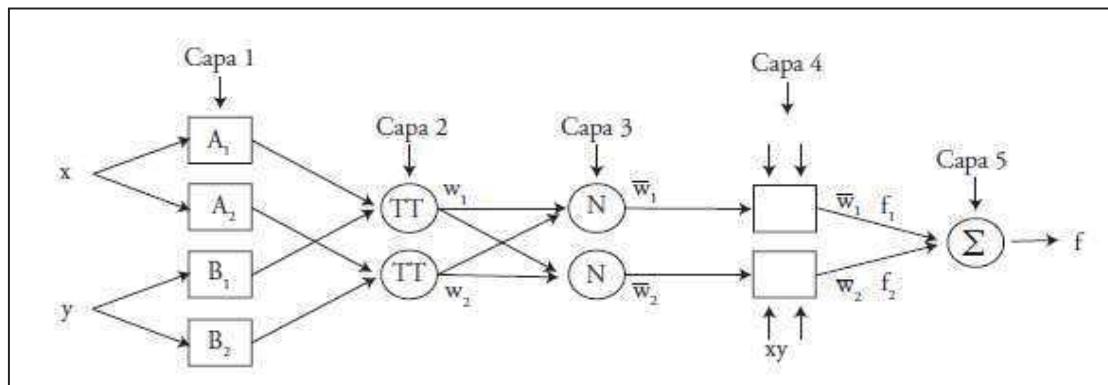


Fig. 2. Arquitectura ANFIS.

Donde las reglas de pertenencia son del tipo

$$\begin{aligned} \text{Si } x \text{ es } A_1 \text{ y } y \text{ es } B_1, \text{ entonces } f_1 = p_1 x + q_1 y + r_1 \\ \text{Si } x \text{ es } A_2 \text{ y } y \text{ es } B_2, \text{ entonces } f_2 = p_2 x + q_2 y + r_2 \end{aligned}$$

Esta estructura permite un aprendizaje híbrido con dos secciones generales, cuando los parámetros de consecuencia son identificados por el estimador de mínimos cuadrados y cuando los índices de error y los parámetros se actualizan por el método de gradiente descendente.

El desarrollo de la aplicación es llevado a cabo a través del paquete ICT(Intelligent Control Toolkit) de Labview, donde se genera el entrenamiento del sistema ANFIS y se analizan los resultados obtenidos

4. Análisis de desempeño

El sistema de control neurodifuso establecido es implementado en una incubadora que permite regular los parámetros de temperatura y humedad con una resistencia incandescente para la temperatura y una sumergida para la humedad, variando su ciclo de trabajo con modulación PWM. La adquisición de datos es realizada con una tarjeta ARDUINO UNO y dos sensores DHT11 que realizan el monitoreo de los mismos parámetros, adicionalmente para asistencia al control se desarrolla un circuito aerodinámico alimentado por un ventilador de CD que del mismo modo se regula su trabajo con modulación por ancho de pulso. Sean estos componentes los que establecen el número de entradas y salidas en el sistema MIMO con estructura ANFIS.

El trabajo con el razonamiento Neurodifuso logra su trabajo considerando el motor de inferencia mostrado en la figura N° 3.

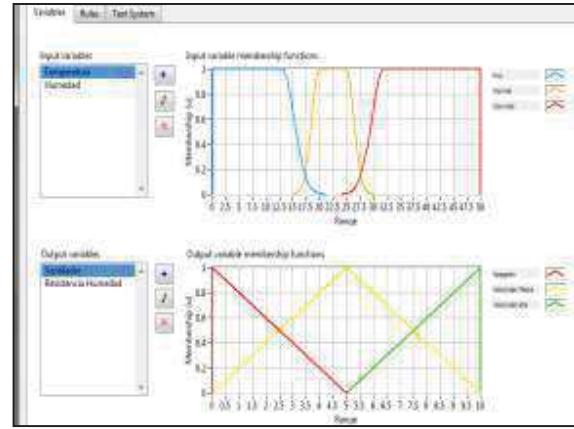


Fig. 3. Motor de inferencia implementado.

Y las gráficas del comportamiento de los actuadores que mantienen a los parámetros indicados se marcan en la figura N° 4.

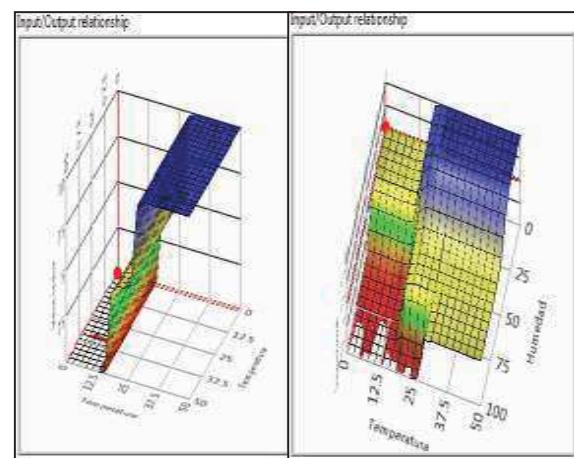


Fig. 4. Comportamiento del sistema para los actuadores del prototipo.

El avance en adaptabilidad del sistema se refleja en la figura N° 5.

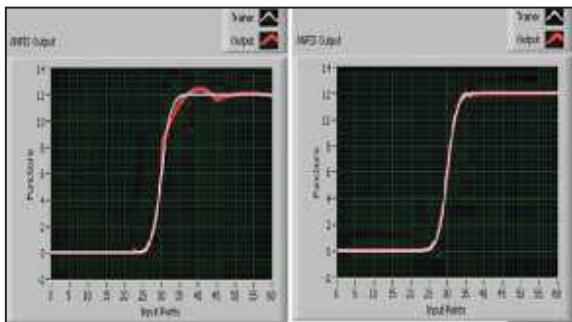


Fig. 5. Adaptabilidad del sistema implementado.

Para notar el desempeño del diseño implementado, se ha realizado una comparativa entre distintos sistemas de control de tal modo que se refleje la complejidad de implementación, el objetivo completado y finalmente bajo el tiempo de trabajo de los componentes involucrados de tal modo sea el reflejo del consumo de energía. Cabe mencionar que el tiempo de comparación es de una semana de trabajo y que el muestreo de los parámetros fue realizado cada tres minutos.

5. Resultados

La siguiente tabla enmarca de manera global los resultados obtenidos.

Técnica de control	Complejidad de implementación	Desempeño del control del ambiente	Tiempo de trabajo de los componentes(1 semana)	Consumo de energía
Proporcional	Bajo	Medio	95%	Alto
Proporcional integral	Media	Medio/Alto	90%	Alto
Difuso	Media/Alta	Alto	70%	Medio/Bajo
Neurodifuso	Alta	Alto	50%	Bajo

Como se puede ver conforme el avance en técnica del modelo establecido, el aumento de la complejidad fue exponencial; sin embargo en ese punto alto de diseño se logra una desviación menor del funcionamiento de los componentes para establecer los parámetros requeridos. Finalmente el apartado del modelo neurodifuso propuesto marca esa complejidad alta en su implementación pero el desempeño de los componentes fue mejorado en comparación a los modelos anteriores y el último apartado de consumo de energía, notablemente refleja este concepto.

El siguiente paso es alejar a la incubadora del sistema computacional con que se han realizado las pruebas preliminares para trasladar su trabajo a un microcontrolador de 8 bits y que en costo en muy

inferior, sin perder de vista que el diseño propuesto mantenga los beneficios logrados.

6. Conclusiones

En este trabajo se presenta un modelo de control neurodifuso basado en una estructura ANFIS para mantener en óptimas condiciones el ambiente en el interior de una incubadora para aves de corral.

Después de haber obtenido la caracterización de la incubadora bajo control clásico y habiendo realizado su implementación, se logró la obtener el producto que justifica el funcionamiento de sistema en general.

El tiempo de funcionamiento en que los componentes eléctricos estén activos refleja el consumo de energía eléctrica por ello que es de vital importancia que este funcionamiento sea el mínimo necesario pero cumpliendo con las condiciones necesarias para el desarrollo de las aves. Esto garantiza una reducción en el costo de producción para el prototipo, teniendo un impacto social sobre todo en poblaciones rurales en las cuales la crianza de estos animales favorece su alimentación e ingreso económico, logrando apoyar así, su sustento diario.

La adaptabilidad del sistema obtenido por el modelo neurodifuso, mantiene el logro en el consumo de energía reduciendo su tiempo de trabajo. La técnica utilizada ha sido probada en un ordenador, teniendo como parámetro solo el tiempo de ejecución. El paso siguiente es la migración de esta técnica a un microcontrolador que mantenga la trayectoria trazada de mantener el bajo costo de producción pero sin perder de vista su capacidad de soporte para estas tareas complejas. Del mismo modo se pueden efectuar mejoras del trabajo ideando nuevas técnicas de aprendizaje en la neurona establecida.

Referencias

- [1] Abonyi, Janos; Babuska, Robert y otros. Fuzzy modeling and model based control whith use of a priori knowledge. 1999
- [2] Abonyi, Janos and Nagy, Lajos. Adaptive Fuzzy Inference System and its Application in Modelling and Model Based Control, 1999.
- [3] Babuska, R.; Verbruggen, H.B. and Hellendoorn,H. Promising Fuzzy Modeling and Control Methodologies for Industrial Applications, 1998.
- [4] Johansen,T. A. ; Shorten, R. and MurraySmith, R. On the Interpretation and Identification of Dynamic TakagiSugeno Fuzzy Models, 1999.

Ingeniería Mecatrónica. Proyectos y Aplicaciones. Capítulo 29, pp.158-163, Octubre 2014.
® Asociación Mexicana de Mecatrónica A.C
ISBN: 978-607-9394-00-4

- [5] Kosko, Bart. Neural Networks and Fuzzy Systems. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [6] Jang, Jyh-Shing Roger and Sun, Chuen-Tsai. NeuroFuzzy Modeling and Control. The Proceeding of IEEE, 83(3): 378-406, 1995.

Instrumentación de Señales de Frecuencia Cardíaca para un Sistema Ambulatorio en Dispositivo

Sánchez Medel Luis Humberto¹, Juan Manuel González Sobal², Olivia Labastida Puertos³ y Solís Jiménez Miguel Ángel⁴

Instituto Tecnológico Superior de Huatusco
^{1,2,3} Departamento de Ingeniería en Sistemas Computacionales, ⁴Departamento de Ingeniería Industrial

Resumen

Las enfermedades cardiovasculares han llegado a ocupar el segundo lugar en mortandad en México, solo superado por la diabetes de acuerdo al censo de INEGI en el 2009. Se denomina Holter a una prueba diagnóstica que consiste en la monitorización ambulatoria del registro electrocardiográfico por un tiempo prolongado, habitualmente unas veinticuatro horas, en una persona que está en movimiento. Los dispositivos que actualmente son usados para medir la frecuencia cardíaca del corazón son: electrocardiógrafo, el estetoscopio, pulsímetro, aparatos digitales como muñequeras de pulso que miden frecuencia cardíaca y tensión arterial que el sector deportista emplea.

La evolución de los dispositivos móviles busca la integración de los mismos a nuestras actividades diarias, muchas veces las personas usan aparatos electrónicos tales como tabletas y teléfonos inteligentes para poder ver su estatus en las redes sociales, entretenimiento o gestionar su tiempo, sin embargo un campo que no ha sido explotado como los otros es el de la salud.

El objetivo de esta investigación es adquirir vía bluetooth, registrar y analizar los latidos del corazón por minuto en estado de reposo empleando un dispositivo móvil para realizar un análisis de FCR y dar un mensaje auditivo sobre su estado.

Palabras clave: Android, móvil, corazón, sensor, FCR, ambulatorio

1. Introducción

México es un país que ha llegado a ocupar el segundo lugar en mortandad a causa de enfermedades cardiovasculares, solo por encima de la diabetes y en varios países desarrollados ha llegado al primer lugar

[1]. Se denomina Holter a una prueba diagnóstica que consiste en la monitorización ambulatoria del registro electrocardiográfico por un tiempo prolongado, habitualmente unas veinticuatro horas, en una persona que está en movimiento.

Los dispositivos que actualmente son usados para medir la frecuencia cardíaca del corazón son: electrocardiógrafo, el estetoscopio, pulsímetro, aparatos digitales como muñequeras de pulso que miden frecuencia cardíaca y tensión arterial. Instrumentación de Señales de Frecuencia Cardíaca para un Sistema Ambulatorio en Dispositivo Móvil Estos dispositivos ambulatorios que el sector deportivo utiliza para obtener datos y mejorar su rendimiento del practicante, estos dispositivos funcionan con un software embebido en un microcontrolador, dicho principio puede ser aplicado a otros usuarios lo cuales empleando tecnología móvil a su alcance puedan monitorear su estado cardiaco mediante pruebas FCR que se basan en la relación de las pulsaciones por minuto, peso, edad y sexo.

La evolución de los dispositivos móviles busca la integración de los mismos a nuestras actividades diarias, y se pretende que mediante el auge hoy en día de equipos electrónicos como los son: Tabletas, Teléfonos inteligentes, Agendas electrónicas, etcétera, podamos crear una aplicación en un dispositivo que concientice a la población sobre su salud y la gran problemática del país en cuanto a las enfermedades cardiovasculares que tanto afectan.

La Fig. 1 fue tomada del último censo de población en México en el 2007 muestra el nivel de mortandad en México debido en relación a las enfermedades que afectan al corazón.



Fig. 1 Principales afecciones de las enfermedades cardiovasculares en 2007, fuente INEGI

El objetivo de esta investigación es registrar y analizar los latidos del corazón por minuto en estado de reposo empleando un dispositivo móvil para realizar un análisis de FCR y dar un mensaje auditivo sobre su estado. La interfaz diseñada para el dispositivo móvil se muestra en la Fig. 2.



Fig. 2 Interface del sistema en una Tablet con Android 4.2

2. Arquitectura del sistema

Cuando las células cardíacas son estimuladas con un impulso eléctrico, ya sea de manera espontánea o de manera externa, la polaridad de la célula cambia, esto se conoce como potencial de acción. El potencial de acción tiene 4 distintas fases, en el potencial de acción de las células automáticas hay una fase 0 de ascenso más lento respecto a la de las no automáticas o de repuesta rápida, y sus fases 1,2 y 3 no pueden diferenciarse de una manera clara, solo se toman como si fueran una fase 3. En la fase 4, en el potencial de acción de las células automáticas hay una despolarización espontánea, lo cual genera que las células automáticas alcancen el potencial umbral para comenzar de nuevo el potencial de acción. Esto se puede ver de manera gráfica en la figura 3.



Fig. 3 Descripción de la forma de onda

Las pulsaciones por minuto o BPM son obtenidas a partir de la forma de onda de las pulsaciones eléctricas del corazón. Las fórmulas para la obtención de estas varían dependiendo la edad y el sexo de la persona [2]. El Tabla 1 muestra un extracto de los BPM ideales en una persona saludable dependiendo su edad, sexo y hábitos.

Tabla 1. Extracto de tabla de BPM ideales con relación a hábitos

HOMBRES SEDENTARIOS		20	a	25	25	a	30		
Pulsaciones máximas teóricas		100,00		200	a	195	195	a	190
Ejercicio anaeróbico máximo		90,00		180	a	175,5	175,5	a	171
Ejercicio anaeróbico		80,00		160	a	156	156	a	152
Ejercicio aeróbico		70,00		140	a	136,5	136,5	a	133
Aeróbico adelgazar		60,00		120	a	117	117	a	114
Aeróbico iniciación		50,00		100	a	97,5	97,5	a	95

MUJERES SEDENTARIAS		20	a	25	25	a	30		
Pulsaciones máximas teóricas		100,00		206	a	201	201	a	196
Ejercicio anaeróbico máximo		90,00		185,4	a	180,9	180,9	a	176,4
Ejercicio anaeróbico		80,00		164,8	a	160,8	160,8	a	156,8
Ejercicio aeróbico		70,00		144,2	a	140,7	140,7	a	137,2
Aeróbico adelgazar		60,00		123,6	a	120,6	120,6	a	117,6
Aeróbico iniciación		50,00		103	a	100,5	100,5	a	96

El FCR sirve como indicador para determinar el estado de salud de una persona, se basa en el conteo de BPM [3], la tabla parcial de relación se muestra en el Tabla 2.

Tabla 2. Extracto de tabla de BPM ideales en FCR

HOMBRES	Mala	Normal	Buena	Muy Buena
20-29	86 o más	70-84	62-68	60 o menos
30-39	86 o más	72-84	64-70	62 o menos
40-49	90 o más	74-88	66-72	64 o menos
50-59	90 o más	74-88	68-74	66 o menos
60 o más	94 o más	76-90	70-76	68 o menos
MUJERES	Mala	Normal	Buena	Muy Buena
20-29	96 o más	78-94	72-76	70 o menos
30-39	98 o más	80-96	72-78	70 o menos
40-49	100 o más	80-98	74-78	72 o menos
50-59	104 o más	84-102	76-82	74 o menos
60 o más	108 o más	88-106	78-88	78 o menos

El sistema emplea el BPM y FCR para realizar un análisis de basado en las pulsaciones, de esta forma obtener la condición del corazón. El sistema cuenta la arquitectura mostrada en la Fig. 4, la cual consiste en un enlace inalámbrico mediante un dispositivo Bluetooth entre la tableta/celular y el sensor del corazón [4,5,6]

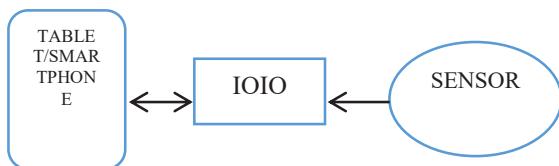


Fig. 4 Arquitectura implementada para el procesamiento digital de la señal

La frecuencia de muestreo es delimitada por el tiempo que toma al ADC y a la señal inalámbrica en ser registrada por el software, dicho tiempo es de 1ms, por lo que basados en el teorema de Nyquist se determina que la frecuencia muestreada debe de ser por lo menos 4 veces mayor que la frecuencia a muestrear. La frecuencia muestreada es tomada de la

tabla 1, cuyo máximo valor es 206, sustituyendo en la ecuación 1.

$$f_m = \frac{4 * BPM_{max}}{60} = 13.73\text{Hz} \quad (1)$$

La frecuencia de muestreo máxima que permite el sistema es determinado por el periodo de conversión y el teorema de Nyquist, lo cual se calcula en la ecuación 2.

$$f_{m,max} = \frac{1}{T * 4} = 250\text{hz} \quad (2)$$

La resolución requerida para reconstruir la señal fue hecha a 10 bits, por lo que el voltaje de paso es determinado por la ecuación 3, dicha resolución fue delimitada la amplitud de la señal proveniente del sensor, la cual varia de 0-3.3 volts.

$$Vp = \frac{V_{cc}}{2^{10}} = 3mV/paso \quad (3)$$

La determinación de los BPM se obtiene al aplicar la siguiente fórmula basada en el ángulo de la forma de onda obtenida por el sensor de forma inalámbrica por bluetooth.

$$B = [y(t_n) - y(t_{n+20})] \quad (4)$$

si: $B > 30$, entonces FASE 0, de lo contrario FASE 3

La fórmula propuesta indica que la respuesta del sensor es candidato a ser un latido de corazón si el ángulo que muestra entre cada 20 tomas supera los 30 grados. Obteniendo la cantidad de latidos verdaderos en 14 segundos es posible hacer una proporción a 1 minuto.

3. Análisis de resultados

El dispositivo creado se muestra en la figura 5, se encuentra ubicado a la altura de la cintura y el sensor se halla conectado en la oreja del usuario.

Se realizaron pruebas con otras aplicaciones para Android y se obtuvieron resultados muy similares a estas, las aplicaciones probadas fueron: Instant Heart Rate, Heart Beat Rate Pro y Heart Beat. Se relacionó el dispositivo con un electrocardiógrafo y se obtuvo el siguiente resultado mostrado en la figura 6.

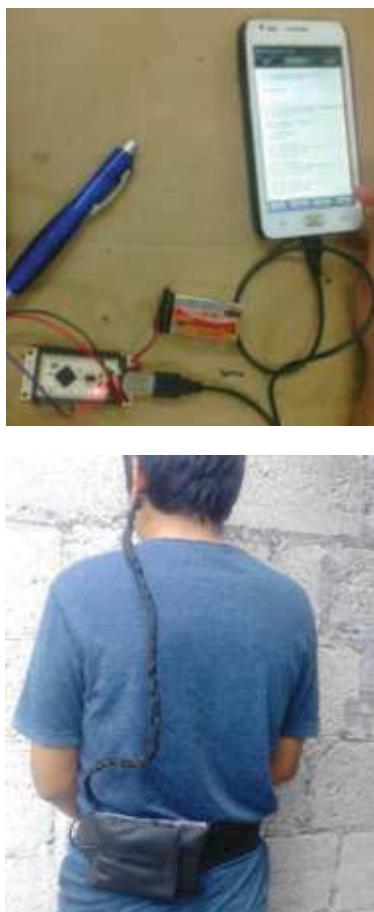


Fig. 5 Dispositivo construido y operando

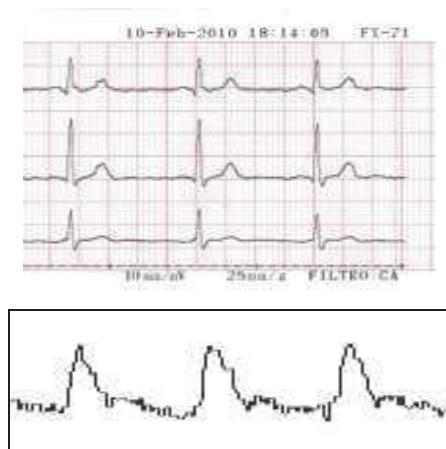


Fig.6. Comparación entre una señal de un cardiógrafo comercial y el propuesto

4. Conclusiones

La comparación mostrada en la fig. 6 resalta que la resolución es inferior con uno comercial, sin embargo el dispositivo no fue diseñado para realizar

un análisis exhaustivo de la señal, sino para detectar los latidos por minuto, por lo cual este tipo de resolución es aceptable para su uso en el algoritmo de detección por diferenciación en el tiempo.

El instrumento visualizador/registrador e indicador permite adquirir la señal, grabar y analizar la señal para indicarnos un análisis en base a los latidos por minutos del corazón, que pese a que no se tiene una resolución como los electrocardiógrafos comerciales, si nos permite el conteo de latidos por minuto que es la base de nuestro análisis y también el cómo los doctores generales obtienen información de nuestros cuerpos a través de un estetoscopio. El dispositivo diseñado no tiene la ambición de ser una sustitución de un médico, sino para concientizar a las familias de recursos medios sobre su propia salud empleando dispositivos móviles como una herramienta más en su integración.

Como trabajo a futuro se requiere un filtro digital para eliminar el ruido proveniente del sensor, así como un sistema de detección de estabilidad de la señal en la tableta/celular, sin embargo si es posible determinar los BMP.

Referencias

- [1] Página Oficial del Instituto Mexicanos del Seguro Social. *Preguntas frecuentes sobre enfermedades del corazón. IMSS (2013)*. http://www.imss.gob.mx/preguntas/salud/Pages/fa_qscardio.html
- [2] Angeles Alfredo, “*Conocimientos Básicos del Corazón*”. Distribuido por Novartis, 2004.
- [3] Bayes Antoni, “*Electrocardiografía Clínica*”. Ed. Espaxs S.A. Publicaciones, Médicas, 2004.
- [4] Tomas Girones, J., “*El gran libro de Android*”, editorial AlfaOmega, grupo editor México, junio 2011, ISBN: 978-607-707-226-3.
- [5] Organización Editorial Mexicana, Artículo Hasta 20% de la población padece trastorno del ritmo cardiaco: IMSS, 24 de noviembre del 2009. <http://www.oem.com.mx/laprensa/notas/n1415996.html>
- [6] Monk S., “*Making Android Accessories with IOIO*”, Going mobile with sensors, lights, motors, and robots, editorial O’reilly, primera edición febrero 2012 ISBN: 978-1-449-32328-8.

Síntesis de un Sistema de Monitoreo del Consumo Eléctrico Implementado en un Dispositivo Móvil

Sánchez Medel Luis Humberto¹, Solís Jiménez Miguel Ángel² y Ramírez Rebolledo Miguel Ángel³

Instituto Tecnológico Superior de Huatusco

¹Departamento de Ingeniería en Sistemas Computacionales, ²Departamento de Ingeniería Industrial,

³Departamento de Ingeniería electromecánica

Resumen

El consumo eléctrico es la cantidad de potencia eléctrica consumida en un intervalo de tiempo, es uno de los indicadores empleados que conjunto a la región, época del año definen el consumo mensual promedio. Una de las características principales de un consumidor doméstico promedio es la ignorancia de sus hábitos de consumo eléctrico, por lo que un plan de ahorro de energía que este tome debe de tomar en cuenta un sistema de medición para medir su eficacia.

El uso de las tecnologías móviles busca la integración de las mismas en nuestras actividades diarias; cada vez las tabletas y teléfonos inteligentes son más poderosos permitiéndonos correr algoritmos complejos con nuevas formas de interacción a través de pantallas táctiles y diversos sensores. Las computadoras, celulares y otros dispositivos electrónicos incluyen opciones de energía cuyo principio puede ser aplicado a una casa habitación para monitorear el consumo eléctrico pero sin altear en gran medida la estructura física o eléctrica de la vivienda.

El objetivo es determinar el consumo eléctrico empleando un sistema basado en un dispositivo móvil empleando sensores de corriente y comunicación Bluetooth, mediante el cual se establece un patrón de hábitos energéticos.

Palabras clave: Android, consumo eléctrico, dispositivo móvil, interfase

1. Introducción

La medición de la potencia eléctrica absorbida permite analizar hábitos de consumo eléctrico y con ello crear un plan de ahorro de energía que auxilie a analizar el rendimiento de cada uno, sin embargo el análisis del consumo eléctrico es realizado cada vez que se presenta la cuenta por dicho servicio, el cual varía dependiendo el lugar y temporada del año, por lo que la efectividad de un plan de ahorro de energía puede variar.

La presente investigación se basa en un sistema de monitoreo del consumo eléctrico implementado en un dispositivo móvil cuyo objetivo es mantener informado en tiempo real el estado del consumo y con ello auxiliar en la elección de un plan ahorrado de energía con la posibilidad de recibir alarmas por exceso de consumo, fallos o por variables eléctricas que salgan de normas o rangos preestablecidos [1]; esto nos da el beneficio de poder realizar un plan de ahorro energético efectivo para poder actuar anticipadamente y prever paros de equipos e instalaciones, minimizando tiempos muertos y evitando pérdidas, es decir, es la solución exacta para lograr un consumo responsable y un uso razonable de la energía eléctrica, con indicadores medibles y con los que podamos tomar acciones puntuales de respaldo energético [2].

El instituto de Investigaciones Eléctricas previó en 1994 que el sector eléctrico en el mundo y en sus próximos treinta años tendría un crecimiento importante de 6% a 9% en sus próximos años, mientras que en los países del primer mundo será de sólo 2% y 3% los resultados de sus predicciones se muestran en la Fig. 1. En cuanto a las causas del crecimiento, es lógico que el nivel de equipamiento crezca más rápidamente en los países en vías de

desarrollo que en los industrializados. Sin embargo, los recursos financieros que se requieren para mantener el crecimiento requerido no existen actualmente ni en los países ni en los fondos internacionales.

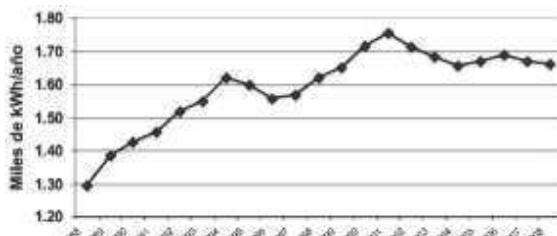


Fig.1 Evolución del consumo eléctrico promedio en el sector residencial de México 1988-2008

Uno de los mayores obstáculos a la hora de ahorrar energía eléctrica en casa es que el consumo eléctrico es invisible a simple vista, de hecho casi nadie sabría calcular en un momento dado cuánta electricidad está consumiendo. Ni si está consumiendo más de lo imprescindible o qué medidas puede tomar para reducir ese consumo que no suponga vivir a oscuras. Por ejemplo, es bastante habitual dejar algo pequeño enchufado porque se presume que consume poco o que está apagado o «en reposo», y en cambio limitar el uso de algún aparato eléctrico más grande, como un aire acondicionado, al que se le presupone un gran consumo [3].

a Fig. 2 muestra la forma en que un sistema basado en SCADA se emplea para medir la energía eléctrica, sin embargo se requiere la modificación de la estructura física del cableado eléctrico y su adaptación a una casa habitación ya construida es más cuantiosa.

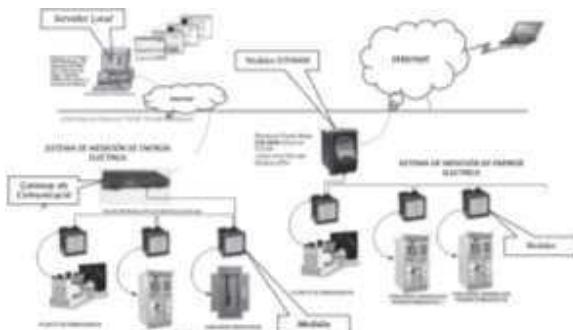


Fig. 2 Ejemplo de sistema de medición de la energía eléctrica

La investigación realizada en este artículo tiene por objetivo el diseño de un sistema indicador de pase de consumo entre tarifa de esta forma concientizar al usuario de la importancia de llevar un

control y monitoreo continuo de la energía eléctrica que consume, para ello se hace uso del diagrama de la Fig. 3.

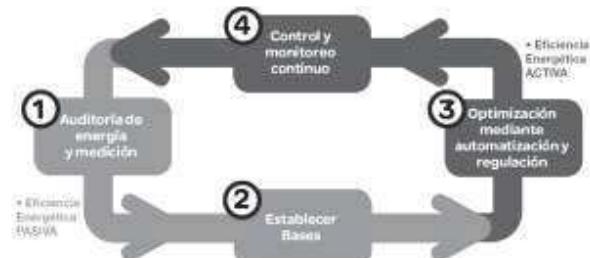


Fig. 3 Ciclo de vida de eficiencia energética

El modelo del ciclo de eficiencia energética propuesto por la empresa Schneider se basa en: todas las mejoras comienzan con una decisión acerca de lo que hay que implementar. Una auditoría de energía es un primer paso apropiado para recopilar información sobre las oportunidades de ahorro y sacar el máximo provecho a las medidas que ya están disponibles, con esta información, se podrán tomar decisiones bien fundamentadas acerca de los siguientes pasos a dar.

Las primeras acciones identificadas serán medidas pasivas para establecer las bases. La sustitución de dispositivos poco eficientes por dispositivos de bajo consumo, la reparación de las fugas y la colocación de aislamientos suelen ser medidas de bajo costo que resultan rápidamente rentables.

Optimización de los sistemas. La automatización de estos sistemas garantiza el cumplimiento de medidas de sentido común como el apagado de las luces durante la noche.

Realizar mediciones y controles adicionales. Si disponemos de datos energéticos más detallados, es probable que podamos ahorrar más energía. Esto nos lleva a comenzar de nuevo el ciclo.

El sistema diseñado se realizó con el objetivo de detectar el cambio de consumo por parte del usuario, sin que el sistema implementado requiriese un medio exhaustivamente invasivo a la instalación eléctrica del hogar; dichos requerimientos cubren los puntos descritos por Schneider y son exemplificados en la Fig. 4.

2. Arquitectura del sistema

La arquitectura del sistema de monitores inalámbricos se describe en la Fig. 4, el cual consiste de:

Sensor de corriente, que detecta el consumo eléctrico y su salida es analógica. La cual es conectada a un Arduino y IOIO.

Sensor de foto resistivo, detecta el índice de luz y verifica que el foco (carga eléctrica) realmente haya encendido.

Transceptor Bluetooth, envía y recibe la información de forma inalámbrica al módulo de control de interface.

Módulo de control, sistema de interface entre el celular/tableta y los módulos que encienden, registran el índice de consumo eléctrico.

Un monitor (pantalla de Tablet/celular) que sirve parella visualizar instantáneamente nuestro consumo.

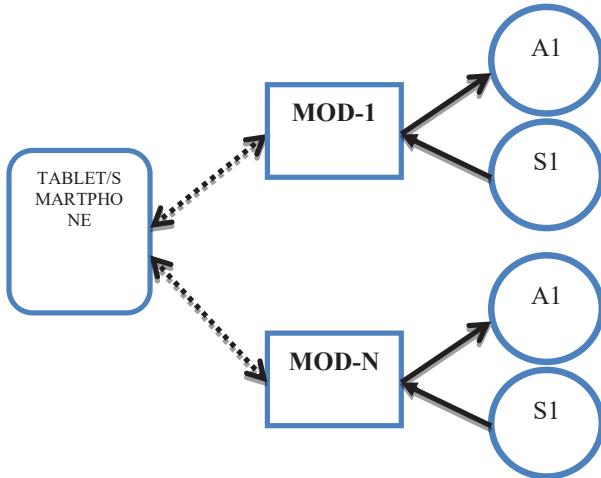


Fig. 4 Arquitectura implementada para el sistema de medición de energía eléctrica

El principio para la medición del consumo eléctrico se basa en la potencia eléctrica-Hr, el cual está dado en KW-Hr, la respuesta del sensor es en forma de voltaje análogo.

La resolución del ADC necesaria para el dispositivo es 10 bits, esto es debido al span de 50 A que el sensor es capaz de medir y la alimentación de 5 volts hacen que el voltaje de paso sea:

$$V_{paso} = \frac{5v}{2^{10}} = 4.8mV$$

La señal de salida del sensor es de 0.8 a 5volts, por lo que tiene un span de 4.2 volts. Por lo que el rango del ADC será de 16 (0.8 volts) a 1023 (5 vlt). Cada incremento en amperaje convertido a decimal por el ADC corresponde a:

$$A_{paso} = \frac{50A}{1006} = 0.49mA$$

La tarjeta Arduino cuenta con un microcontrolador AVR el cual tiene una resolución nativa de 10 bits. La frecuencia de actualización de los datos es importante debido a que el dato tabular de consumo eléctrico usa como variable la corriente máxima alcanzada. El tiempo de conversión es de 1ms del Arduino, sin embargo el algoritmo empleado para el cálculo de la potencia absorbida toma 2.3 milisegundos, esto fue determinado usando mediante el diagrama de flujo en la Fig. 5.

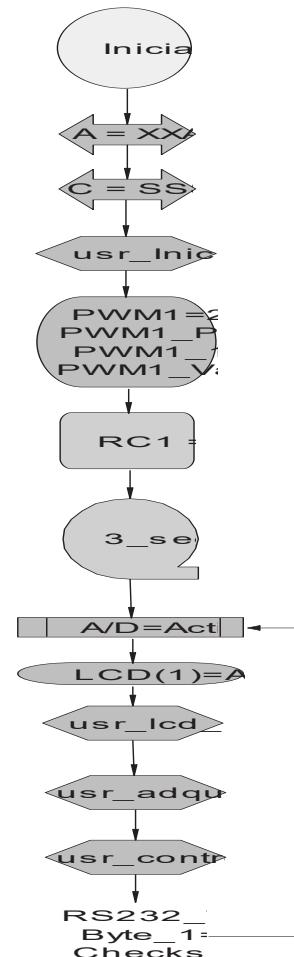


Fig. 5 Diagrama de flujo del firmware

El enlace entre los módulos y el sistema de control está mantenido por una comunicación Bluetooth continua [4,5]. Un ejemplo de los módulos implementados se muestra en la Fig. 6. Los elementos que comprende el módulo son 1. Sensor de corriente, 2. Actuadores, 3. IOIO con módulo Bluetooth y 4. Microcontrolador.

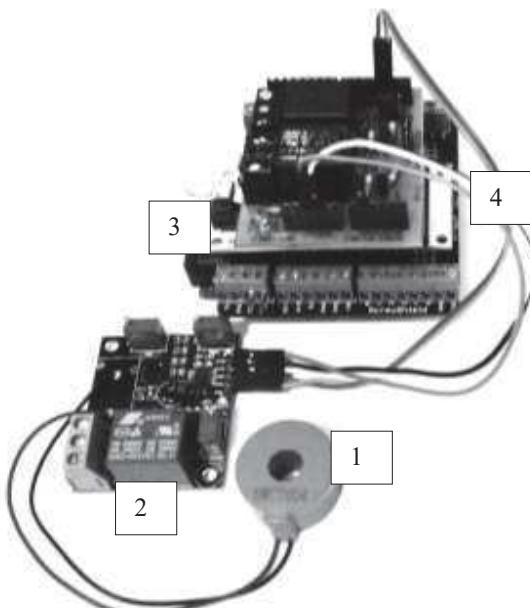


Fig. 6 Modulo inalámbrico con sensor de corriente y actuadores

El registro del consumo eléctrico es realizado de forma aditiva, por ello la gráfica del consumo no disminuye, lo cual es análogo a los medidores comerciales. El cálculo del consumo eléctrico se basa en la corriente consumida medida en espacios de 10 milisegundos y el conocimiento que la línea es de 120 VCA, por lo cual mediante ley de ohm se puede obtener la potencia eléctrica.



Fig. 7 Interface a usuario del control de consumo

El diseño de la interfaz se basa en el diseño minimalista de objetos visuales, los cuales son botones que generan cambios en la salida de los módulos y deja el control del suministro a una tarjeta

microcontroladora que contiene software embebido. La interfaz cuenta con los elementos de activación, generación de reporte semanal y una barra que se incrementa de forma bimestral la cual nos indica el consumo actual y al llenarse su paso al próximo consumo. El diseño de la interfaz se muestra en la Fig. 7.

3. Análisis de resultados

Se comparó el resultado de la medición del sistema diseñado y un medidor de potencia eléctrica comercial análogo que se emplea en instalaciones de casa hogar como indica la Tabla 1. Se emplearon 2 cargas eléctricas que consumen una cantidad de corriente lo suficientemente amplia para que se muestre en un medidor comercial como una plancha a su máxima potencia y un ventilador a su máxima velocidad, ambas cargas eléctricas funcionaron por un espacio de 10 minutos con repeticiones de 15 y 20 minutos para asegurar la repetición en las mediciones.

Tabla 1. Resultados de la medición del consumo eléctrico

	Calculada	App	Medidor comercial
Carga 8 A (Plancha)	960 W	957.8 W	962 W
Carga 5 A (Ventilador)	600W	598.5 W	604 W

Se observó que el dispositivo sensor tiene una medida similar y un error del 0.7%, sin embargo la prueba no puede ser concluyente debido a que todo instrumento es sometido a una calibración por lo que se emplearon 3 medidores comerciales distintos obteniendo una similitud en los resultados con variaciones de error pequeñas.

4. Conclusiones

Muchas personas nos sorprendemos y quejamos cuando tenemos que pagar grandes cantidades de dinero por la electricidad que usamos en nuestras casas y pensamos que nuestro consumo es menor al que marca el recibo. Medir y conocer el dato de consumo eléctrico es el primer paso para poder reducirlo y ahorrar energía y dinero.

El acceso a información de costos de energía en forma diaria es un factor de decisión importante y un apoyo decisivo para implementar proyectos de eficiencia energética.

Cada kilowatt-hora de electricidad no generado se deja de emitir a la atmósfera:

- 680.4 gramos de Bióxido de Carbono
- 5.8 gramos de Bioxido de Azufre
- 2.5 gramos de Oxidos Nitrosos

El uso del sistema de monitoreo presentado puede mejorar la eficiencia, fiabilidad y flexibilidad, reducir el costo del suministro de electricidad, y reducir los impactos ambientales.

El ahorro energético y económico no vendrá dado por sí mismo por unos dispositivos con una serie de mejores funcionalidades, sino por como aprovechen los usuarios sus ventajas. Por lo tanto, este despliegue tecnológico debe ir acompañado de políticas y planes que fomenten e impulsen el aprovechamiento efectivo de estos sistemas.

“Lo que no se define no se puede medir. Lo que no se mide, no se puede mejorar. Lo que no se mejora, se degrada siempre.” (Lord Kelvin).

Referencias

- [1] Ramos Niembro, Variables que influyen en el consumo de energía eléctrica, Boletín ENERO-FEBRERO 1999, Instituto de investigaciones eléctricas, México 1994. <http://www.iie.org.mx/medioamb99/apli.pdf>
- [2] Díaz y Mansera. 2002. “*Uso de en México: Situación actual, retos y oportunidades*” en SENER, Balance Nacional de Energía, Secretaría de Energía, México, D..D. p.99-109
- [3] Sánchez, L 2012 “*El consumo energético de los hogares en México*”, Coyuntura Demográfica, No. 2, Sociedad Mexicana de Demografía, México
- [4] Tomas Girones, J., “*El gran libro de Android*”, editorial AlfaOmega, grupo editor México, junio 2011, ISBN: 978-607-707-226-3.
- [5] Monk S., “*Making Android Accessories with IOIO*”, Going mobile with sensors, lights, motors, and robots, editorial O’reilly, primera edición febrero 2012 ISBN: 978-1-449-32328-8.

Simulación del Modelo VegSyst para Determinar el Crecimiento de la Planta de Jitomate

Mancilla Morales J. Arturo¹, Tornero Campante Mario¹, López Cruz Irineo², Ríos Silva Eduardo³ y Regalado López José¹

¹ Colegio de Postgraduados, Campus Puebla. Postgrado en Estrategias para el Desarrollo Agrícola Regional.

² Universidad Autónoma de Chapingo, Postgrado en Ingeniería para el uso Integral del Agua.

³ Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica.

Resumen

*Actualmente en México la electrónica y la agricultura son áreas del conocimiento separadas. Un claro ejemplo se presenta en la producción de jitomate (*Lycopersicum esculentum*) producido en invernadero.*

A pesar de que la producción de este cultivo se hace en un ambiente controlado, existen factores que limitan la producción, como el clima, humedad, temperatura, radiación, factores edáficos, plagas, enfermedades, manejo de cultivo y nutrición. Este último factor cuando se maneja de forma adecuada ayuda a incrementar el rendimiento del cultivar.

Para lograr lo anterior es necesario contar con aparatos electrónicos que puedan determinar con precisión los factores que ocurren durante el crecimiento y desarrollo del mismo.

De esta forma en este trabajo se considera la nutrición del cultivo de jitomate. Para esto se utiliza el modelo VegSyst, el cual fue desarrollado para simular la producción de biomasa del cultivo diario, absorción de Nitrógeno (N) y evapotranspiración del cultivo (ETc). Haciendo uso de este modelo matemático y herramientas de simulación se lograron determinar eventos futuros de la producción, lo cual puede ayudar a corregir errores de producción mediante el control de las variables que puedan influir en el rendimiento.

Palabras clave: Jitomate, modelos, parámetros, evapotranspiración, Nitrógeno, biomasa.

1. Introducción

El desarrollo de la ciencia y tecnología implica una constante retroalimentación entre diferentes áreas

y disciplinas. Por ejemplo, actualmente los avances de la mecatrónica en México se usan más en el área industrial pero es de considerar que existen aspectos de esta ciencia que pueden servir para el desarrollo del sector rural sobre todo en la producción de ciertos cultivos bajo condiciones más idóneas para el control de los principales factores que limitan la producción y rentabilidad de los mismos.

Una forma para lograr lo anterior es mediante la simulación computarizada de modelos matemáticos, la cual permite reducir tiempo, esfuerzo y recursos. Con estas herramientas se puede estimar y predecir el rendimiento de cultivos, permitiendo analizar la respuesta a distintas variaciones [1].

Un cultivo que se ha estudiado utilizando la simulación ha sido el jitomate, ya que en México esta hortaliza es la segunda más importante después del chile. Su importancia radica en que posee cualidades muy esenciales para adecuarse a la dieta alimenticia, ya sea para consumo en fresco o procesado, lo que representa una rica fuente de sales minerales y de vitaminas A y C [2].

Cabe mencionar este cultivo es un ejemplo de un sistema biológico, el cual se basa en interacciones químicas y físicas y esto hace que su naturaleza sea altamente no lineal o tenga un comportamiento caótico [3].

Sin embargo la simulación basada en modelos cuantitativos de procesos biológicos y sus interacciones, puede proveer una considerable visión en el comportamiento de estos sistemas y dirigirlos a metas específicas.

Por esta razón los sistemas computacionales han ofrecido poderosas herramientas en las ciencias biológicas, lo cual ha hecho posible estimar y corregir errores en el ciclo del cultivo. Esto último en un

furto podrá ayudar a cumplir con los retos mundiales que se presentan en la búsqueda de producir alimentos sanos y de calidad, así como elevar la producción de los mismos.

Por esta razón surge la necesidad de desarrollar simulación matemática, usando técnicas de control ambiental y desarrollar la electrónica adecuada en los invernaderos donde se produce el jitomate [4].

Cabe mencionar que en la agricultura protegida existen de manera general dos formas para incrementar la producción: a) mejorar las condiciones climáticas, b) aplicar las cantidades óptimas de nutrientes. En este último el elemento más importante es el Nitrógeno, ya que es el principal nutriente que las plantas requieren. Pero cuando se aplican cantidades excesivas las consecuencias más importantes son: toxicidad en la planta, contaminación a los mantes acuíferos y salinidad en los suelos. De aquí el interés de crear nuevas técnicas de control que sean viables y que respeten el ambiente [5, 6].

Con base a lo anterior, en este trabajo se propone analizar el crecimiento de jitomate, considerando la parte nutrimental mediante la simulación del modelo VegSyst, técnicas de control y programación [7], con la finalidad de que se puedan predecir eventos, sin la necesidad de esperar la cosecha, buscando rendimientos potenciales, y generando una dosis optima de solución nutritiva. Tratando de beneficiar a la población local que cuenta con invernaderos con superficies promedio de 1000 m² y con la visión de proteger el ambiente.

2. Materiales y Métodos

El modelo que se utiliza este trabajo es conocido como “VegSyst”, el cual fue desarrollado para simular la producción de biomasa del cultivo, absorción de Nitrógeno (N) y evapotranspiración del cultivo (ETc). Este modelo considera el tiempo térmico y por lo tanto se puede adaptar a las diferentes fechas de siembra, así como diferentes prácticas en el invernadero. También se asume que los cultivos no tienen agua o limitaciones de nutrientes, lo que es real para cultivos manejados intensivamente cultivados en invernaderos [8, 9].

Las entradas del modelo son datos diarios de temperatura máxima y mínima, humedad relativa máxima y mínima, y la radiación solar integral diaria, radiación fotosintéticamente activa (PAR).

Se consideran dos periodos en el ciclo del cultivo: (i) el período 1 del trasplante hasta la máxima intercepción PAR y (ii) el período 2 de máxima intercepción PAR hasta la madurez del cultivo (el fin de la cosecha).

La estructura que presenta el modelo es la siguiente:

$$RTT_1 = \frac{CTT_i}{CTT_f} \quad (1)$$

$$RTT_2 = \frac{CTT_i - CTT_f}{CTT_{mat} - CTT_f} \quad (2)$$

$$f_{i-PAR} = f_0 + \left[\frac{f_f - f_0}{1 + B_1 \exp(-\alpha_1 RTT_1)} \right] \quad (3)$$

$$f_{i-PAR} = f_f + \left[\frac{f_f - f_{mat}}{1 + B_2 \exp(-\alpha_2 RTT_2)} \right] \quad (4)$$

$$B_1 = \frac{1}{\exp(-\alpha_1 RTT_{0.5})} \quad (5)$$

$$B_2 = \frac{1}{\exp(-\alpha_2 RTT_{0.5})} \quad (6)$$

$$DMP_i = PAR_i \times RUE \quad (7)$$

$$\%N_i = a \times DMP_i^b \quad (8)$$

$$kc_i = kc_{ini} + (kc_{max} - kc_{ini}) \left(\frac{f_{i-SR}}{f_{f-SR}} \right) \quad (9)$$

$$kc_i = kc_{end} + (kc_{max} - kc_{end}) \left(\frac{f_{i-SR} - f_{mat-SR}}{f_{f-SR} - f_{mat-SR}} \right) \quad (10)$$

$$f_{i-SR} = 1 - \exp \left[\frac{\ln(1 - f_{i-PAR})}{1.4} \right] \quad (11)$$

$$ET_c = kc ET_0 \quad (12)$$

Donde:

f_{i-PAR} : Fracción interceptada de la radiación fotosintéticamente activa.

DMP_i : Producción de materia seca.

N_i : Contenido de Nitrógeno del cultivo.

ET_c : Evapotranspiración del cultivo.

CTT_i : Acumulación de tiempo térmico.

PAR_i : Radiación fotosintéticamente activa.

T_{max} : Temperatura máxima.

T_{min} : Temperatura mínima.

Hum_{max} : Humedad máxima.

Hum_{min} : Humedad mínima.

CTT_f : Acumulación de tiempo térmico en la máxima intercepción de PAR.

CTT_{max} : Acumulación de tiempo térmico en la madurez del cultivo.

α_1 : Coeficiente de forma.

α_2 : Coeficiente de forma.

RTT_{05} : Tiempo térmico relativo.

f_f : Máxima fracción interceptada de PAR.

f_0 : Fracción inicial de PAR.

f_{mat} : Fracción de PAR en la madurez del cultivo.

RUE : Uso eficiente de radiación.

a : Factor de calibración.

b : Factor de calibración.

kc_{ini} : Coeficiente inicial del cultivo.

kc_{end} : Coeficiente final del cultivo.

kc_{max} : Coeficiente máximo del cultivo.

f_{f-SR} : Máxima fracción interceptada de radiación.

f_{mat-SR} : Radiación solar interceptada en la madurez del cultivo.

ET_0 : Evapotranspiración de referencia.

De las anteriores ecuaciones se definen las entradas (I), salidas (F) y los parámetros (P):

$$F = [f_{i-PAR} ; DMP_i ; N_i ; ET_c] \quad (13)$$

$$I = [CTT_i ; PAR_i ; T_{max} ; T_{min} ; Hum_{max} ; Hum_{min}] \quad (14)$$

$$\theta = [CTT_f ; CTT_{max} ; \alpha_1 ; \alpha_2 ; RTT_{05} ; f_f ; f_0 ; f_{mat} ; RUE ; a ; b ; kc_{ini} ; kc_{end} ; kc_{max} ; f_{f-SR} ; ET_0] \quad (15)$$

Para complementar el modelo se utilizaron dos ecuaciones de evapotranspiración de referencia (ET_0). La primera es la descrita en la ecuación 16, la cual es propuesta por la FAO [10], mientras que en la ecuación 17 se utiliza una función más simple, propuesta en el mismo método, pero es aplicable cuando no se tienen las entradas y parámetros de la ecuación 16.

$$ET_0 = \frac{0.408(R_n - G) + \gamma \frac{900}{T+273} U_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34U_2)} \quad (16)$$

$$ET_0 = 0.0023(T_{mean} + 17.8)(T_{max} - T_{min})^{0.5} Ra \quad (17)$$

Donde:

R_n : Radiación neta en la superficie del cultivo.

G : Densidad de calor del suelo.

T : Temperatura del aire a 2m de altura.

U_2 : Velocidad del viento a 2m de altura.

e_s : Saturación de presión de vapor.

e_a : Presión real de vapor.

Δ : Pendiente de la curva de presión de vapor.

γ : Constante psicométrica.

Ra : Radiación extraterrestre.

T_{mean} : Temperatura promedio.

T_{max} : Temperatura máxima en el invernadero.

T_{min} : Temperatura mínima en el invernadero.

Haciendo uso de las anteriores ecuaciones y las respectivas entradas, se programa el modelo VegSyst.

3. Resultados y discusión

Para resolver el modelo se deben utilizar programas computacionales que ayuden a facilitar el trabajo. Por lo tanto en este trabajo se hace uso del programa MATLAB®, el cual es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, visualización y programación. Usando este programa se pueden analizar los datos, desarrollar algoritmos y crear modelos y aplicaciones [11].

Una vez programado el modelo en un archivo de texto tipo m y en versión Simulink, lo que sigue es analizar los resultados. Primeramente se muestra el comportamiento del modelo:

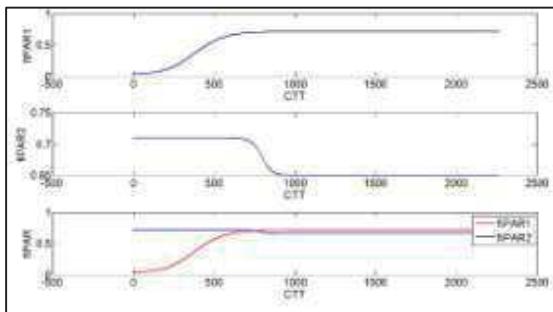


Fig. 1. Fracción interceptada de la radiación fotosintéticamente activa, con respecto a la acumulación de tiempo térmico.

La relación entre la fracción interceptada del PAR en el cultivo y la acumulación de tiempo térmico se presenta en la figura 1. Para obtener la acumulación de tiempo térmico se utilizó el método de triangulación descrito por [12]. También se observa que se tienen dos períodos, el primero representa la fracción interceptada del PAR desde el trasplante hasta la máxima intercepción del PAR, mientras el segundo representa desde la máxima intercepción hasta la madurez del cultivo.

En la figura 2 se muestra el contenido de Nitrógeno en la planta con respecto a la producción de materia seca, en donde se tiene que a mayor materia seca, el contenido de Nitrógeno disminuye, mientras la absorción de Nitrógeno por la planta aumenta con respecto al tiempo, lo cual se observa en la figura 3.

En la figura 4 se observa la producción de materia seca la cual aumenta con respecto a la acumulación de tiempo térmico y esta producción se puede ajustar dependiendo de las fechas de siembra.

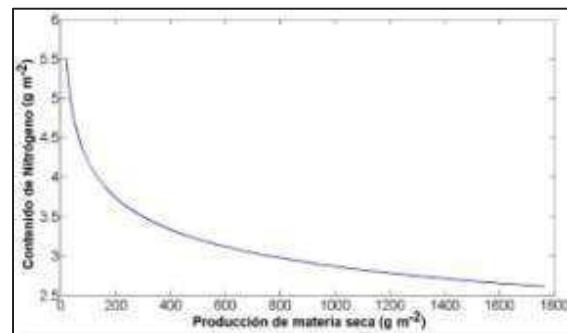


Fig. 2. Contenido de Nitrógeno en la planta con respecto a la producción de materia seca.

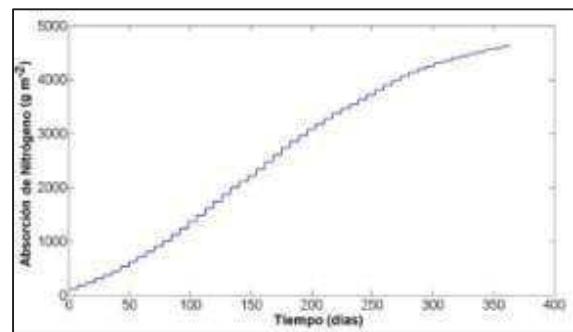


Fig. 3 Absorción de Nitrógeno.

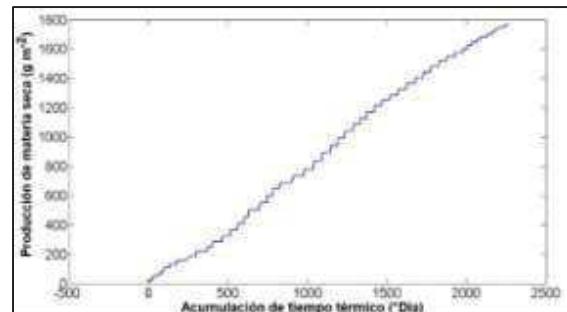


Fig. 4. Producción de materia seca contra la acumulación de tiempo térmico.

En la figura 5 se tienen 4 simulaciones de evapotranspiración del cultivo, utilizando las dos ecuaciones para los dos períodos de f_{i-PAR} descritas por el modelo y de igual manera estas se simularon con las dos ecuaciones $dekc_i$, así como con las ecuaciones 16 y 17 de evapotranspiración de referencia, en donde se puede observar que la ecuación 16 muestra un comportamiento más acorde con la realidad.

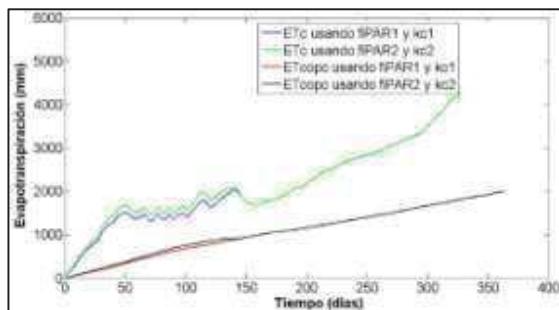


Fig. 5. Evapotranspiración del cultivo.

Con las anteriores simulaciones se puede mostrar que con los modelos matemáticos se pueden realizar experimentos sin la necesidad de tener un cultivo real, esto adaptando los parámetros y las entradas a la región de estudio. Hasta el momento se ha tratado de ajustar las curvas con los parámetros encontrados, pero en próximos trabajos se realizará una validación entre los datos simulados y los reales [13].

4. Conclusiones

Como se ha mencionado existe el interés de aplicar soluciones nutritivas, que sean óptimas, que respeten el ambiente y que puedan incrementar los rendimientos.

Por esta razón en este trabajo se plantea la utilización de un modelo matemático que optimice los niveles de Nitrógeno. Para lograr lo anterior se utilizaron herramientas de simulación, las cuales hacen posible predecir eventos futuros de una producción sin la necesidad de esperar una cosecha para poder corregir errores mediante el control de las variables que puedan influir en el rendimiento de la producción.

De esta manera en este documento se presentaron las simulaciones del modelo VegeSyst, el cual mediante sus ecuaciones pudieron determinar la producción de biomasa del cultivo diaria, la absorción de Nitrógeno por la planta y evapotranspiración del cultivo. Cabe mencionar que para simular la evapotranspiración se consideraron dos ecuaciones de Penman-Monteith, las cuales se simularon para poder ajustar mejor los resultados.

Por lo tanto en las gráficas se pudo observar que mediante la simulación se pueden modificar las variables de entrada, lo cual ayuda a obtener

rendimientos potenciales y el mejor comportamiento del crecimiento del cultivo de jitomate bajo condiciones de agricultura protegida, tratando de incrementar en la vida real la producción en los invernaderos y así beneficiar los ingresos de los productores que siembran dicho cultivo.

En próximos trabajos es necesario continuar con la programación del modelo, tratando de obtener mejores ajustes, así como realizar un análisis de sensibilidad para determinar los parámetros más sensibles al modelo y a su vez realizar la identificación paramétrica de los coeficientes del modelo. Todo lo anterior se desarrolle electrónica adecuada que se pueda adaptar a invernaderos de una específica región y que ayuden a controlar las variables que influyen en el rendimiento del cultivar.

Referencias

- [1] López C. 2002. Efficient evolutionary algorithms for optimal control. PhD Thesis, Wageningen University, Wageningen, Netherlands.
- [2] SAGARPA. 2010. Resumen nacional de producción agrícola. Revisado en: <http://www.contraparteinformatica.com/despliegue-noticia.php?Noticias-Puebla{En-10-anos-crecio-en-10-mil-hectareas-el-uso-de-invernaderos-en-el-pais&id=29665>. Consultada: Julio de 2013.
- [3] Pear M. R, Cuary R. B. *Agricultural systems modeling and simulation*. The University of Florida, Gainesville, Florida. Edit Marcel Dekker Inc, ISBN: 0-8247-0041-4. 1998.
- [4] López I., Ramírez A., Rojano A. 2005. Modelos matemáticos de hortalizas en invernadero: Trascendiendo la contemplación dinámica de cultivos. Revista Chapingo. Serie Horticultura. 11 (2). Universidad Autónoma Chapingo. Chapingo México, pp. 257-267.
- [5] Thornley J. H., Johnson I. R. 2000. Plant and Crop Modelling, A Mathematical Approach to Plant and Crop Physiology, The Blackburn Press. New Jersey, USA. 669 p. Revista Chapingo Serie Horticultura 11(2): 257-267, 2005.
- [6] Thompson RB, Martínez Gaitán C., Gallardo M., Giménez C., Fernández M. 2007. Identification of irrigation and N management practices that contribute to nitrate leaching loss

- from an intensive vegetable production system by use of a comprehensive survey. Agric Water Manage 89:261–274.
- [7] Jones J. W., Dayan E., Allen L., Van Keulen H., Challa H. 1991. A dynamic tomato growth and yield model (TOMGRO). Transactions of the ASAE. 34(2):663-672.
 - [8] Gallardo M., Giménez C., Martínez C., Stöckle C. O. Thompson R. B., Granados M.R. 2011. Evaluation of the VegSyst model with muskmelon to simulate crop growth, nitrogen uptake and evapotranspiration. Agricultural Water Management 101 ELSEVIER (2011) 107– 117.
 - [9] Giménez C., Gallardo M., Martínez C., Stöckle C. O., Thompson R. B., Granados M. R. VegSyst, a simulation model of daily crop growth, nitrogen uptake and evapotranspiration for pepper crops for use in an on-farm decision support system.
 - [10] Allen R.G., Pereira L.S., Raes D., Smith M. 1998. Crop evapotranspiration. Guidelines for computing crop water requirements. FAO Irrigation and Drainage Paper 56. FAO, Rome, Italy.
 - [11] Taps, F. 2000. Economics-based optimal control of greenhouse tomato crop production. Ph. D. Thesis. Wageningen Agricultural University. Wageningen the Netherlands. 127 p.
 - [12] Zalom F.G., Goodell P.B., Wilson L.T., Barnett W.W., Bentley W.J. 1983. Degreedays: The Calculation and Use of Heat Units in Pest Management. Cooperative Extension Publ. Univ. California, 10 pp.
 - [13] Jones J. W., Tsuji G. Y., Hoogenboom G., Hunt L., Thornton K., Wilkens P., López C. 2002. Efficient evolutionary algorithms for optimal control. PhD. Thesis, Wageningen University. Wageningen, The Netherlands, 126 p.
 - [14] Seginer I., Van Straten G., Buwalda F. 2000. Lettuce growth limited by nitrate supply. Acta Horticulturae 507: 141-14.

Simulador Applet de un Péndulo Simple Orientado a la Enseñanza de la Física a Nivel Secundaria

Altamirano Soria José Edgar; Jimenez Nava Jimena, Quijas Martínez Julio, Ramos Gutiérrez Jesús, Ulloa Calzada José Luis, José Emilio Vargas Soto y Jesús Carlos Pedraza Ortega.

Universidad Autónoma de Querétaro – Facultad de Informática
Av. de las Ciencias Sin Número, Campus Juriquilla, Juriquilla, Querétaro, Qro. C.P. 76230 México

Resumen

Java es un lenguaje moderno, tiene buenas características como lo es la programación orientada a objetos, su compatibilidad con diversos sistemas operativos entre otros. Se utiliza NetBeans para construir este proyecto de aplicación práctica en java, que nos permite simular las ecuaciones que modelan el funcionamiento del péndulo simple mediante un Applet, todo como un elemento tecnológico para una mejor enseñanza a nivel secundaria de este fenómeno.

Palabras clave: Java, Applet.

1. Introducción

Como punto de partida conoceremos un poco de historia sobre la invención del péndulo, Jean-Bernard-León Foucault es la persona a quien se le atribuye la invención del péndulo que lleva su nombre. Fue la primera demostración física de la rotación de la tierra, indujo que la oscilación de un péndulo también sería independiente del movimiento de rotación del punto de sujeción al techo, y lo comprobó el 8 de enero de 1851.¹

El lenguaje de programación en Java, es una herramienta muy importante y necesaria que puede utilizarse en diversas áreas como en la industria, educación, entretenimiento, etc., cabe destacar que el beneficio de utilizar Java es la portabilidad. Una de las tareas fundamentales en este proyecto y en cualquier otro es que no existe una metodología definida para la solución, es por ello que hay que hacer un análisis previo tanto del mercado al cual va dirigido tu producto, como del alcance y el impacto del proyecto, para posteriormente de acuerdo a las experiencias del diseñador y a las necesidades del proyecto se tomen en cuenta los diversos modelos matemáticos que hay y las distintas formas de solucionar el problema en cuestión, tomando en cuenta que la educación secundaria presenta un reto

constante. Al trabajar con adolescentes es altamente recomendable la didáctica, dinámicas de grupo, y desde luego tecnología educativa.² Por ello, el propósito de este proyecto es apoyar a los alumnos y reforzar su aprendizaje tanto en el aula como al momento de comprobar resultados; nuestra meta es que se utilice la instrumentación virtual como elemento tecnológico para la enseñanza a nivel secundaria de un fenómeno físico.

2. Comparación de lenguajes de programación

Es importante, antes de comenzar con la programación, saber qué lenguaje utilizar para trabajar. A fin de comprender mejor las ventajas y desventajas de algunos lenguajes de programación, analizaremos la siguiente tabla:

Características	C	C++	Java
Expresividad	Regular	Muy buena	Muy buena
Bien definido	Regular	Muy buena	Muy buena
Estructura de datos	Deficiente	Muy buena	Muy buena
Modularidad	Regular	Muy buena	Muy buena
Facilidades de entrada/salida	Buena	Buena	Buena
Portabilidad	Buena	Buena	Excelente
Eficiencia	Excelente	Excelente	Muy Buena
Estandarización	Buena	Buena	Excelente
Evolución	Estable	Estable	Acelerada
Soporte de librerías	Buena	Muy buena	Excelente

Tabla 1.- Tabla comparativa³

1. Expresividad: Facilidad del lenguaje para expresar los algoritmos.
2. Bien definido: Consistencia y falta de ambigüedad.
3. Tipos y estructuras de datos.
4. Modularidad: Permitir el desarrollo de componentes independientemente.
5. Facilidades de entrada-salida: Soporte para interacción con el entorno.
6. Portabilidad.
7. Eficiencia.

Es importante tomar en cuenta que existen más de tres lenguajes de programación, para fines prácticos solo se mencionan algunos para realizar la comparación.

3. Metodología del proyecto

En la metodología que realizamos, comprende que con anterioridad ya se tenía contemplado el lenguaje de programación a utilizar, tanto por portabilidad como por diseño.

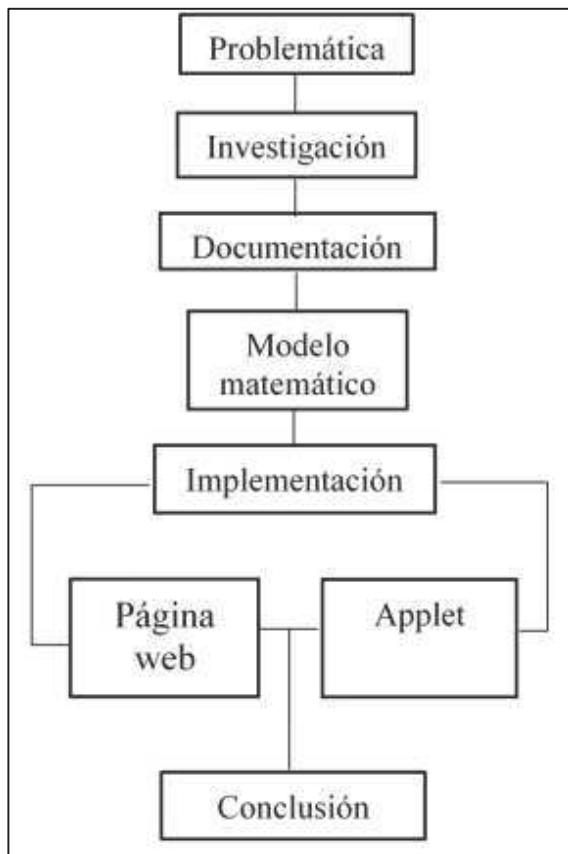


Fig. 1 Metodología del proyecto

La metodología podría cambiar dependiendo de las necesidades que se vayan dando o de las prioridades que se tengan, puede ser por diseño, tiempo en el proceso de implementación, etc. Son muchos factores los que podrían afectar, pero es necesario una metodología ya sea general o particular al momento de comenzar un proyecto, esto nos ayuda a seguir la forma de trabajo dependiendo los requerimientos que tengamos para poder concluir satisfactoriamente.

3.1 Impacto al sector educativo

El sector educativo es decisivo para la formación y el desarrollo del capital humano de cada nación, ya que permite trabajar en la innovación para lograr un crecimiento sostenible. Según datos del Instituto Nacional de Estadística, Geografía e Informática (INEGI), de los mexicanos que asisten actualmente a la escuela, 56% utiliza Internet y 69% usa o ha empleado una computadora en el último año. La importancia del manejo de las Tecnologías de la Información y la Comunicación (TIC) en el contexto actual es crucial para el desempeño económico y educativo del país.⁴

4. Desarrollo del proyecto

Partiremos de lo general a lo particular para poder completar el proyecto, el modelo que se implementó así mismo como información relacionada a la resolución de nuestra problemática.

4.1 El péndulo simple

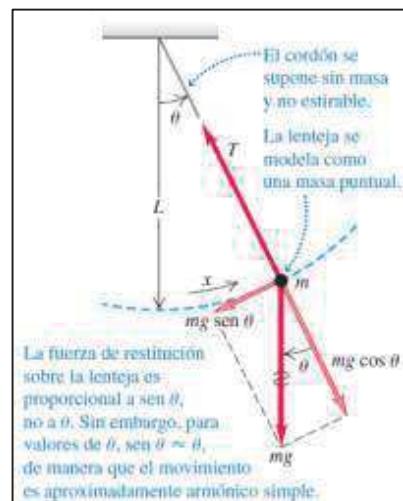


Fig. 3 Péndulo simple idealizado.⁵

Se trata de un sistema con un solo grado de libertad que puede describirse expresando en función del tiempo su distancia angular θ a una dirección de referencia, tal como la vertical en un determinado punto (elongación). Sin embargo, la ecuación diferencial (1) que describe este movimiento tiene una solución analítica en absoluto sencilla.⁶

$$\frac{d^2\theta}{dt^2} + w^2 \sin \theta = 0 \quad (1)^6$$

La pulsación de w depende de la longitud del péndulo L y de la aceleración de la gravedad g ; el término $\sin \theta$ suele tratarse en la mayoría de los textos de física general [Tipler 1992] simplificando para el caso de pequeñas oscilaciones, es decir, cuando la amplitud del movimiento de oscilación alrededor de la posición de equilibrio es muy pequeña. Entonces puede sustituirse la función seno por el ángulo correspondiente, lo cual conduce a una solución de la forma:⁶

$$\theta = \theta_0 \cos(wt + \varphi) \quad (2)^6$$

Donde θ_0 representa la amplitud y φ es una fase inicial que puede ser elegida a voluntad para situar la elongación en un valor conveniente al tomar el origen de tiempos. Con esta hipótesis simplificadora, el periodo de la oscilación está dado por:⁶

$$T = \frac{2\pi}{w} = 2\pi \sqrt{\frac{L}{g}} \quad (3)^6$$

Sin embargo, un estudio más cuidadoso del problema [Kittel et al. 1968; Pippard 1989; Rañada 1990], entrando a considerar la solución analítica de la ecuación (1), permite escribir el periodo de oscilación para cualquier amplitud θ_0 , siendo el resultado dependiente de dicha amplitud como muestra la ecuación (4):⁶

$$T = 2\pi \sqrt{\frac{L}{g}} \left\{ 1 + \sum_{n=1}^{\infty} \left[\frac{1 \cdot 3 \cdot \dots \cdot (2n-1)}{2 \cdot 4 \cdot \dots \cdot 2n} \right]^2 \sin^{2n} \frac{\theta_0}{2} \right\} \quad (4)^6$$

4.2 El péndulo en Java

Una vez habiendo resuelto la problemática matemática, procedemos a interpretarla en código para poder crear el Applet en Java.

Una de las partes más importantes en la programación es estructurar nuestro sistema de acuerdo a las necesidades que se nos presentan. La manera más eficiente de comenzar es creando nuestro

diagrama de flujo para poder interpretar las acciones que nuestro sistema va a realizar.

En la Fig. 4 se puede observar el diagrama de flujo, que de manera general nos muestra el funcionamiento de nuestro sistema. Una vez habiendo realizado nuestro diagrama de flujo procedemos a estructurar la interfaz de nuestro Applet.

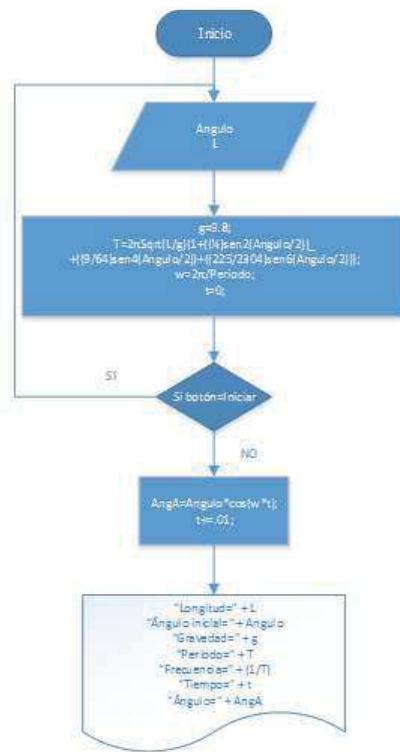


Fig. 4 Diagrama de flujo de nuestro Applet

La interfaz del sistema va a estar estructurado por dos pantallas: 1) Pantalla de presentación, y 2) pantalla de simulación.

1. La pantalla de presentación (Fig. 5) muestra los datos de la universidad de procedencia, los nombres de los integrantes y las instrucciones para poder utilizar el Applet. Se muestran las siguientes barras de Scroll:
 - a. Velocidad de Simulación. Permite manipular la velocidad de la simulación la cual puede contener valores de 0 a 5. Siendo 0 el valor que simula al péndulo en tiempo real (de acuerdo a las capacidades del dispositivo que se utiliza para la ejecución del Applet); los siguientes valores simulan la animación del péndulo

aumentando el tiempo en 0.01s cada 0.05s en 1, 0.1s en 2, 0.2s en 3, 0.4s en 4 y 0.8s en 5. Esto permite observar la simulación de manera detallada, para cuestiones prácticas.

- b. Ángulo inicial. Permite especificarle al sistema qué ángulo deseamos para que se simule en el péndulo, pudiendo ingresar valores de entre 0.25° a 90° , con intervalos de 0.25° .
- c. Longitud de cuerda. Permite especificar la longitud de la cuerda, pudiendo especificarle valores entre 1m y 15m con intervalos de 0.25m.

También cuenta con un botón llamado “Iniciar”, el cual permite, mientras esté la pantalla de presentación, iniciar la simulación del péndulo de acuerdo a los valores que se establecieron en cada una de las barras de Scroll.



Fig. 5 Pantalla de Presentación



Fig. 6 Pantalla de Simulación

2. En la pantalla de simulación (Fig. 6), se puede observar la simulación del péndulo simple de acuerdo a los datos ingresados. También se muestra más información acerca del sistema como son: el periodo, la frecuencia, el tiempo actual desde que se inició la simulación en segundos y la posición angular del péndulo de acuerdo al tiempo actual.

Los dos siguientes extractos de código hacen referencia a las dos funciones esenciales para poder simular y determinar los valores del péndulo simple. La función Calcular, permite calcular el periodo y la velocidad angular, además de inicializar los valores. La función Pendulo, permite determinar el valor del ángulo conforme al tiempo, además de dibujar el péndulo en pantalla con la posición angular.

```
public void Calcular(double ang,double l,double G)
{
    Angulo=ang;
    L=l;
    g=G;
    T = 2 * Math.PI * Math.sqrt( L / g ) * ( 1d + (( 1d / 4d ) * Math.pow(
        Math.sin( Math.toRadians(ang / 2d)), 2d) + (( 9d / 64d ) * Math.pow(
        Math.sin(Math.toRadians(ang/2d)), 4d)) + (( 225d / 2304d ) * Math.pow(
        Math.sin( Math.toRadians( ang / 2d)), 6d)));
    w=(2*Math.PI)/T;
    tiempo=0;
    tiempoi=(double)System.currentTimeMillis();
}

public void Pendulo(double ang,double t)
{
    AngA=(Math.toRadians(ang)*Math.cos(w*t))+Math.toRadians(90);
    gra.setColor(Color.BLACK);
    gra.drawLine((int)X(0), (int)tY(0), (int)tX(AngToX(AngA)), (int)
    tY(AngToY(AngA)));
    gra.setColor(Color.blue);
    gra.fillOval((int)X(AngToX(AngA))-10, (int)tY(AngToY(AngA))-10,
    20, 20);
}
```

5. Análisis de resultados

Se procedió a resolver cada una de las ecuaciones, se generó una tabla para observar de mejor manera cada uno de los resultados, para esto, a la longitud de la cuerda L le asignamos el valor de 1m., a la gravedad se le asigna la gravedad de la tierra suponiendo que esta es aproximada a 9.8m/s^2 y los ángulos manejados son de 90° a 10° , representados en la Tabla 2.

Los pasos realizados fueron los siguientes:

1. Mediante la ecuación (4) se procedió a determinar el periodo (T) de cada uno de los ángulos, tal y como se muestra el resultado en la Tabla 2.

2. Gracias a la ecuación (3), y ya teniendo el valor del periodo para cada uno de los ángulos, podemos obtener la frecuencia angular (w), que también se muestra el resultado para cada uno de los ángulos en la Tabla 2.

θ_0 (°)	L (m)	g (m/s ²)	T (s)	w (rad/s)
90	1	9.8	2.353039	2.670243
80	1	9.8	2.276419	2.760118
70	1	9.8	2.209696	2.843461
60	1	9.8	2.153236	2.918020
50	1	9.8	2.106830	2.982293
40	1	9.8	2.069962	3.035411
30	1	9.8	2.042028	3.076934
20	1	9.8	2.022482	3.106670
10	1	9.8	2.010918	3.124536

Tabla 2. Cálculo del periodo y velocidad angular.

3. Una vez obtenido el periodo y la frecuencia angular, se procedió a utilizar la ecuación (2), para determinar el ángulo en un tiempo definido.

La Tabla 3, nos muestran una comparación entre los resultados matemáticos y la simulación en el Applet de la posición angular del péndulo en distintos lapsos de tiempo.

θ_0 (°)	$\theta(t)$ t=0.5s (°)	Simulación $\theta(t)$ t=0.5s	$\theta(t)$ t=1s (°)	Simulación $\theta(t)$ t=1s
90	21.014917	21.01	-80.186073	-80.19
80	15.166630	15.17	-74.249333	-74.25
70	10.396001	10.40	-66.912090	-66.91
60	6.693227	6.69	-58.506690	-58.51
50	3.978273	3.98	-49.366934	-49.37
40	2.122640	2.12	-39.774720	-39.77
30	0.969706	0.97	-29.937311	-29.94
20	0.349206	0.35	-19.987805	-19.99
10	0.085282	0.09	-9.998545	-10.00

θ_0 (°)	$\theta(t)$ t=1.5s (°)	Simulación $\theta(t)$ t=1.5s	$\theta(t)$ t=2s (°)	Simulación $\theta(t)$ t=2s
90	-58.461665	-58.46	52.884583	52.88
80	-43.319435	-43.32	57.824087	57.82
70	-30.270806	-30.27	57.920795	57.92
60	-19.746513	-19.75	54.101094	54.10
50	-11.834079	-11.83	47.483766	47.48
40	-6.344010	-6.34	39.101417	39.10
30	-2.905065	-2.91	29.749508	29.75
20	-1.047194	-1.05	19.951237	19.95
10	-0.255821	-0.26	9.994182	9.99

Tabla 3. Ángulo resultante en diferentes tiempos

En la Fig.7, a medida de ejemplo, podemos observar el Applet con los resultados de la simulación, con un ángulo inicial de 10° y una longitud de cuerda de 1m.



Fig. 7 Simulación en Applet

Por diseño, únicamente se tomaron dos decimales de precisión en los resultados de la simulación, ya que de esta manera resulta más cómoda la lectura de los valores. El tiempo a excepción de la velocidad de simulación real, aumenta 0.01s por lo cual las mediciones pueden llegar a tener variaciones con las soluciones matemáticas.

5.1 Alcances

Una vez montado el sistema en la web, cualquier persona con acceso a la red, puede utilizarlo de la manera que más le convenga. Este sistema puede ser útil, además de contar con el apoyo de un maestro, para mostrar a niños de educación

secundaria el fenómeno físico de manera más detalla, fácil y agradable.

5.2 Limitaciones

Aunque la mayor parte de la gente ya cuenta con dispositivos capaces visualizar los Applets de Java, es necesario que la gente cuente con el acceso a internet para poder apreciar y trabajar con el Applet.

6. Antonio J. Barbero, Fernando Picazo, Antonio García Cifuentes “Resolución del problema del péndulo sometido a oscilaciones de gran amplitud” [ONLINE] Available at: http://www.uclm.es/ab/educacion/ensayos/pdf/revista8/8_18.pdf [Recopilado el 8 de septiembre de 2014].

6. Conclusión

Gracias a las tecnologías de información, las posibilidades de seguir adquiriendo conocimientos, día con día, resulta más eficaz. Cada vez hay más personas que cuentan con dispositivos capaces de conectarse a internet, lo cual para nuestro propósito se ha vuelto esencial.

Internet a nivel mundial se ha vuelto uno de los principales medios que captan la mayor atención de los usuarios. Aprovechando esto, y las características multiplataforma que nos ofrece Java, podemos realizar aplicaciones didácticas para diferentes niveles de estudio, con el propósito de mostrar de manera virtual los diversos fenómenos que en ocasiones no se pueden captar de manera sencilla en nuestro ambiente cotidiano, pero que simulados podemos visualizarlos y modificarlos a nuestro antojo.

Referencias

1. Cuervo.A () . Fomento a la investigación educativa. [ONLINE] Available at: <http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/2543/A04.pdf?sequence=1> [Recopilado el 17 de Junio de 2014].
2. Cabrera, R. () . Breve historia del péndulo de Foucault. [ONLINE] Available at: <http://neuro.qi.fcen.uba.ar/ricuti/foucault/historia.html> [Recopilado el 17 de junio de 2014].
3. () . Ventajas y desventajas de comparación de los lenguajes. [ONLINE] Available at: http://www.limadata.com/doc/ventajas_c.pdf. [Recopilado el 14 de junio de 2014].
4. Podolsky, R. “La era digital y el impacto educativo”, az Revista de educación y cultura. No. 71
5. Hugh. D, Roger A. “Física universitaria”, Addison Wesley, México, Decimosegunda, 2009.

Control de un Robot con Dispositivos Móviles

Arana-González Lizet-Verenice, Cordero-Victoria Isaac-Jair,
Avecilla-Rangel Francisco-Javier, Juan-Manuel Ramos-Arreguin y Efren Gorrostieta-Hurtado

Universidad Autónoma de Querétaro, Facultad de Informática.
Campus Juriquilla, Avenida de las Ciencias s/n, Juriquilla Querétaro, Qro. CP 76230
jramos@mecamex.net

Resumen

Este proyecto tiene como objetivo darle continuidad a un proyecto pasado, el cual consiste en una herramienta móvil (en adelante 'robot') que tenía funciones didácticas en pruebas de laboratorio. La nueva función implementada consiste en ejercer un control sobre los ejes de movimiento de la misma de manera inalámbrica por medio de un control hecho con un dispositivo móvil (en adelante volante), además de recabar datos en tiempo real de la geo localización del robot y almacenarlos en una memoria extraíble. El control está diseñado bajo la plataforma de operación android, y la comunicación se basa en el protocolo IEEE802.15 teniendo como receptor un micro controlador ATmega 2560 con una memoria flash de 256Kb, embebido en una placa tipo Arduino modelo MEGA.

Palabras clave: Arduino, GPS, bluetooth, Android.

1. Introducción

La robótica ha jugado un papel muy importante en la humanidad, lo cual nos ha permitido ir desarrollando máquinas para facilitar la vida del ser humano.

El desarrollo de robots móviles responde a la necesidad de extender el campo de aplicación de la robótica y el incrementar la autonomía de los mismos dejando en lo posible la intervención humana de lado. Para conocer mejor el tema de la robótica tenemos que conocer el significado de la palabra "robot", el cual aparece por primera vez en 1921, en la obra teatral "r.u.r" (rosom's universal robots) [1], en cuyo idioma la palabra "robot" significa fuerza de trabajo o servidumbre [2].

A si mismo podemos hablar que a partir del desarrollo de la computadora y de los circuitos integrados se logra elaborar el primer prototipo de un

Robot en la década cuarenta. Dentro de dicha década podemos encontrar

La tortuga de Walter, presentada en 1948, que podía reaccionar ante la presencia de obstáculos, subir pendientes y cuando su alimentación comenzaba a ser insuficiente, dirigirse hacia una posición de recarga [3] y años después la compañía unimates, introdujo el primer robot industrial en la General Motors en 1961.

Este proyecto surge con la intención de desarrollar una herramienta que nos ayude a poner en práctica nuestros conocimientos actuales, así como investigar y aplicar nuevos conocimientos.

En este artículo se hace una breve descripción de la metodología que se siguió durante el desarrollo del proyecto; en la primera sección se realizó un análisis comparativo sobre las distintas plataformas móviles, con la finalidad de elegir la que sería más adecuada para nuestros propósitos. [1].

Posteriormente se comenzó a programar la aplicación que funcionaría como control inalámbrico del robot además de realizar la recaudación de los datos de la posición del robot.

En la Sección 3 se detalla el funcionamiento de los componentes en conjunto, las pruebas realizadas y las conclusiones del proyecto.

2. Comparación de plataformas

La decisión de usar un celular como volante, está basada en las capacidades que este nos provee, específicamente una comunicación inalámbrica, de la cual hablaremos más adelante, y un sensor que nos brinda una retroalimentación de la posición del dispositivo en tiempo real, llamado acelerómetro.

Para la elección de la plataforma móvil, se tomaron en cuenta 3 aspectos principales: la

disponibilidad, facilidad de programación y robustez del sistema. Las opciones que se tomaron en cuenta fueron iOS de Apple, Android, y BlackBerry. Despues de analizar lo que cada plataforma ofrece nos decidimos por Android ya que actualmente es la plataforma móvil más usada en el mercado, la programación está basada en Java, que a diferencia de iOS, se puede programar desde múltiples plataformas sin necesidad de licencias especiales.

2.1 Receptor y control

Se realizó una investigación entre 3 diferentes microcontroladores, PIC, Arduino y FPGA; se tomaron en cuenta las ventajas y desventajas de diferentes modelos de los microcontroladores mencionados; finalmente se decidió utilizar Arduino por su facilidad en cuanto a la programación, versatilidad y la tarjeta de desarrollo permite realizar pruebas de forma más rápida y fácil.

El lenguaje de programación de Arduino en una implementación de “Wiring”, que es una lataforma de computación física basada en “Processing”.

Una de las grandes ventajas de Arduino, es que es un sistema multiplataforma, puede trabajar en sistemas operativos como Windows, Macintosh y Linux; también cabe mencionar que el software es de código abierto. Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280.

Los Arduinos seleccionados fueron:

- Arduino MEGA para el control de motores y sensores. Figura 1. Las características de esta tarjeta se muestran en la tabla 1.
- Arduino UNO para el manejo del módulo GPS. Figura 2. Las características de esta tarjeta se muestran en la tabla 2.

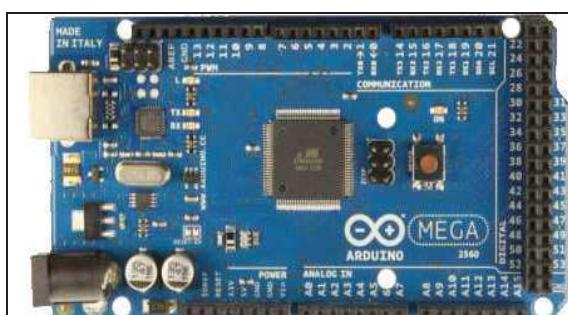


Fig. 1. Arduino Mega.

Voltaje de operación	5V
Voltaje de entrada recomendado	De 7 a 12 V
Puertos de entrada analógicos	16
Puertos de entrada/salida digital	14 (6 proveen salida pwm)
Memoria FLASH	256Kb
Velocidad de reloj	16MHZ

Tabla. 1. Especificaciones Arduino Mega.



Fig. 2. Arduino UNO.

Voltaje de operación	5V
Voltaje de entrada recomendado	De 7 a 12 V
Puertos de entrada analógicos	6
Puertos de entrada/salida digital	14 (6 proveen salida pwm)
Memoria FLASH	32Kb
Velocidad de reloj	16MHz

Tabla. 2. Especificaciones Arduino UNO.

La razón de hacer uso de dos Arduinos es que el GPS cuenta con un shield especialmente diseñado para el Arduino UNO.

2.2 Comunicación

Para que el cambio de dirección del robot tenga efecto conforme a las órdenes del volante, usamos un dispositivo de comunicación inalámbrica Bluetooth (standard 802.15), ya que es una tecnología versátil y compatible con múltiples plataformas móviles, lo cual nos facilita el desarrollo, el modulo que usaremos es un BlueSMIRF-Silver (RN-42) [3]

de Sparkfun, el cual nos brinda una distancia de comunicación optima de hasta 20mts. Este módulo se muestra en la figura 3, y sus características se muestran en la tabla 3.



Fig. 3. Modulo Bluetooth.

Línea de comunicación serial	RX TX
Alcance	18mts.
Flujo de datos	Entre 9600 y 115200bps
Alimentación	De 3.3V hasta 6V
Temperatura de operación	-40 ~ 70C
Consumo de energía	25mA en promedio

Tabla. 3. Especificaciones Bluetooth.

2.3 GPS

El sistema de navegación por satélite ha sido una herramienta muy utilizada en los últimos años, una de sus aplicaciones más importantes es su empleo para la determinación de la posición de cierto usuario u objeto siendo el GPS (Global Positioning System) el nombre el sistema más extendido para este tipo de tareas.

Los sistemas de posicionamiento o radio-localización por satélite nos ofrece una referencia temporal muy precisa gracias a una constelación de satélites en el espacio es posible que los usuarios puedan obtener su posición en tres dimensiones: latitud, longitud y altura [2]. El módulo GPS se muestra en la figura 4 y sus características se muestran en la tabla 4.

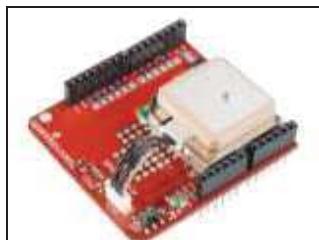


Fig. 4. Modulo GPS.

Canales	20.
Sensibilidad	-159 dBm.
Frecuencia	L1 1575.42 MHz.
Velocidad de actualización	1 us.
Corriente	44mA.
Voltaje	4.5 ~ 6.5V DC.

Tabla. 4. Especificaciones GPS.

Este componente al formar parte de dicho robot puede presentar grandes ventajas a futuro, una de ellas es la relación con la geodesia que facilitan mucho las medidas de terrenos de grandes extensiones e incluso lugares de difícil acceso.

2.4 MicroSD Shield

El sistema diseñado incluye un módulo para el manejo de una tarjeta Micro SD, para el almacenamiento de datos, y su posterior análisis en una computadora personal o en cualquier otro sistema digital. Este módulo se muestra en la figura 5, y sus características se muestran en la tabla 5.



Fig. 5. MicroSD Shield para Arduino

Voltaje de operación	3.3V
Medio de Comunicación	Interface SPI
Implementación de Sistema	FAT 16 ó FAT 32
Memoria Interna	1KB
Capacidad de Memoria de Soporte	256MB- 32GB

Tabla. 5. Especificaciones MicroSD.

3. Desarrollo

El trabajo se desarrolló de acuerdo a la metodología mostrada en la figura 6, donde se muestra el desarrollo del proyecto dividido en varias etapas.

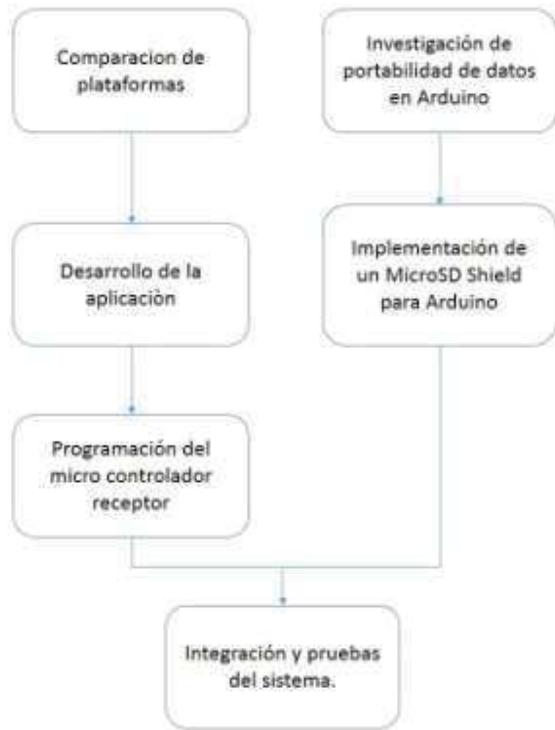


Fig 6. Metodología desarrollada.

3.1 Interacción de Componentes:

En la figura 7 se muestra la distribución de los componentes en su estado final. La parte superior corresponde al área de reacción y comunicación del robot, ya que aquí se encuentra la conexión del módulo bluetooth que permite recabar provenientes del dispositivo móvil y procesarlos con el Arduino MEGA mediante señales digitales. Una vez que el Arduino obtiene dichos datos, es el encargado de procesarlos y de acuerdo a la programación dispuesta por el usuario, y accionar los puentes H con diferentes señales PWM. Posteriormente los motores en la última línea de la conexión responden a los estímulos mencionados y giran desplazando el robot de forma correcta según se necesite.

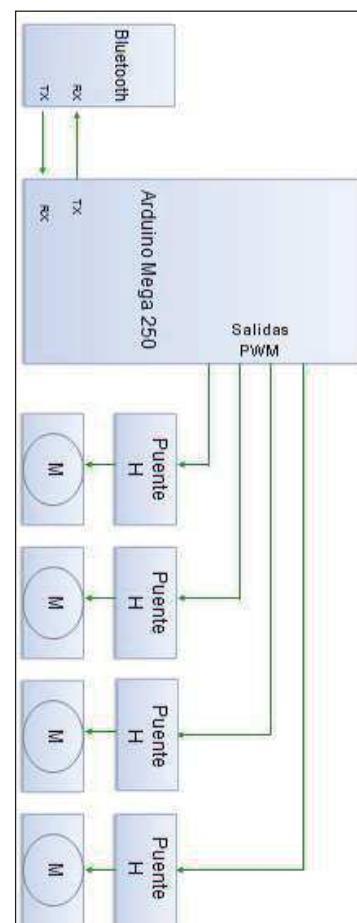


Fig. 7. Diagrama de bloques para el control de motores.

La figura 8 muestra la interacción entre el Arduino UNO y los módulos GPS y de la tarjeta Micro SD.

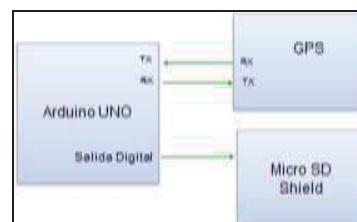


Fig. 8. Diagrama de bloques para el control de los sistemas GPS y Micro SD.

4. Programación.

En esta sección se habla de la programación del dispositivo móvil, utilizando el sistema operativo Android.

4.1 Programación del volante.

Como se hizo mención previamente, el dispositivo que elegimos para que hiciera de volante, fue un celular que contara con sistema operativo Android. En cuanto a los requerimientos para la aplicación que se desarrolló, el dispositivo debe contar con una versión de OS Android superior a 3.0 en cuanto a software, y en cuanto al hardware se requiere comunicación bluetooth y acelerómetro.

La programación del volante, se realizó en su totalidad por integrantes de este equipo y con uso de la herramienta gratuita Eclipse y el plugin ADT en el lenguaje de programación Java [4].

Para la aplicación se realizó una primera activity (Fig. 9), la cual tiene la finalidad de corroborar que el dispositivo cuente con comunicación bluetooth, así como verificar el estado del mismo. Para poder pasar a la siguiente parte que es específicamente el control del vehículo, es primordial que el bluetooth se encuentre encendido.

El diseño se basa en que una buena aplicación consta de pocos “clicks” para llegar a la meta, por ello se optó por un diseño minimalista e intuitivo para el usuario.



Fig. 9. Pantalla principal de la aplicación.

La figura 10 muestra la pantalla correspondiente a la actividad desarrollada para

buscar el módulo Bluetooth conectado en el robot móvil desarrollado previamente.



Fig. 10. Actividad para la conexión Bluetooth entre el dispositivo móvil y el robot móvil.

La siguiente parte de la aplicación, se enfoca en el control de la dirección del vehículo, al mismo tiempo que controla el área de iluminación de los faros del mismo. En la Fig. 11 se explica la función de cada botón de la aplicación.

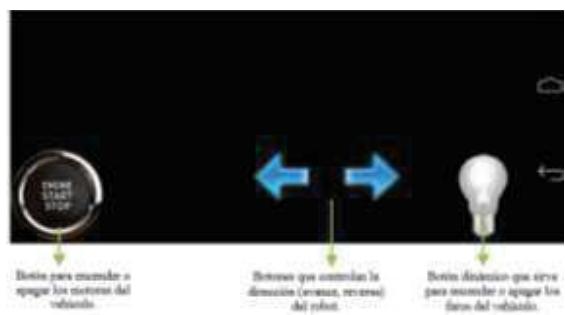


Fig. 11. Pantalla de control del vehículo.

Hasta ahora, hemos visto la forma en que podemos controlar algunos aspectos del vehículo, sin embargo lo primordial, al menos al referirnos a la idea principal del proyecto, es la dirección del vehículo. La dirección, como previamente se habló se logra a través de los valores que nos devuelve el acelerómetro del volante, los cuales son tres y corresponden a los tres ejes cartesianos X, Y y Z. La

manera en que esto funciona de ejemplifica mejor en la Fig. 12.



Fig. 11 Ejes en los que el dispositivo tiene un valor de posición

Como se observa en el código de la Fig. 12 y tomando en cuenta que la posición del dispositivo cuando se usa la aplicación, es en modo “Landscape” (horizontal), solo tomaremos en cuenta los valores de posición correspondiente al eje Y, ya que estos cambiaran cuando el dispositivo se incline a hacia la izquierda o hacia la derecha [4]. La manera en que dicho valor es recabado se muestra en la Fig. 12.

```
public void onSensorChanged(SensorEvent event){
    val = (int)event.values[1];
    txtY.setText("Y = " + val);

    if (val < 0){val = val * -10;}

    if (val >= 0 && val <= 2){key = 0;}
    if (val >= 3 && val <= 5){key = 1;}
    if (val >= 6 && val <= 8){key = 2;}

    if (val >= 10 && val <= 20){key = 0;}
    if (val >= 30 && val <= 50){key = 3;}
    if (val >= 60 && val <= 80){key = 4;}
```

Fig. 12. Función listener que maneja el acelerómetro.

Como ya se mencionó, cada que el dispositivo sufre un ligero cambio, obtenemos un valor en X, uno en Y y uno en Z, pero como solo nos importa el valor en Y, se emplea la variable *val*, para recabar específicamente el valor de este eje.

Así mismo, se delimitan regiones en las que tomaremos valores específicos, ya que existe una gran cantidad de valores arrojados, esto debido a la sensibilidad del acelerómetro.

Los valores o regiones que delimitamos corresponden a posiciones de la Fig. 13.

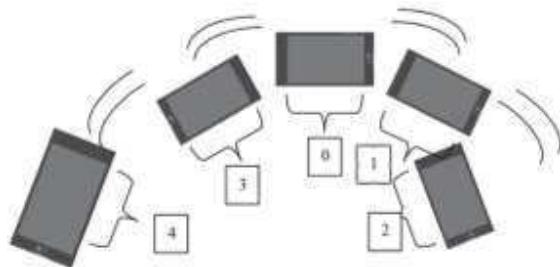


Fig. 13 Valor que toma el volante según su posición en el espacio

4.2 Programación del receptor.

Para la programación del micro controlador, se realizó con la herramienta gratuita de desarrollo de software de Arduino. La programación se basa en una función *Switch()*, que básicamente dependiendo del valor que recibe por medio del bluetooth, se ejecuta el código de el “case” correspondiente. En la Fig. 14 se muestra un menú de los case que se emplean para dar órdenes a los actuadores que en este caso son los 4 motores del vehículo y los distintos leds del cada faro.

```
void loop() {
    if(Serial.available() > 0){
        if(Serial.read() == '0'){
            Serial.println("0");
        }
    }
}

//VALORES
//00 - Volante estatico. (Control de Motores)
//01 - Volante gira derecha leve. (Control de Motores)
//02 - Volante gira media pronunciado. (Control de Motores)
//03 - Volante gira izquierda leve. (Control de Motores)
//04 - Volante gira angularmente pronunciado. (Control de Motores)
//05 - Se encienden los faros.
//06 - Se encienden los intermitentes.
//07 - Se apagan los intermitentes.
//08 - Se apaga los faros.
//09 - Enciende intermitente izquierdo.
//10 - Apaga intermitente izquierdo.
//11 - Enciende intermitente derecho.
//12 - Apaga intermitente derecho.
//13 - Volante estatico. (Control de Faros)
//14 - Volante gira derecha leve. (Control de Faros)
//15 - Volante gira derecha pronunciado. (Control de Faros)
//16 - Volante gira izquierda leve. (Control de Faros)
//17 - Volante gira angularmente pronunciado. (Control de Faros)
```

Fig. 14 Menú que muestra la acción que toma el vehículo dependiendo de la orden del volante.

4.3 Sistema de captura de datos para geolocalización.

El sistema de captura de datos consta de 3 componentes:

- GPS – Sensor
- MicroSD Shield – Almacenamiento de datos
- Arduino UNO – Microcontrolador

Una vez conectados los componentes entre sí, el Arduino se encarga de recabar los datos del GPS, el cual triangula la posición del robot a través de los datos que recibe de 4 satélites, esto nos permite obtener datos de latitud, longitud y altitud; una vez procesados los datos, son almacenados en la memoria SD. Todo esto, se ilustra en el diagrama a bloques de la figura 15.



Fig. 15. Diagrama a bloques del proceso del Arduino UNO.

5. Análisis de resultados

En base al estudio llevado a cabo y la información recabada durante este periodo nos percatamos de la importancia que conlleva la elección de los componentes de manera tal que estos nos brindaran una óptima respuesta a la exigencia de calidad que el proyecto demandada; por ello las pruebas a las que estos fueron sometidos de manera independiente y en conjunto, nos proveen una certeza plena del el buen desempeño del robot.

Durante la investigación establecimos que el robot mantiene una estrecha relación entre el peso y la potencia de los motores; lo cual refutamos mediante observaciones prácticas, a lo cual respondimos empleando una fuente de alimentación que permitiese al robot desplazarse con fluidez.

En base al objetivo planteado en un inicio, esta herramienta es útil con fines didácticos y satisface la necesidad de apoyo en material práctico para impartir algunas materias y contribuye a mejorar la comprensión de los estudiantes ya que provee al alumno una interacción amigable tanto en hardware como en software. La figura 16 muestra la información de los valores que son devueltos y su discretización.

COM4	
	COM4
270	5 cm
251	5 cm
264	6 cm
249	8 cm
431	10 cm
781	11 cm
1250	12 cm
1547	13 cm
1824	19 cm
1339	25 cm
671	30 cm
417	33 cm
869	34 cm
1376	35 cm
1696	35 cm

Fig. 16. Valor devuelto y valor discretizado.

Dado que le sensor nos devuelve señales de voltaje, es necesario discretizar las mismas para poder comprenderlas y usarlas, en la figura se muestra un

ejemplo: a la izquierda observamos el valor (en volts) devuelto por el sensor, y a la derecha observamos el mismo valor (en mts) después de haber sido procesado mediante una ecuación de equivalencia (volts-mts). La figura 17 muestra los valores obtenidos del sistema GPS.

Altitude (meters)	Course (degrees)	Speed (kmph)
1057.00	300.96	0.52
1857.00	285.87	0.81
1857.10	285.87	0.81
1857.10	290.97	0.26
1857.20		

Fig. 16. Información obtenida del GPS.

Datos recabados en una de las pruebas finales, en la imagen se puede apreciar la forma en que se despliega la ubicación del robot en un instante determinado.

6. Conclusiones

El rápido avance de los sistemas computacionales embebidos, ha permitido el diseño y construcción de robots móviles con distintas finalidades, un uso muy importante que se les ha dado es la adquisición de datos con la finalidad de reducir las horas hombre en la misma actividad.

Este robot móvil presenta un avance importante a la hora de recabar datos de terrenos como son la altitud y longitud gracias a la implementación de un GPS.

Gracias a la escalabilidad del mismo, hay opciones a futuro de implementar mejoras y así contar con un sistema cada vez más robusto que pueda cumplir con más funciones.

El desarrollo de este proyecto contó con el apoyo de rectoría, a partir del programa de Fondos Especiales de Rectoría, de la Universidad Autónoma de Querétaro.

Referencias

- [1]. Capek K. (2001). R.U.R. (Rossum's Universal Robots), New York: Dover
- [2]. Asimov I. (1961). I Robot. Doubleday.
- [3]. Baturone (2001). Robótica Manipuladores y Robots Móviles, Barcelona: MARCOMBO, S.A.
- [4]. Moore K., Davidson M., Bahl V., Rich S., Jirgal S. Modelling and Control of a Six-Wheeled Autonomous Robot, Proceedings of the American Control Conference, Chicago, Illinois, 2000.
- [5]. Huerta E., Mangiaterra A., Noguera G.“GPS Posicionamiento satelital.” UNR Editora, Volumen 1, 136 pg., 2005.

Uso de los Software LinuxCNC y Eagle 6.4.0. en el Diseño y Fabricación de Circuitos Impresos por Medio de un Robot Cartesiano

Urbalejo Contreras Arturo¹, Jiménez López Eusebio², Luna Acosta Noé³, Valdez Palafox Carlos Omar⁴, José Antonio Cuadras Gerardo⁵ y Castro Bojórquez Julio Cesar⁶

^{1,3}Centro de Investigación Aplicada en Automatización y Mecatrónica de la Universidad Tecnológica del Sur de Sonora, arturouc@yahoo.com.mx, urbalejo@uts.edu.mx

²Universidad La Salle Noroeste – CIAAM-UTS-IIMM, cjmenezl@msn.com

^{4,5}Universidad Tecnológica del Sur de Sonora

⁶Universidad Tecnológica de Tijuana, Tijuana Baja California, México

Resumen

El software libre es una opción económica y versátil para el diseño, manufactura y control de sistemas mecatrónicos. Linux es un sistema operativo abierto o libre que tiene diversas aplicaciones, como por ejemplo para el control de la manufactura. LinuxCNC o EMC2 es un software que corre en Linux y se usa para el manejo del Control Numérico por Computadora (CNC). En este artículo se describe el uso del LinuxCNC para el control de un robot cartesiano el cual realiza una operación de fresado en la fabricación de circuitos impresos. Se usa el software CAD Eagle 6.4.0 para el diseño de las tarjetas y para la transferencia de información al software LinuxCNC. El robot cartesiano es movido por tres motores a paso y con transmisiones de tornillo.

Palabras clave: LinuxCNC, Eagle 6.4.0, Mecatrónica, Robótica, Circuitos Impresos.

1. Introducción

El aprendizaje de la disciplina Mecatrónica se ha convertido en un reto para las universidades mexicanas. Algunas instituciones educativas les es difícil competir en equipamiento e infraestructura con las principales potencias en esta área, sin embargo, cabe mencionar que existen un sinnúmero de recursos libres tanto de hardware como de software que hacen que la tarea de enseñar-aprender la Ingeniería Mecatrónica sea asequible a la mayoría de los estudiantes y profesores [1]. En países emergentes como México, existe el problema de la falta de estandarización industrial, ocurre que la

tecnología de Máquinas Herramienta de Control Numérico dado su elevado costo, es sólo accesible para las grandes empresas. La micro, pequeña y mediana industria metal-mecánica nacional, en su mayoría, no tiene acceso a este tipo de tecnología, por lo que utilizan máquinas herramientas convencionales o manuales, lo cual produce un deterioro en la calidad de la manufactura. Por otra parte para lograr elevar el nivel de capacitación de los estudiantes de ingeniería las universidades y centros de capacitación, deben realizar fuertes inversiones monetarias en equipos de control numérico computarizado (CNC), lo anterior no siempre resulta factible dado el costo de inversión en los equipos y el mantenimiento requerido [2]. Por lo anterior, es importante generar prototipos propios en donde sea posible utilizar software y hardware libre o de bajo costo con la finalidad de desarrollar tecnología para la enseñanza de la Ingeniería Mecatrónica, particularmente en las aplicaciones del Control Numérico por Computadora (CNC).

Dentro de la amplia gama de software libre que se encuentra disponible hoy en día destaca por su gran versatilidad en su implementación en prototipos que se usan para maquinado, el software LinuxCNC el cual opera bajo el sistema operativo Linux. Un CNC de código abierto como LinuxCNC, consiste en un programa de software para control numérico, del cual se encuentra accesible de manera gratuita el código fuente de programación. El usuario tiene el poder de utilizar, modificar y compartir el código fuente [3]. El LinuxCNC ha sido implementado con éxito de manera global, lográndose operar de manera satisfactoria máquinas de cinemática sencilla tales como tornos [4] y fresadoras [5]. La Figura 1

muestra una ventana operativa del software Linux CNC.

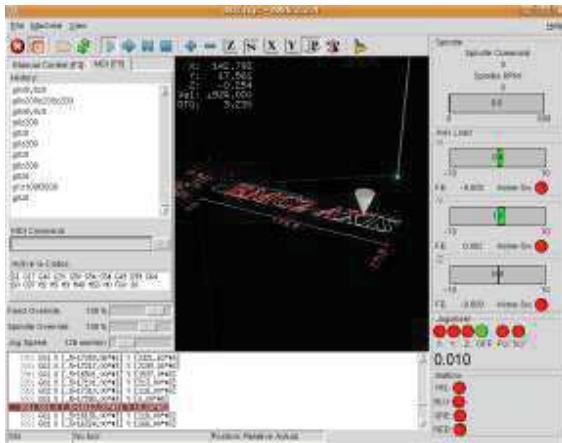


Fig. 1. Software LinuxCNC.

Otro software libre de fácil manipulación es el Eagle 6.4.0 el cual se utiliza para el diseño de tarjetas electrónicas, particularmente para diseño tarjeas de circuitos impresos o placas PCB. Este software tiene diversas aplicaciones. Por ejemplo, en [6] se ha utilizado Eagle para el diseño de una tarjeta con aplicaciones en el diseño de un estetoscopio de bajo costo. En [7] se desarrolló una plantilla para monitorear el caminar de niños autistas. Se diseñó una tarjeta electrónica usando Eagle para el desarrollo de la plantilla. La Figura 2 muestra una pantalla del software Eagle.

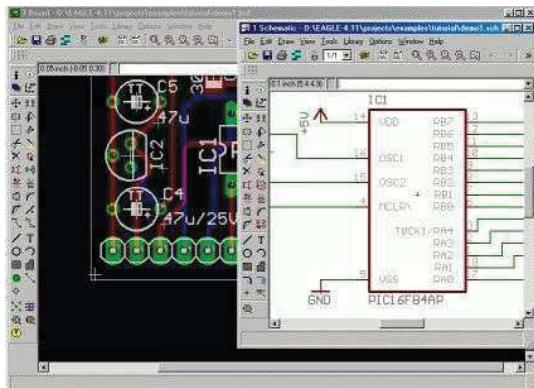


Fig. 2. Software Eagle 6.4.0.

En este artículo se presenta la aplicación de los softwares libres LinuxCNC y Eagle 6.4.0 en el diseño y el maquinado de tarjetas electrónicas. El maquinado se trata de un fresado realizado por una herramienta la cual está montada en un robot cartesiano de 3 GDL (Grados de Libertad).

2. Algunas consideraciones sobre el software libre y LinuxCNC

Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software [8]:

- 1) La libertad de usar el programa, con cualquier propósito.
- 2) La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades. El acceso al código fuente es una condición previa para esto.
- 3) La libertad de distribuir copias, con lo que se puede ayudar a otras personas.
- 4) La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es requisito previo para esto.

El LinuxCNC se basa en la filosofía del software libre. El sistema LinuxCNC se basa en el uso de librerías de tiempo real del sistema operativo de libre distribución Linux capaces de ser ejecutados en una computadora personal (PC), actualmente LinuxCNC puede comunicarse con la máquina a controlar mediante el puerto paralelo o una tarjeta de expansión PCI (Interconexión Periférica de Componentes) y puede ser ejecutado en computadoras personales de arquitectura Intel dotadas con una unidad de procesamiento de al menos 700MHz de velocidad y 384 Mb de memoria de acceso aleatorio RAM. LinuxCNC está diseñada para capturar las principales características de la tecnología CNC y permite desarrollar un control específico que esté acorde con las características que requiere los sistemas mecánicos, eléctricos y electrónicos de una máquina herramienta [3].

3. Descripción del problema

Los alumnos de TSU (Técnico Superior Universitario) de la carrera de Mecatrónica y de Ingenierías de la carrera de Tecnologías de la Automatización de la Universidad Tecnológica del Sur de Sonora (UTS) enfrentan día con día problemas al elaborar una PCB (Printed Circuit Board) para sus prácticas y proyectos. Al fabricar una PCB manualmente se tienen distintas complicaciones, algunas de ellas se resumen a continuación:

-La impresión del circuito no es de la mejor calidad para planchar en la placa.

-Se pueden presentar errores de movimiento de la persona que está elaborando la PCB.

-Por el mal planchado al momento de introducir la placa al ácido, se tiene que las pistas tengan problemas de comunicación o simplemente no funcione la PCB.

-El proceso del diseño del circuito, la impresión, el planchado y el vertimiento de la placa en el ácido, es un proceso muy tardado y se pueden tener algunos errores durante la elaboración de la misma PCB.

-Uno de los problemas principales que se tienen es con el material de la hoja de impresión, no se utiliza el material adecuado y la tinta no se transfiere adecuadamente a la placa.

-El material que se utiliza al momento de realizar un circuito impreso como: la placa fenólica, el ácido para delimitado de las pistas, las impresiones en el papel indicado y el tiempo, son los aspectos que se pierden al momento de un error en la elaboración del circuito impreso.

Para solucionar los problemas descritos anteriormente se propuso diseñar un sistema automático que auxiliara la fabricación de los circuitos impresos. El sistema debería de cumplir con los siguientes aspectos:

- 1) De bajo costo.
- 2) Se debe reducir el tiempo de fabricación de la tarjeta.
- 3) El diseño y propuesta de evitar el uso de materiales no necesarios para el proceso.
- 4) El Sistema debe robusto.
- 5) El robot formará parte de una celda de manufactura para propósitos didácticos del proyecto PROMEP de la UTS y para hacer pruebas de sincronización en celdas de manufactura de las universidades que integran la red ALFA.

El prototipo debería poder usarse tanto en la fabricación de las tarjetas como en la enseñanza de la mecatrónica.

3. Propuesta de solución para el desarrollo del prototipo.

Debido a la robustez y al bajo costo del prototipo, fue necesario evaluar y valorar diseños de otros autores con el objetivo de estudiar configuraciones y actuadores [9], así como

paquetes computacionales. Después de haber discutido algunas ideas, se propuso el siguiente desarrollo del prototipo:

- 1) Sistema cartesiano para el movimiento.
- 2) Tecnología CNC para el control de la fabricación.
- 3) Sistema motriz de precisión.
- 4) Utilización de software libre, tanto para el dibujado de los circuitos, como para su fabricación.
- 5) Utilizar motores a pasos y control por tarjetas comerciales.

4. Propuesta de tecnologías

Una vez elegido el camino y las ideas principales para el desarrollo del prototipo se tomó la decisión de utilizar los siguientes paquetes y plataformas computacionales, tecnologías de la fabricación y control:

- 1) Eagle 6.4.0. para el diseño en CAD de los circuitos.
- 2) LinuxCNC para el control de la fabricación.
- 3) Motores a paso bipolares.
- 4) Fabricación por fresado de tarjetas.
- 5) Transmisión por sistema de tornillo.
- 6) Tarjeta ToshibaTB6560AHQ

5. Desarrollo del prototipo

El desarrollo del prototipo se sistematizó en tres partes: 1) Diseño y fabricación del prototipo, 2) Uso de paquetes computacionales, 3) Control del robot.

5.1 Diseño y fabricación del prototipo.

El prototipo mostrado en la Figura 3 fue diseñado considerando robustez, puesto que será utilizado con alta frecuencia por alumnos y profesores en la fabricación de circuitos impresos para las prácticas y proyectos de clase. El prototipo consta de tres motores a pasos bipolares que mueven tres eslabones integrados por sistemas de tracción y movimientos tipo tornillo sin fin. Una herramienta de corte Dremel es incorporada al sistema y por medio de la cual se realiza el fresado de las tarjetas. El prototipo está montado sobre una mesa la cual es fijada al piso con tornillos.

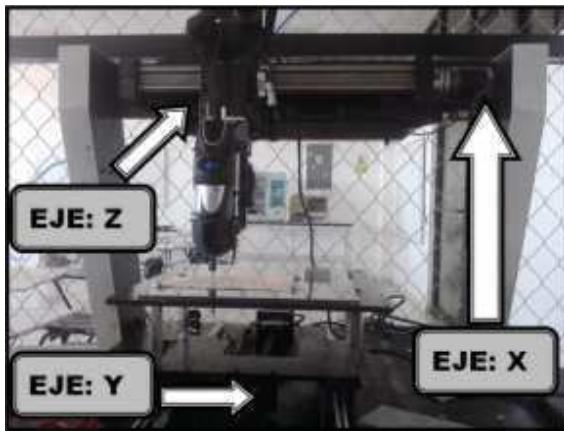


Fig. 3. Prototipo de robot cartesiano

5.2 Uso de paquetes computacionales

Para el diseño y el control de las tarjetas y el robot se usaron Eagle 6.4.0 y LinuxCNC, respectivamente. La Figura 4 muestra un proceso de integración CAD/CAM [9]. Para el caso del robot mostrado en la Figura 3 dicha integración es Eagle 6.4.0/ LinuxCNC.

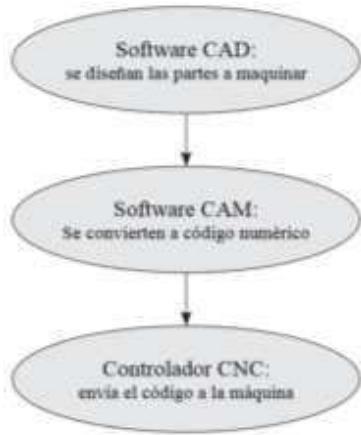


Fig. 4. Proceso CAD/CAM

El proceso general es el siguiente: en el software Eagle 6.4.0. se diseña un circuito, previamente analizado y calculado. Posteriormente, el diseño es transferido por medio de archivos al sistema LinuxCNC en donde la información se convierte en Código Numérico con el cual se controlarán los motores del robot. La Figura 5 muestra un circuito diseñado en Eagle 6.4.0.



Fig. 5. Circuito dibujado en Eagle 6.4.0.

Por otro lado, las máquinas en las que se desee implementar un sistema LinuxCNC deberán de contar con algunas secciones básicas que son universales sin importar del tipo de proceso industrial que se requiera controlar. Para el caso del software LinuxCNC la Figura 6 presenta de manera esquemática los componentes básicos requeridos para su implementación en una máquina herramienta. Es de vital importancia recalcar el uso de tarjetas de aislamiento entre la computadora huésped en la cual se ejecuta LinuxCNC y las etapas de potencia de los motores y husillos, ya que sin este aislamiento se pone en riesgo la integridad de la computadora ante la presencia de fallas eléctricas en las etapas de potencia [3].



Fig. 6. Componentes requeridos para la implementación básica de LinuxCNC en una máquina herramienta.

Para el caso del robot cartesiano motivo de estudio fueron usados motores a pasos sin encoders. La Figura 7 muestra una simulación de fresado de tarjetas en LinuxCNC.

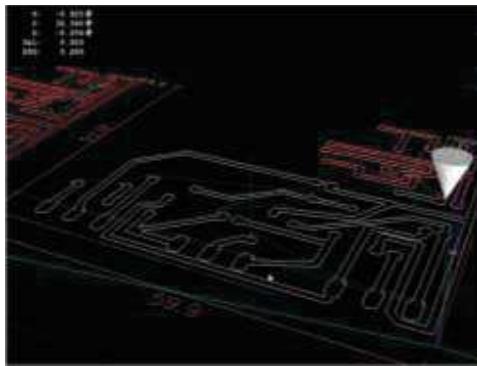


Fig. 7. Simulación de fresado de tarjetas en LinuxCNC

5.3 Control del robot.

Para el caso del control implementado en el prototipo a continuación se enlistan las características técnicas:

- Tarjeta controladora para 4 motores bipolares de 4, 6 u 8 hilos los cuales pueden operar simultáneamente.
- Fuente de alimentación de corriente directa, recomendable 24V.
- Circuitos integrados **TOSHIBA TB6560AHQ** 3.5 amperes por eje.
- Micro pasos 1,1/2, 1/4, 1/16 para mayor precisión.
- Corriente ajustable por eje - 25%, 50%, 75%, 100%
- Protección contra sobre-Carga, Sobre-corriente y temperatura.
- Interface paralela opto-aislada.
- Relevador mecánico interno para controlar una salida de hasta 10 Amperes
- 4 Canales de entrada, límites o EStop.
- Soporta la mayoría del software de control CNC por puerto paralelo MACH3, KCAM4, EMC2 etc.
- Entradas por Puerto DB25 Paralelo.
- Expansión a cuarto eje ya que da acceso a las señales para colocar un driver externo.

Las Figuras 8 y 9 muestran la tarjeta considerada y la configuración de un motor a pasos en el software LinuxCNC.

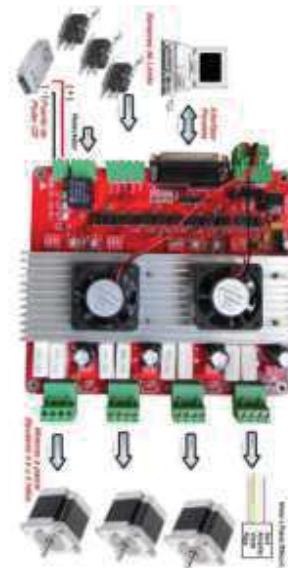


Fig. 8. Tarjeta de control.



Fig. 9. Configuración de un motor a pasos en LinuxCNC

Finalmente, la Figura 10 muestra el prototipo de del robot fresando una tarjeta.



Fig. 9. Fresado de una tarjeta con el prototipo.

6. Conclusiones

En este artículo se ha descrito la aplicación de los softwares LinuxCNC y Eagle 6.4.0, para dibujar y fabricar tarjetas electrónicas en un

prototipo de robot cartesiano. Las conclusiones se resumen en los puntos siguientes:

-El uso del software libre en aplicaciones mecatrónicas es una gran opción frente a los paquetes computacionales con licencia y altamente costosos. Los softwares LinuxCNC y Eagle 6.4.0 resultaron ser útiles para diseñar y fabricar circuitos impresos o placas PCB por medio del robot cartesiano diseñado por los alumnos y profesores de la Universidad Tecnológica del Sur de Sonora.

-Una de las ventajas que se tiene al utilizar Linux CNC en la fabricación de placas PCB es la no utilización de modelos matemáticos para la cinemática y dinámica de los movimientos del robot. Aunque esta ventaja es útil para las aplicaciones rápidas, lo cierto es que los alumnos deben conocer los modelos matemáticos que están soportando el LinuxCNC, ya que las matemáticas y la física son en realidad el corazón de la Mecatrónica.

-La construcción de prototipos mecatrónicos de bajo costo, como es el caso del robot cartesiano descrito en este artículo, ayuda a las universidades a equiparse para las prácticas de la enseñanza y el aprendizaje de la ingeniería. Además, el prototipo ayuda a los alumnos de TSU e ingenierías de la UTS a fabricar con precisión las placas PCB para sus proyectos en diversas clases.

Agradecimientos

Los autores de este trabajo agradecen a las universidades y empresas que integran la RED RECAMEN (Red de Cuerpos Académicos de Mecatrónica del Noroeste) y a la empresa Spin – OFF Innovación en Ingeniería de Manufactura y Mantenimiento S. de R.L. MI, al Cuerpo Académico CIAAM de la UTS, al Programa del Mejoramiento del Profesorado (PROMEP) bajo el proyecto Desarrollo de un Prototipo de Robot Industrial para Aplicaciones en Celdas de Producción Automatizadas Clave UTSSON-CA-4, por el apoyo brindado para la realización de este trabajo.

Referencias

- [1] Curiel F., Jiménez E. Aproximación metodológica en la enseñanza de lenguajes de Programación orientados a sistemas mecatrónicos. *13º Congreso Nacional de Mecatrónica*. Octubre

30 a Noviembre 1. Querétaro, Qro. (Por presentarse). 2014.

[2] Chaides O., Ávila S., Horacio Ahuett H. Restauración de Torno de Control Numérico Empleando Software Libre. *Conciencia Tecnológica* No. 42, Julio-Diciembre. 2011

[3] Chaides O., Ávila S., Horacio Ahuett H. Implementación de un control numérico de código abierto. Memorias del XVIII Congreso Internacional Anual de la SOMIM 19 al 21 de septiembre, salamanca, Guanajuato, México. 2012.

[4] Tie Duan, “Real-Time CNC Machine Tool Control with Linux” Senior Design Project at MUSE, American Society for Engineering Education, 2008.

[5] Tomislav Staroveški, “Implementation of a LINUX-BASED CNC Open Control System”, 12th International Scientific Conference on Production Engineering (CIM 2009), Biograd Croacia, 2009.

[6] Carrasco C. Diseño y construcción de un estetoscopio electrónico de bajo costo con filtrado de frecuencias para la detección de afecciones pulmonares y cardíacas. Tesis de Ingeniero Eléctrico y Electrónico. Universidad Nacional Autónoma de México. 2014.

[7] Ximenes M. Peron G. (2013). Implementação e avaliação de uma palma- lha de látex com sistema de sensores para investigação da distribuição de pressão plantar em indivíduos autistas. Dissertação de graduação em engenharia eletrônica, Universidade de Brasília, Campus Gama, DF, 83p. 2013.

[8] Stallman R, La definición de software libre, 1996. Disponible en <http://www.gnu.org/philosophy/free-sw.es.html>. Recuperado el 8/Septiembre/2014.

[9] Sánchez H. Construcción de una máquina fresadora de control numérico. Tesis profesional que para obtener el título de Ingeniero Mecánico. Facultad de Ingeniería. UNAM. 2011.

Diseño y Simulación de los Sistemas de Fundición, Instrumentación Electrónica y Diseño Mecánico, para un Prototipo de Maquina Fundidora de PET

Guillermo Córdova Morales, Elmar Montiel Jiménez y
Espinoza Hernández Andrés

Instituto Tecnológico Superior de Libres
Camino Real Esq. Calle Cuauhtémoc, Barrio de Tételes, Libres Puebla. C.P. 73780
gcmora@hotmail.com, pumandrin2@hotmail.com, andres.espinoza@live.com.

Abstract

Research shows that pollution levels are increasing day to day without showing reduction of plastic waste that damages the environment. It is necessary to provide technological systems to assist in the reduction of pollutants, achieving market place waste have already been used by the consumer, and this way keep energy required in developing new products and preserve material from nonrenewable resource.

The prototype called "PET foundry Machine" (polyethylene terephthalate), is based on three important phases of design proposals with the goal of produce synthetic fiber and offers the market a low cost product using it as filled for pillows or stuffed animals, contributing an environmental impact generous with a production of about 18 kg / hour.

The objective was to select the equipment and tools needed to satisfy with a control system the PET foundry process, and get the simulation of the important parts of the process that do the prototype, thereby obtaining a sophisticated model and approximate to the actual results before creating manufacturing costs of a real prototype.

Resumen

Investigaciones demuestran que los índices de contaminación aumentan día a día sin mostrar disminución de los desechos plásticos que deterioran el medio ambiente.

Es necesario aportar sistemas tecnológicos que colaboren en la reducción de contaminantes de este tipo, logrando colocar en el mercado los residuos que ya han sido utilizados por el consumidor, y de esta manera ahorrar energía requerida en la elaboración de productos nuevos y guardar materia prima obtenida de recursos no renovables.

El prototipo denominado Maquina Fundidora de PET (PoliEtileno Tereftalato), se basa en tres fases

importantes de propuestas de diseño con el objetivo de producir fibra sintética y ofrecer al mercado un producto de bajo costo utilizando como relleno para almohadas o peluches, aportando un impacto ambiental generoso con una producción aproximada de 18 kg/hora.

El objetivo fue seleccionar los equipos e instrumentos necesarios para cumplir con un sistema de control del proceso de fundición de PET y conseguir la simulación de las partes importantes del proceso que lleva a cabo el prototipo, obteniendo con ello un modelo sofisticado y mayormente acercado a los resultados reales antes de crear costos en fabricación de un prototipo real.

Palabras clave: Maquina fundidora, PET, extrusor, fibra sintética, instrumentación electrónica.

1. Introducción

El proceso para obtener fibra de PET comienza en la entrada de la materia reciclada, limpieza, trituración, precalentado, fundición y extrusión. El proceso realizado por la máquina para la fundición de plástico comparte créditos con los sistemas de extrusión que se usan actualmente en distintos ramos de la industria para dar forma a un material principalmente plásticos, cerámicas y también a algunos aceros.

Se propone un sistema con la particularidad de solo fundir y transportar el material fundido hacia un mecanismo de centrifugación con el objetivo de obtener finas hilaturas en forma de algodón, que posteriormente sean llevadas hasta las manos de un empacador por medio de un transportador de banda. Cada una de las partes de la máquina realiza una tarea importante, sin embargo la necesidad del control de proceso más significativa dentro de la misma, está en la fundición de pellets debido a que se requiere una intervención que sea capaz de mantener la temperatura deseada a lo largo de la fundición del

plástico llevando hasta el depósito centrifugador un material fluido, el cual pueda expedirse por las paredes del mismo.

El presente trabajo propone un sistema basado en tres componentes generales del diseño conceptual de una máquina, que conjuntamente pretenden ser los mecanismos correctos para obtener el resultado deseado, siendo estos los siguientes;

- Fundición de pellets.
- Centrifugación de plástico fundido.
- Acarreo de producto (fibra) hacia el empaque.

En la figura N° 1, se observa un cuadro general del proceso que describe de forma breve los pasos desde la entrada de material (pellets) hasta la obtención de fibra de PET.



Figura 1. Cuadro de proceso realizado por la máquina.

Debido a que no existen maquinas que realicen singularmente el proceso de obtención de fibras con PET, esta máquina es un prototipo que con ayuda de los softwares:

- *SolidWorks®* (Mecánico)
- *COMSOL Multiphysics®* (Térmico)
- *ISIS de Proteus®* (Instrumentación electrónica)

Con licencias de evaluación, se logran simulaciones optimizando un prototipo virtual sin crear costos por fabricación de un modelo existente intentando acercar los resultados requeridos.

2. Metodología

Se crea un sistema metodológico el cual se sigue hasta generar sistemas independientes para cada parte del prototipo, dando énfasis a las piezas que se cree serán las más importantes para poder lograr los resultados requeridos. En la figura N° 2 se muestra la sistemática que se sigue para conseguir un prototipo virtual, antes de generar costos por un modelo real.

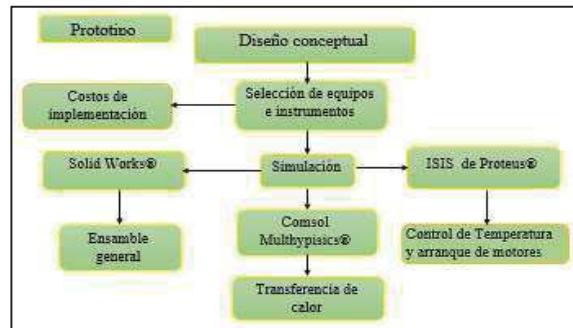


Figura 2. Metodología.

2.1. Concepto del diseño de la fundidora de PET

La máquina se está diseñando para procesar PET, el sistema de fundición, ingresara pellets con dimensiones de 7 x 5 mm, con un espesor promedio de 0.6 mm, estos deben estar libres de impurezas y humedad, en la figura N° 2 se observa una muestra de pellets cortados y completamente limpios.



Figura 3. Pellets previos a fundirse

La temperatura difiere en tres zonas a lo largo de la fundidora, por lo que se manejan sensores que emiten el valor del proceso a los controladores que se encargaran de corregir el valor deseado, activando o desactivando las resistencias correspondientes a la zona. Se propone un tornillo extrusor no mayor a 1½" con la finalidad de que el sistema no sea tan robusto.

Debe contar con un sensor en la tolva para mandar una señal de materia prima faltante o existente, finalmente el control tendrá la tarea del encendido y apagado de los motores correspondientes a cada parte de la máquina, en total tres motores.

2.2. Propuesta del sistema de fundición de PET

El diseño del prototipo se realiza con apoyo de *SolidWorks®*, tomando en cuenta los parámetros de diseño correspondientes al material que se requiere fundir y eligiendo materiales de gran entereza que puedan resistir las altas temperaturas que se manejan dentro del procedimiento.

El sistema es diseñado en forma seccionada, se ensambla con bridas logrando la fijación con tornillos con el fin de realizar un fácil mantenimiento posterior a ser utilizado.

El tornillo se diseña con una relación longitud: diámetro, de 24:1, así mismo la camisa o cañón de alojamiento se elige con una relación de diámetros de 2.66:1, quedando el diámetro de la camisa de 4".

La tolva de alimentación se diseña con la capacidad de suministrar por una hora material hacia el interior del sistema de fundición.

2.3. Calculo de la potencia requerida por el husillo

Se realizaron cálculos para obtener la potencia requerida por el husillo utilizando un análisis de la geometría que observamos en la figura 4.

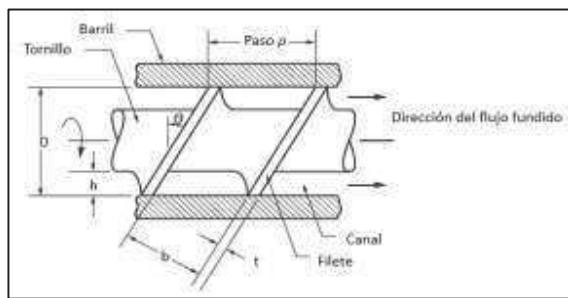


Figura 4. Geometría del tornillo

Se establecen parámetros iniciales con los cuales el diseño varía según las necesidades del sistema requerido, los valores de la tabla 1 son sustituidos en las ecuaciones que mencionan la potencia requerida.

Parámetros iniciales para la obtención de la potencia	
Diámetro husillo	D 1 1/2"
Profundidad del canal en la zona de dosificación	h 1/8"
Paso	p 1 1/2"
Ancho del filete	t 0.18"
Velocidad (RPS)	N 1.66 rps
Apertura del filete - cañón	δ .003"
Ángulo de hélice	ϕ 17.65°

Tabla 1. Parámetros iniciales

Se realiza la sustitución de valores en las ecuaciones que se obtienen del análisis de la geometría del tornillo, siendo estas las siguientes:

$$\text{Ancho de canal} \quad b = (p - t) \cos \Theta \quad (1)$$

$$\text{Largo de la hélice} \quad dz = \frac{p * \pi}{\cos \Theta} \quad (2)$$

$$\text{Velocidad longitudinal} \quad W = \pi * D * N * \cos \Theta \quad (3)$$

$$\text{Flujo de arrastre} \quad Q_d = \frac{W}{2} * b * h \quad (4)$$

Potencia del canal por vuelta

$$E_{\text{canal}} = \frac{\eta * W^2 * b * dz}{h} [4 * (1 + \tan^2 \Theta) - \frac{6 * Qd}{W * b * h}] \quad (5)$$

$$\text{Potencia del filete} \quad E_{\text{filete}} = \frac{\eta * W^2 * t * dz}{\delta * \cos \Theta} \quad (6)$$

Finalmente se suman las potencias y se obtiene la potencia total requerida por el husillo para su correcto funcionamiento.

$$\text{Potencia total} \quad E_{\text{total}} = E_{\text{canal}} + E_{\text{filete}} \quad (7)$$

2.4. Calculo de la potencia requerida por las resistencias de calentamiento

Se calcula la potencia requerida para calentar el sistema en un tiempo de 15 minutos llegando al punto de fusión establecido, se obtiene una gráfica de la potencia requerida al inicio (P_1) y la potencia requerida para mantener el ciclo de trabajo continuo (P_0), siendo la mayor de 11,604.08 Watts, con la cual se establece la potencia demandada.

En la figura 6, se muestran las dimensiones de la camisa con la cual se realiza el cálculo.

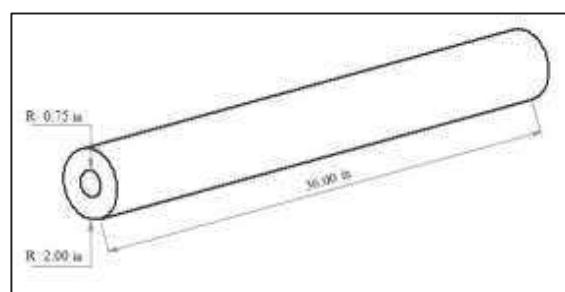


Figura 5. Dimensiones del cañón.

Las ecuaciones toman variables de las características de la camisa, tales como; el peso (w), el calor específico (Cp), el calor latente de fusión (Hf), el incremento de temperatura (ΔT), el tiempo de inicio (t_i), el tiempo de ciclo (t_c), las pérdidas de calor (Q_L), y un factor de seguridad regularmente

establecido del 10% al 30%. Las fórmulas utilizadas son:

$$\text{Potencia de inicio} \quad P_I = Q_A + Q_C + \frac{2}{3} Q_L + FS \quad (8)$$

$$\text{Potencia de ciclo} \quad P_O = Q_B + Q_D + Q_L + FS \quad (9)$$

Siendo:

Q_A = Potencia requerida para incrementar la temperatura del equipo al punto de operación en el tiempo deseado.

Q_B = Potencia requerida para mantener la temperatura durante el ciclo del proceso.

Q_C = Potencia requerida para fundir o vaporizar el material en el periodo de inicio.

Q_D = Potencia requerida para fundir o vaporizar el material durante el ciclo de trabajo.

Y finalmente las ecuaciones que referencian a las potencias, resultan de:

$$Q_A \text{ y } Q_B = \frac{w * Cp * \Delta T}{ti * 3.142} \quad (10)$$

$$Q_C \text{ y } Q_D = \frac{w * Hf}{tc * 3.142} \quad (11)$$

2.5. Propuesta del sistema de centrifugado del PET

El sistema de centrifugación se diseña con el apoyo de *SolidWorks*®, tomando en cuenta las dimensiones del sistema de fundición se diseña un deposito centrifugador que captara y expedirá por sus paredes el plástico fundido, se expone su manufactura con lámina de acero inoxidable calibre 14 perforada en las paredes de su circunferencia con pequeños agujeros no mayores en diámetro a 3 mm. Al depósito se le acopla la flecha de transmisión dando el giro al plato centrifugador conectado a un motor, el material de esta flecha es redondo de 1" AISI 1045.

La base del sistema se propone de un material PTR de 1 ½" x 1/8" de espesor, y finalmente las paredes que cubren el sistema de centrifugación (carcasa), se propone de un material blando, como lo es la lámina galvanizada en un calibre 18. Se modela la fijación de la flecha por medio de dos chumaceras de pared de 1", colocadas linealmente a la misma distancia partiendo de sus extremos las cuales se soportan con solera de ¼" x 1 ¼", figura 6.

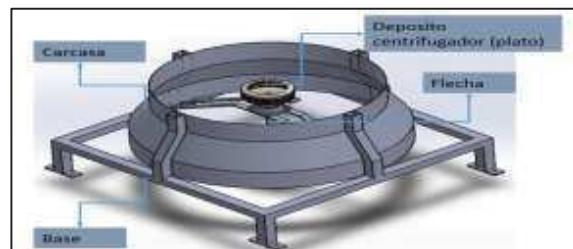


Figura 6. Propuesta de sistema de centrifugado

2.6. Propuesta del sistema de transporte del producto

Se diseñó un pequeño sistema de transporte del producto de material, el cual se encarga de llevar el producto hasta el empaque. El sistema aporta la estructura principal, cuya función es soportar todo el peso de los sistemas anteriores, los materiales que se manejan para el modelo es P.T.R. de 1/8" x 1 ½", el cual es un material que cumple con la resistencia para el sistema. Se colocan dos rodillos sencillos de 4" de diámetro, y se modelan 2 cintas de 14" de ancho para transportar la fibra. Los rodillos se fijan a 4 chumaceras de piso de 1" de diámetro, dos de ellas se modelan fijas y dos con la opción de moverse para realizar previos ajustes de tensión, figura 7.



Figura 7. Propuesta del sistema de transporte

2.7. Diseño del hardware de control de la instrumentación electrónica

Se efectuó un modelo de hardware electrónico en el software de simulación *Isis de PROTEUS*® (Figura 8), logrando un simulacro del control por medio de Micro Controladores. Este sistema es de On-Off tipo ventana, y es gobernado por una tarjeta arduino mega.

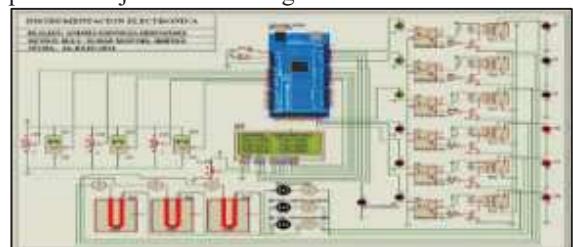


Figura 8. Hardware de control electrónico

Las etapas de este sistema tienen comienzo con el encendido del sistema general, seguidamente se encarga el usuario de establecer los rangos de temperatura que se requieren permanezca el sistema, el micro controlador se encargara de emitir instrucciones que previamente se realizan en un algoritmo de control para que finalmente los equipos finales de control realicen las tareas de encenderse o apagarse, según sean las condiciones requeridas.

2.8. Diseño del algoritmo de control ON-OFF

El algoritmo efectúa el control del proceso, manteniendo la temperatura deseada dentro del rango que previamente el usuario ajusta de forma manual mediante los potenciómetros, el estado de la misma de acuerdo al rango prefijado dependerá del comportamiento de los equipos finales de control, motores y resistencias. El código se realizó en el software de programación de arduino.

La figura 9, muestra el diagrama de flujo que obedecerá el hardware de control, en él se describe el inicio del proceso y las condicionantes que debe respetar cada una de las instrucciones que se le dan mediante el código de programación.

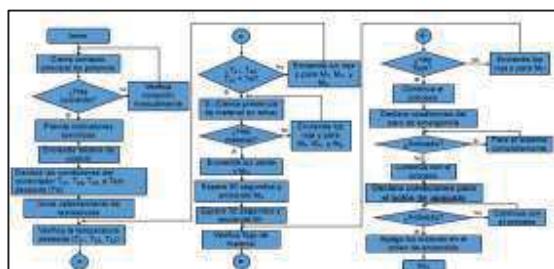


Figura 9. Diagrama de flujo implementado por el micro controlador

3. Resultados

Se logra la presentación de la propuesta de modelo completamente ensamblado, tal muestra la figura 10.



Figura 10. Ensamble general

Se corrobora la resistencia de los materiales propuestos, mediante estudios de esfuerzos máximos permisibles de las estructuras en *SolidWorks*[®]. Talmente muestra en la figura 11.

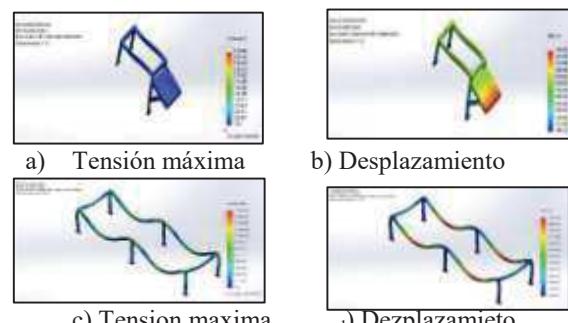


Figura 11. Estudio de tensiones y desplazamientos máximos en estructuras

Se confirma mediante una simulación en *COMSOL Multiphysics*[®], que la potencia obtenida de los cálculos es factible para cumplir con la fusión del polímero, realizando el análisis para el sistema de fundición en un tiempo de 15 minutos escalándolo a 5 minutos, como se muestra en la figura 12.

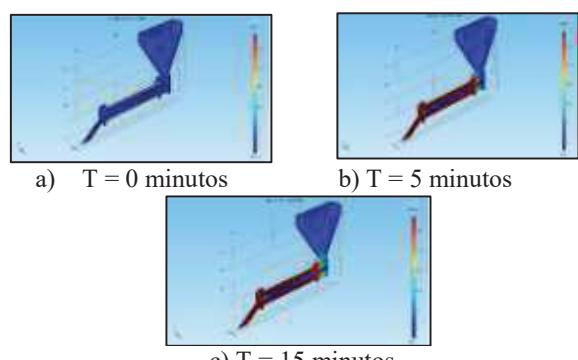


Figura 12. Simulación de temperatura en *COMSOL Multiphysics*[®].

Se comprueba que la resistencia del plato centrifugador es factible, simulándolo con una carga constante y un giro de 3600 R.P.M., en un estudio de esfuerzos máximos permisibles con *SolidWorks*[®], figura 13.

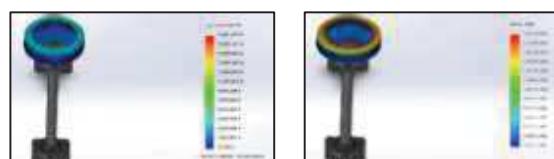


Figura 13. Estudio de tensión y deformación máxima del plato centrifugador.

Se garantiza el correcto funcionamiento del algoritmo de control, realizando simulaciones del hardware y software implementado por Arduino en *ISIS de Proteus*® observando el correcto trabajo de los instrumentos y equipos y se obtiene una estimación de costos (Tabla 2) para la implementación física de los equipos e instrumentos.

Descripción	Cant.	P.U.	Total
Motorreductor 5HP	1 Pza.	\$ 12,500.00	\$ 12,500.00
Motorreductor ¼ HP	1 Pza.	\$ 6,000.00	\$ 6,000.00
Motor ¾ HP	1 Pza.	\$ 3,800.00	\$ 3,800.00
Resistencia de banda	6 Pza.	\$ 1,500.00	\$ 9,000.00
Resistencia de nariz	1 Pza.	\$ 900.00	\$ 900.00
Controlador digital	3 Pza.	\$ 2,100.00	\$ 6,300.00
SSR trifásico	2 Pza.	\$ 1,800.00	\$ 3,600.00
SSR monofásico	1 Pza.	\$ 720.00	\$ 720.00
Termopar STD	3 Pza.	\$ 350.00	\$ 1,050.00
Cable de alta temp.	15 Mts.	\$ 46.00	\$ 690.00
Block de 2 polos	1 Pza.	\$ 30.00	\$ 30.00
Gabinete de control	1 Pza.	\$ 4,500.00	\$ 4,500.00
Juego de botoneras	1 Lote	\$ 2,800.00	\$ 2,800.00
Subtotal			\$ 51,890.00

Tabla 2. Costos por implementación física de instrumentación y equipos.

4. Conclusiones

La eficiencia del sistema de fundición se comprueba gracias a la simulación de transferencia de calor en el tiempo deseado, elaborada en el software de simulación multifísica *COMSOL Multiphysics*®, en el cual se ingresan parámetros de los materiales, tiempos y factores para los cuales calcula mediante el FEA (*Finite Element Method*), y arroja una interfaz virtual con respecto a un modelado, el comportamiento del mismo. El ensamblaje del prototipo general nos permite optimizar el equipo sin crear gastos en prototipos reales, al mismo tiempo se permiten realizar estudios de resistencia de los materiales ocupados, en las partes más importantes del sistema, dando con ello un voto de confianza en la correcta eficiencia al momento de fabricarlo. El sistema de instrumentación electrónica logra comprobar el resultado requerido en la simulación realizada en *ISIS de Proteus*®, manteniendo la temperatura en los rangos preestablecidos por el usuario, y controlando los equipos finales de control. Cabe mencionar que el sistema es tentativo para lograr un bajo costo en fabricación a comparación del uso un PLC comercial y controladores de alto costo.

5. Trabajos futuros

Hasta el momento este trabajo conforma un prototipo virtual que bien se puede seguir analizando para lograr optimizar a su máximo la eficacia del mismo, se puede recurrir al método experimental (Implementación) y en ello mejorar detalles que se puedan presentar. Se recomienda prestar atención al sistema de fundición, implementando algunas medidas de seguridad usuario-máquina.

6. Referencias

- [1] Creus A., *Instrumentación industrial*. Alfaomega-Marcombo, Barcelona, España. 6ta edición, 1998.
- [2] L. Mott R., *Diseño de elementos de máquinas*. Prentice Hall, 4ta edición, México, 2006.
- [3] P. Groover M., *Fundamentos de manufactura moderna*. McGraw-Hill, México, 2007.
- [4] D. H. Morton-Jones., *Procesamiento de plásticos*. Limusa, México, 2012.
- [5] Gómez S., *El gran libro de SolidWorks*®. Alfaomega-Marcombo, Barcelona, España, 2008.
- [6] Torrente O., *Arduino Curso práctico de formación*. Alfaomega, México D.F., 2013
- [7] Luna R., Moreno E., Ledesma R., “DISEÑO DE MAQUINA RECICLADORA DE PET”, Congreso Nacional de Tecnología Industrial, CONATI, 9, Guanajuato, México, 2009.
- [8] WATLOW® “THERMAL APPLICATIONS GUIDE [Online]”. Disponible en: <http://www.watlow.com/literature/catalogs.cfm>
- [9] COMSOL (1998–2013), Heat Transfer Module User’s Guide, Versión 4.4, disponible en la documentación del software.
- [10] COMSOL (1998–2013), COMSOL Multiphysics User’s Guide, Versión 4.4, disponible en la documentación del software.

Uso del Software Solidworks en el Diseño de Sistemas Mecatrónicos

Ruiz Domínguez Alejandra¹, Meléndez Campos Javier¹, Jiménez López Eusebio², Montoya Quezada Manuel Norberto³, García Velásquez Luis Andrés¹, Lucero Velásquez Baldomero⁴, Delfín Vázquez Juan José⁴ y Luna Sandoval Gabriel⁵

¹ Universidad La Salle Noroeste, alejandra_rui92@hotmail.com, jimelendezcampos@hotmail.com, lgarcia@ulsa-noroeste.edu.mx

² Universidad La Salle Noroeste – CIAMM-UTS-IIMM, ejimenezl@msn.com

³ Instituto Tecnológico de Sonora, norbert@hotmail.com

⁴ Instituto Tecnológico Superior de Cajeme, ijdel1704@yahoo.com.mx, blv72@hotmail.com

⁵ Universidad Estatal de Sonora, gabriel.luna@ues.mx

Resumen

El CAD (Computer Aided Design) es el uso de herramientas computacionales que ayudan al diseño de productos y máquinas, y en especial a la concepción y diseño de sistemas mecatrónicos, como es el caso de los robots industriales. El Solidworks es un software definido dentro de la filosofía CAD que por su versatilidad se usa para diversas aplicaciones en el campo de la mecatrónica. En este artículo se presenta una aplicación del Solidworks 2014 en el diseño y la fabricación de un robot cartesiano, y en la construcción de un simulador para iPad. Se discute la transferencia de información entre archivos generados en Solidworks 2014 y una máquina de impresión 3D. Se propone una metodología para migrar archivos Solidworks a OPEN GL ES 2.0. Se describe la importancia del modelado de sólidos y la generación de planos de fabricación para la fabricación del robot cartesiano.

Palabras clave: CAD, Solidworks 2014, Mecatrónica, Robótica, Manufactura Aditiva.

1. Introducción

El Solidworks es una herramienta computacional que asiste al diseño de productos y maquinaria. Hoy en día esta herramienta tiene diversas aplicaciones, en particular en el desarrollo de una gran diversidad de sistemas mecatrónicos, como por ejemplo en el diseño y control de prototipos de robots [1]. Los softwares concebidos bajo la filosofía CAD, para el caso de la enseñanza de la mecatrónica, no deben limitarse al dibujado convencional, sino que deben

estudiarse en el contexto de la integración y sinergia. La exportación de archivos de partes y componentes hacia otros softwares es una actividad que debe ser ampliamente explotada, pues diversos análisis y procesos son automatizados. Así, existe integraciones como el CAD/CAM o el CAD/CAE que potencializan el desarrollo de productos.

El Solidworks se ha usado en combinación con LABVIEW para el desarrollo de simuladores de robots cartesianos [2]. En [3] se presenta el diseño y la construcción de un robot hexápolo usando Solidworks e impresión 3D. En este artículo se presenta la aplicación del Solidworks 2014 en el desarrollo de un robot cartesiano. Se describen y proponen cuatro metodologías que puede auxiliar a los alumnos y profesores a utilizar el forma ordenada y sistemática dicho software en el dibujado de piezas y en la transferencia de archivos a sistemas de manufactura aditiva, así como en la graficación y generación de simuladores operables desde dispositivos móviles. Este desarrollo forma parte de la RED ALFA y del proyecto PROMEP UTSSON-CA-4.

2. Aplicaciones del Solidworks en el dibujado de un robot didáctico

El enfoque de las aplicaciones de los softwares relacionados con las filosofías CAD difiere si se trata de diseñar sistemas mecatrónicos, pues la estrategia del uso del software no se limita al dibujo convencional, sino que deben tomarse en cuenta aspectos de transferencia de información a otros paquetes. Además, el tipo de problema a resolver o las

necesidades de los clientes indican el camino a seguir en el uso de los paquetes de dibujo. Para el caso del prototipo del robot cartesiano, éste fue solicitado por una empresa y los requerimientos iniciales pedidos fueron:

- 1) Diseño en CAD de las partes y ensamble del robot.
- 2) Desarrollo de un simulador del robot integrando trayectorias rectilíneas con perfiles de trayectoria trapezoidales.
- 3) Uso de técnicas de manufactura aditiva para la fabricación del robot.
- 4) Debe incluir motores paso a paso.
- 5) El control y manejo del robot debe realizarse por medio de un dispositivo móvil (Tablet o celular).

Después de haber analizado los requerimientos del cliente y evaluado los conocimientos necesarios para desarrollar el proyecto, se integró un equipo de trabajo conformado por alumnos de Diseño Industrial de quinto semestre y Mecatrónica de octavo semestre, los cuales cursaban las materias de Mecanismos y Tecnología de Materiales, y Seminario de Investigación, respectivamente. Los alumnos propusieron la siguiente utilización del software de dibujo:

- 1) Utilizar el software Solidworks 2014 para el dibujo en 3D (Modelado de sólidos) y en 2D (planos de fabricación).
- 2) Transferir los archivos de dibujo de piezas a sistemas de manufactura aditiva.
- 3) Transferir los archivos de Solidworks a un software de amplio poder gráfico para desarrollar un simulador que pueda ser operado desde una Ipad.

Para el caso del dibujado del robot cartesiano se desarrollaron todos sus planos de fabricación y un modelo en sólidos (ver Figuras 1 y 2).

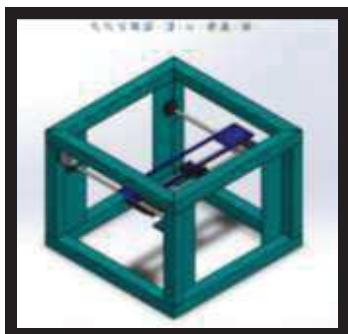


Fig. 1. Modelo en sólidos del robot.

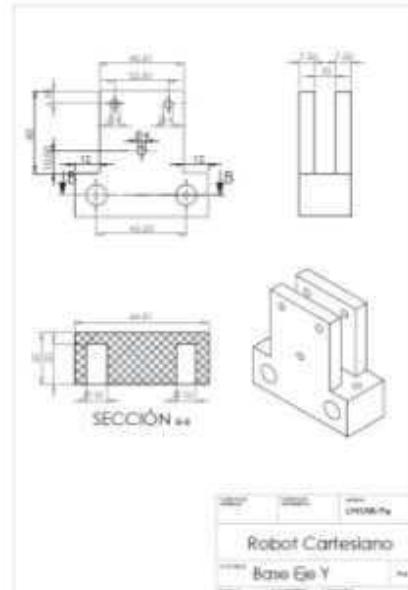


Fig. 2. Plano de fabricación en Solidworks.

Con el propósito didáctico de brindar un camino para el desarrollo del dibujo en Solidworks, a continuación se proponen dos metodologías básicas.

2.1. Metodología para creación de un modelo CAD en Solidworks

1) Crear archivo de Parte. En la opción Nuevo del menú Archivo, seleccionar “Parte” para comenzar a crear un modelo 3D.

2) Verificar unidades. En la esquina inferior derecha, se tiene la posibilidad de seleccionar el sistema de unidades que se desee manejar, ya sea MKS, MMGS, IPS, o ya sea un sistema personalizable.

3) Crear croquis. Seleccionar la opción Croquis y crearlo seleccionando el plano de mayor conveniencia, ya sea Alzado, Planta o Vista Lateral.

4) Realizar dibujo 2D. Una vez que se tiene el croquis activo, es posible seleccionar diferentes instrucciones de geometrías básicas, como son Línea, Rectángulo, Círculo, entre otros. Se selecciona la opción que se desea usar, y se comienza a dibujar en el plano.

5) Acotar y restringir. Es posible dar medidas específicas a los dibujos, esto se logra a través de la opción Cota Inteligente. De igual manera se pueden agregar restricciones para el dibujo creado, como Paralelo, Perpendicular,

Coincidente, Vertical, Horizontal, por mencionar algunos.

6) Operaciones. Una vez que se tiene un dibujo listo, se procede a realizar una operación con él. La opción más básica es Extruir Saliente, que permite darle un espesor específico al dibujo 2D creado anteriormente. De igual forma existen opciones como Revolución de Saliente que crea sólidos circulares, o Extruir Corte que sustrae material de sólidos ya creados.

7) Guardar. Una vez terminado el modelado de un sólido, este será guardado en formato .SLDPRT de manera predeterminada, formato que sólo es compatible con Solidworks, pero también es posible guardarlo en diversos formatos como son .IGS, .STL, o .STEP, entre otros.

2.2. Metodología para creación de planos de fabricación a partir de un modelo CAD en Solidworks

1) Abrir archivo CAD. Abrir en Solidworks el modelo CAD a partir del cual se desea generar el plano de fabricación

2) Crear archivo de dibujo. Seleccionar la opción “Crear dibujo desde pieza”.

3) Seleccionar formato y tamaño de hoja. A continuación el software permite seleccionar el formato de la plantilla que se desea utilizar para crear el modelo. De manera predeterminada maneja los formatos estándar ISO (A0, A1, A2, A3, A4), de igual manera se pueden crear plantillas personalizadas según los requerimientos necesarios.

4) Insertar vistas. Una vez seleccionado el formato de hoja, se desplegará un menú del lado derecho, con las diferentes vistas del modelo creado. La vista que se deseé insertar se selecciona y se arrastra al centro de la hoja. Una vez insertada la primera vista se pueden insertar rápidamente vistas proyectadas del mismo.

5) Insertar anotaciones. Es posible insertar diferentes anotaciones a las vistas creadas, ya sea Cotas, Acabado superficial, Indicaciones de soldadura, Tolerancias geométricas, Anotaciones de taladro, entre otras.

6) Personalizar configuraciones. En el menú Opciones es posible seleccionar las unidades de las cotas mostradas; de igual manera es posible configurar los decimales activos.

7) Guardar. Una vez terminado el plano, éste es guardado en formato .SLDDRW de manera predeterminada, formato que sólo es compatible con Solidworks, pero también es posible guardarlo

en diversos formatos como son .DWG o .PDF, por mencionar algunos.

3. El Solidworks y la manufactura aditiva

La concepción de la fabricación de un modelo parte por analizar sus características y atributos de diseño, ergonomía, costos, etc. Hasta la fecha se hacia el diseño de un molde en un software CAD para ser llevado a una máquina que podía ser una inyectora y ver los resultados de ese modelo. Los costos de diseño y desarrollo del producto se incrementaban drásticamente al ver que un producto no cumplía con las especificaciones por lo que había que rediseñar el molde, fabricarlo y probar nuevamente el producto. Todos estos costos que son cargados al costo del producto terminado pueden ser reducidos considerablemente si antes se realiza el diseño de la pieza a nivel prototipo y después se genera el molde de ese producto para su fabricación en serie. Esta ventaja la ofrece la nueva tecnología que realiza una impresión tridimensional llamada impresora 3D. El diseño de los prototipos puede ser realizado en cualquier software CAD. La impresora es compatible con todos los software 3D del mercado que tengan la opción de exportar diseños con extensión STL (Stereo Lithography) o Lenguaje Estándar de Triangulación como Solidworks, Inventor, AutoCAD, IronCAD, etc. [4].

Para el caso del robot motivo de estudio y bajo el requerimiento de la empresa de que se usara impresión 3D para la fabricación de algunas piezas del robot, se procedió a generar los archivos con extensión .STL en Solidworks y a seleccionar una máquina de impresión 3D modelo AW3D V5. Una de las piezas fabricadas con esta tecnología se muestra en la Figura 3.



Fig. 3. Pieza fabricada en impresión 3D.

A continuación se presenta una metodología para la impresión 3D con archivos de Solidworks.

3.1 Metodología de importación de un modelo CAD para impresión 3D

1) Realizar modelado CAD. El diseño puede ser realizado en cualquier software tipo CAD, como lo son: Solidworks, Inventor, AutoCAD, por mencionar algunos. Para el caso del robot se utilizó el software Solidworks 2014 para la realización de los modelos sólidos, el cual crea los archivos en extensión .SLDPRT.

2) Convertir el modelo CAD a formato STL. Una vez que se tiene el diseño listo, en la plataforma de Solidworks se selecciona la opción “Guardar como...” y se guarda el archivo en formato .STL. Su nombre proviene de las siglas en inglés *Standard Tessellation Language*. Los archivos .STL representan superficies geométricas de un objeto tridimensional a base de triángulos que dan forma a la figura, sin embargo no se transfieren datos como texturas o colores.

3) Ejecutar software de codificación. En el software Slic3r, que se utiliza para generar el código de impresión, en la opción “Abrir” se selecciona el archivo .STL que se desea imprimir.

4) Crear archivo .GCODE. Se crea un archivo de código G con extensión .GCODE. Éste es el lenguaje de programación más utilizado en control numérico (CNC), que consiste en una serie de instrucciones, como coordenadas, trayectorias y velocidades, que la impresora va a seguir.

5) Ejecutar software de impresión. El software de impresión seleccionado fue Pronterface.

6) Verificar conexión del software con la impresora. Para que esta conexión sea exitosa, es necesario que el firmware de la impresora utilizada sea compatible con Pronterface.

7) Configurar condiciones de impresión. Una vez realizada la conexión, se abre el archivo .GCODE generado con anterioridad. Dentro de la interfaz del software se pueden seleccionar opciones como “Mover”, “Rotar”, “Escalar”, “Densidad” y “Temperatura”, para configurar el sólido a imprimirse y las condiciones en las que se realizará.

8) Imprimir. Una vez configuradas todas las condiciones, se verifica la cantidad de material necesario y se procede a imprimir. La impresora se precalentará hasta la temperatura configurada, y comenzará a realizar la impresión siguiendo las trayectorias generadas en el GCODE. En el

software se mostrará el porcentaje del proceso de impresión.

4. El Solidworks y el Open GL

Otras de las aplicaciones que tiene el Solidworks es en la simulación. Para el caso del robot en estudio las tecnologías usadas para el diseño del simulador fueron: archivos de MatLAB (funciones y cálculo de la cinemática del robot), SolidWorks (migración de partes gráficas del robot), Xcode, Migrador SolidWorks-OBJ-VertexData y Open GL ES 2.0. La Figura A.1. muestra el simulador desarrollado [5]. A continuación se presenta una metodología que sirve para interrelacionar el Solidworks y el Open GL.

4.1 Metodología de migración de un modelo Solidworks a Open GL ES 2.0

1) Descargar e instalar el Macro de Solidworks llamado OBJ Exporter (<https://forum.solidworks.com/thread/54270>).

Dicho complemento convierte la información del archivo de Solidworks a un formato llamado OBJ, el cual se puede entender como un tipo de archivo que guarda información básica de figuras hechas a base de primitivas, incluyendo de esta manera vértices, caras, colores, materiales entre otros datos. OBJ Exporter secciona las figuras dibujadas en Solidworks de tal manera que las traduce a una serie de triángulos, guardando la información en un archivo .OBJ y en una serie de puntos correspondientes a los vértices, así como los colores utilizados en el modelo Solidworks.

2) Descargar y copiar a una ubicación accesible el transformador blender2opengl. (<http://www.raywenderlich.com/48293/how-to-export-blender-models-to-opengl-es-part-1>). Este sencillo programa de consola toma como argumento un archivo OBJ y traslada toda la información a una estructura de arreglo en lenguaje C.

3) Realizar el modelado de todas las piezas en Solidworks.

4) Realizar un ensamblaje general de la máquina en Solidworks para comprobar y validar el diseño.

5) Proporcionar un seccionado del modelo ensamblado, de tal manera que las piezas a mover queden agrupadas dependiendo del tipo de movimiento al que se someterán. Este punto es de vital importancia, ya que de aquí se determinarán los movimientos que realizan las diferentes partes

de la máquina en el simulador. Es decir, si un grupo de piezas forman un pedazo de la máquina que se traslada o rota, estas deberán ir unidas y referenciadas entre sí en un sólo ensamblaje.

6) Realizar los ensambles por separado de los grupos de partes propuestos en el punto 3. Al final de este paso, se debe contar con una serie de ensambles de Solidworks los cuales incluyan cada grupo de partes móviles en la máquina.

7) En cada ensamble realizado, ejecutar el Macro OBJ Exporter, desde la barra de herramientas de Solidworks. Al presionar el botón aparece una ventana donde se puede modificar algunos parámetros. Todas las opciones pueden ser dejadas como vienen por default. Seleccionar de preferencia una ubicación (carpeta) determinada para todos los ensambles a guardar. El archivo se guardará en dicha dirección con el nombre que el archivo de Solidworks pero en una diferente extensión.

8) Es fácil observar la información que se guardó en este archivo abriéndolo mediante un editor de texto básico. Se encontrará con una serie de líneas descriptas a continuación:

- Una “v” seguida por tres números (v -0.16 0.24 0.80). Esto indica la posición de un vértice en x=-0.16 y=0.24 z=0.80.
- Una “vn” seguida por tres números (vn -0.20 0.28 0.10) indica las normales, es decir un vector que apunta hacia fuera en un vértice. Esta información es usada principalmente para cuestiones de iluminación en animaciones.
- Una “vt” seguida por dos puntos, los cuales indican la coordenada de alguna textura desde la cual el vértice se emparejará para tomar parte de la textura.
- Una “f” seguida por tres grupos de números (f 117/117/117 146/146/146 150/150/150) que agrupa vértices, normales y vértices de textura para formar una cara de la figura representada por un triángulo.

9) Ejecutar el programa OBJ to VertexData. (Repetir este paso por cada ensamble exportado en formato OBJ). Este programa lee el archivo OBJ producto del paso 7 y separa la información de tal manera que la va guardando en arreglos de lenguaje C, generando un par de archivos con extensión .h y .c, donde se pueden observar arreglos con los datos de vértices ordenados. No importa si son millones de vértices a guardar, mientras el archivo OBJ cuente con la estructura estándar antes descrita, todos los vértices serán guardados en los arreglos.

10) Importar los archivos .h y .c al proyecto donde se vaya a graficar la figura.

11) Configurar la ventana o en su defecto la vista de OpenGL ES 2.0 donde se va a renderizar el modelo (Depende de cada entorno de programación que tan fácil o difícil se hace el proceso para el desarrollador este paso).

12) Importar y referenciar el modelo (incluir el archivo .h) de datos basados en arreglos creados en el paso 9 (incluir cada .h generado, para poder graficar el ensamble completo), para usar la información y generar así todos los triángulos necesarios para formar la figura.

13) Realizar las transformaciones (traslado, rotado, escalado) necesarias para cada pedazo del ensamble. De tal manera que el modelo se mueva como se desea.

Finalmente, la figura A.1 descrita en el apéndice A, muestra el Solidworks y sus interrelaciones con otras tecnologías.

Conclusiones

En este artículo se ha utilizado el software Solidworks 2014 para el diseño, la fabricación y la simulación de un sistema mecatrónico. Las principales conclusiones se resumen en los puntos siguientes:

- El Solidworks es una herramienta concebida dentro de la filosofía CAD que auxilia de gran manera al desarrollo de prototipos. Para el caso del robot cartesiano, fue posible utilizar los archivos generados en Solidwork para dibujar el prototipo, para la manufactura convencional (interpretación de planos de fabricación), para la fabricación de piezas por medio de la técnica de manufactura aditiva (impresión 3D) y para la construcción de un simulador en combinación con Open GL. Toda esta versatilidad de aplicaciones hace posible afirmar que el Solidworks es una herramienta concebida dentro del concepto de alta integración y, por lo tanto, forma parte de la gama de tecnologías con las cuales se conciben los productos y sistemas mecatrónicos.

- Las metodologías descritas y propuestas en este trabajo, relacionadas con el dibujado y las interrelaciones entre el Solidworks y la manufactura aditiva, y entre el Solidworks y el Open GL, ayudarán a los alumnos y profesores a realizar aplicaciones con más facilidad y rapidez, a pesar de que sólo son propuestas y que no son únicas (debido que cada diseñador tiene una lógica propia para manejar el software).

- Debido a la enorme importancia que tienen los softwares de dibujado, como es el caso del Solidworks, para el desarrollo de productos y

sistemas mecatrónicos, es necesario que su enseñanza no sólo se oriente al dibujado, si no que se exploren todas las interfaces que puedan existir entre las diversas tecnologías que asisten a la mecatrónica, tales como, software para la manufactura por CNC, impresión 3D, gráficos en Open GL, software que asisten a la ingeniería (Análisis por elemento finito y análisis de mecanismos), software de control como es el caso de LABVIEW y softwares de cálculo formal, entre otras tecnologías.

Agradecimientos

Los autores de este trabajo agradecen a las Universidades y Empresas que integran la RED Interinstitucional ALFA, a la RED Pyme del Noroeste y a la empresa Spin – OFF Innovación en Ingeniería de Manufactura y Mantenimiento S. de R.L. MI, al Cuerpo Académico CIAAM de la UTS, al Programa del Mejoramiento del Profesorado (PROMEP) bajo el proyecto Desarrollo de un Prototipo de Robot Industrial para Aplicaciones en Celdas de Producción Automatizadas Clave UTSSON-CA-4, por el apoyo brindado para la realización de este trabajo.

Referencias

- [1] Carro J., Flores I., Flores F. Simulación y Control de un Sistema Mecatrónico Aplicando Diseño Asistido por Computadora. *La Mecatrónica en México*, Vol. 2, No. 3, páginas 90 - 98, Septiembre 2013. Disponible en línea en www.mecamex.net/revistas/LMEM
- [2] Sobarso J. Modelo y simulación de las trayectorias de un robot cartesiano usando la integración LabVIEW-SolidWorks para aplicaciones industriales. Tesis de Maestría en Ciencias de la Ingeniería Mecatrónica. Instituto Tecnológico Superior de Cajeme. 2012.
- [3] Eugenio López F, Betancourt J., Martínez H. Diseño y construcción de un robot hexápedo. Memorias del XVII congreso internacional anual de la SOMIM. 21 al 23 de Septiembre, San Luis Potosí, México. 2011.
- [4] Velásquez J. Tecnología para el prototipado rápido: Impresoras 3D. PARADIGMAS 3 (1) 203-208. ISSN: 2220-2056. 2011.
- [5] Meléndez J., Jiménez E., Ruiz A., Urbalejo A., Reyes L., Luna N., Vázquez I. Desarrollo de un simulador de un robot cartesiano en dispositivos

móviles iOS utilizado Solidworks y Open GL ES 2.0. *13º Congreso Nacional de Mecatrónica*. Octubre 30 a Noviembre 1. Querétaro, Qro. (Por presentarse). 2014.

Apéndice A



Fig. A.1 El Solidworks y sus interrelaciones en el desarrollo del robot.

Control por Dinámica Inversa de un Manipulador Paralelo en el Espacio Articular

Alvarado-Juárez Diego Armando, Villarreal-Cervantes Miguel Gabriel
y Sepúlveda-Cervantes Gabriel

Centro de Innovación y Desarrollo Tecnológico en Cómputo - IPN.

Resumen

En este trabajo se presenta la implementación experimental del control por dinámica inversa en el espacio articular de un robot manipulador paralelo de 3 grados de libertad para el problema de seguimiento. El sistema de control requiere la dinámica del sistema por lo que se presenta el modelo dinámico así como los parámetros cinemáticos y dinámicos reales del robot.

Palabras clave: Robot Paralelo, Control, Dinámica Inversa.

1. Introducción

Un robot es un manipulador multifuncional reprogramable diseñado para mover piezas, herramientas, materiales o dispositivos especiales mediante movimientos programados para llevar a cabo diferentes tareas [1]. La manera en la que se programan dichas tareas para que el robot pueda realizarlas depende directamente del sistema de control que se implemente. Existen diferentes tipos de controladores diseñados para cumplir características específicas dependiendo de la tarea que se quiera realizar. Dentro de los diferentes tipos de controladores existen los denominados “libres de modelo”, lo cual se refiere a que no se necesita el modelo dinámico del sistema para poder implementar el sistema de control. En [2] se presenta un control por modos deslizantes para un robot manipulador con carga variable utilizando el modelo dinámico de los motores, pero no del sistema completo. En [3] se hace el control de posición en el espacio de operación de un robot manipulador bajo el esquema de un controlador PID y se hace hincapié en la ausencia del modelo ya que facilita su implementación.

Por otro lado, se establecen los controladores denominados “basados en el modelo”, los cuales, dependen del modelo dinámico obtenido del sistema

para proponer la estrategia de control. Es bien sabido que los controladores basados en el modelo tienen en general un mejor desempeño que los denominados “libres de modelo”, debido a que compensan adecuadamente las no linealidades existentes en el sistema físico. Sin embargo, las principales desventajas de utilizar este tipo de controladores es que se requiere de la dinámica que describe el sistema real y de la identificación de los parámetros del sistema, los cuales no es una tarea sencilla, en la mayoría de los casos. Existen en la literatura estrategias que compensan las incertidumbres en el sistema, pero ese enfoque no se aborda en el artículo. En [4] se realiza el análisis de estabilidad de un control de fuerza híbrido, un control de rigidez y un control de impedancia para el problema de control de fuerza en manipuladores. En [5] se obtiene el modelo dinámico integrado de un manipulador con una plataforma móvil con el propósito de obtener una estrategia de control basado en el modelo que compensa los efectos dinámicos relacionados entre ambos sistemas. Se verifica la importancia de acoplar dinámicamente ambos sistemas en el diseño del controlador.

Debido a la importancia en el control basado en el modelo para sistemas mecatrónicos, en este trabajo se presenta la implementación de un control basado en el modelo llamado control por dinámica inversa para un prototipo de robot paralelo de 3 g.d.l. que se encuentra en las instalaciones del Centro de Innovación y Desarrollo Tecnológico en Cómputo del Instituto Politécnico Nacional (CIDEDEC-IPN). Se muestra la efectividad del sistema de control con base en resultados experimentales. Con los resultados presentados en este trabajo, se pretende que en futuras aplicaciones e investigaciones se pueda explorar diversas estrategias de control basado en el modelo.

2. Modelo dinámico y cinemático del robot manipulador paralelo.

El diagrama esquemático del robot paralelo al cual se le implementa la ley de control, se muestra en la Fig. 1a, donde l_i , m_i , I_{ci} , I_i son longitud, masa, longitud del centro de masa e inercia del i-ésimo eslabón, q_i es la posición angular del i-ésimo eslabón, τ_i es el par aplicado para mover la i-ésima articulación. En la Fig. 1b se muestra una foto del sistema real que se encuentra en las instalaciones del CIDETEC-IPN.

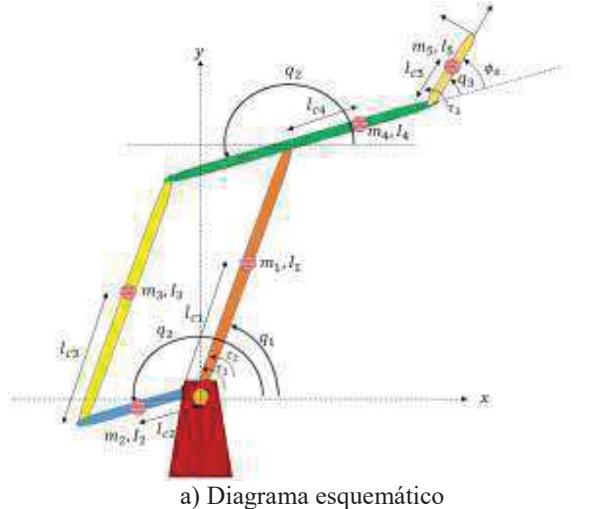


Figura 1. Manipulador Paralelo de 3 G.D.L.

En las siguientes subsecciones se muestra el modelo dinámico del robot así como su cinemática y Jacobiano que servirán para mapear la posición, velocidad y aceleración de la trayectoria deseada en el espacio Cartesiano al espacio articular.

2.1 Modelo dinámico.

La dinámica del robot se centra en el estudio de las fuerzas requeridas para producir movimiento en un sistema [6], es decir, la manera en la que se mueve un robot debido a los momentos de torsión aplicados por los actuadores, o a fuerzas externas aplicadas al robot.

Se utiliza una formulación basada en las ecuaciones de movimiento de Euler – Lagrange [6] para obtener el modelo dinámico del robot, por lo que se define el Lagrangiano $\mathcal{L}(q, \dot{q})$ como la diferencia entre la energía cinética $K(q, \dot{q})$ y la energía potencial $U(q)$ del robot manipulador como se muestra en (1), donde $q = [q_1, q_2, q_3]^T$, $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$, V_i y W_i es la velocidad lineal y angular del i-ésimo eslabón, g es la fuerza de gravedad que actúa sobre cada eslabón, y h_i es la posición de ese eslabón respecto a alguna referencia.

$$\mathcal{L}(q, \dot{q}) = \sum_{i=1}^3 K_i(q, \dot{q}) - \sum_{i=1}^3 P_i(q) \quad (1)$$

$$K_i = (1/2)m_i v_i^2 + (1/2)w_i^2 \quad (2)$$

$$P_i = m_i g h_i \quad (3)$$

Quedando definidas la energía cinética y potencial de cada eslabón como:

$$\begin{aligned} K_1 &= \frac{\dot{q}_1^2(m_1 l_{c1}^2 + I_{z1})}{2} \\ K_2 &= \frac{\dot{q}_2^2(m_2 l_{c2}^2 + I_{z2})}{2} \\ K_3 &= \frac{m_3 l_{c2}^2 \dot{q}_2^2}{2} - m_3 \cos(q_1 - q_2) l_2 l_{c3} \dot{q}_1 \dot{q}_2 + \frac{m_3 l_{c3}^2 \dot{q}_1^2}{2} + \frac{I_{z3} \dot{q}_1^2}{2} \\ K_4 &= \frac{m_4 l_{c1}^2 \dot{q}_1^2}{2} - m_4 \cos(q_1 - q_2) l_1 l_{c4} \dot{q}_1 \dot{q}_2 + \frac{m_4 l_{c4}^2 \dot{q}_2^2}{2} + \frac{I_{z4} \dot{q}_2^2}{2} \\ K_5 &= m_5 \left[\frac{l_1^2 \dot{q}_1^2}{2} + \frac{l_4^2 \dot{q}_2^2}{2} + \frac{l_{c5}^2 (\dot{q}_2 + \dot{q}_3)}{2} + l_1 l_4 \dot{q}_1 \dot{q}_2 \cos(q_1 - q_2) \right. \\ &\quad \left. + l_1 l_{c5} \dot{q}_1 \dot{q}_3 \cos(q_1 - q_2 - q_3) \right. \\ &\quad \left. + l_4 l_{c5} \dot{q}_2 \cos(q_3) (\dot{q}_2 + \dot{q}_3) \right] \\ &\quad + I_{z5} \left(\frac{\dot{q}_2}{2} + \frac{\dot{q}_3}{2} \right) (\dot{q}_2 + \dot{q}_3) \end{aligned}$$

$$P_1 = m_1 g (l_{c1} \sin(q_1))$$

$$P_2 = m_2 g (l_{c2} \sin(q_2))$$

$$P_3 = m_3 g (l_2 \sin(q_2) + l_{c3} \sin(q_1))$$

$$P_4 = m_4 g (l_1 \sin(q_1) + l_{c4} \sin(q_2))$$

$$P_5 = m_5 g (l_1 \sin(q_1) + l_4 \sin(q_2) + l_{c5} \sin(q_2 + q_3))$$

Por lo tanto las ecuaciones de movimiento de Euler – Lagrange se obtienen de (4), donde $\tau = [\tau_1, \tau_2, \tau_3]^T$.

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right] - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \quad (4)$$

De (4) se puede obtener el modelo dinámico del robot en su forma cerrada expresada en (5), donde $\tau = [\tau_1 \ \tau_2 \ \tau_3]^T$, $\ddot{q} = [\ddot{q}_1 \ \ddot{q}_2 \ \ddot{q}_3]^T$, $\dot{q} = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3]^T$, M es la matriz de inercia simétrica, definida positiva con elementos $M_{ij} \forall i, j = 1, 2, 3$, C es la matriz de fuerzas centrifugas y de coriolis con elementos $C_{ij} \forall i, j = 1, 2, 3$, y G es el vector de gravedad definido como $G = [G_1 \ G_2 \ G_3]^T$.

$$\tau_m = M\ddot{q} + C\dot{q} + G \quad (5)$$

$$\begin{aligned}
 M_{11} &= (l_{z1} + l_{z3} + l_1^2 m_4 + l_1^2 m_5 + l_{c1}^2 m_1 + l_{c3}^2 m_3) \\
 M_{12} &= (l_2 l_{c3} m_3 \cos(q_1 - q_2) - l_1 l_4 m_5 \cos(q_1 - q_2) \\
 &\quad - l_1 l_{c5} m_5 \cos(q_1 - q_2 - q_3) \\
 &\quad - l_1 l_{c4} m_4 \cos(q_1 - q_2)) \\
 M_{13} &= (-l_1 l_{c5} m_5 \cos(q_1 - q_2 - q_3)) \\
 M_{21} &= (l_2 l_{c3} m_3 \cos(q_1 - q_2) - l_1 l_4 m_5 \cos(q_1 - q_2) \\
 &\quad - l_1 l_{c5} m_5 \cos(q_1 - q_2 - q_3) \\
 &\quad - l_1 l_{c4} m_4 \cos(q_1 - q_2)) \\
 M_{22} &= (m_3 l_2^2 + m_5 l_2^2 + 2m_4 \cos(q_3) l_4 l_{c5} + m_2 l_{c2}^2 + m_4 l_{c4}^2 \\
 &\quad + m_5 l_{c5}^2 + l_{z2} + l_{z4} + l_{z5}) \\
 M_{23} &= (m_5 l_{c5}^2 + l_4 m_5 \cos(q_3) l_{c5} + l_{z5}) \\
 M_{31} &= (-l_1 l_{c5} m_5 \cos(q_1 - q_2 - q_3)) \\
 M_{32} &= (m_5 l_{c5}^2 + l_4 m_5 \cos(q_3) l_{c5} + l_{z5}) \\
 M_{33} &= (m_5 l_{c5}^2 + l_{z5}) \\
 C_{11} &= 0 \\
 C_{12} &= (l_2 l_{c3} m_3 \sin(q_1 - q_2) - l_1 l_4 m_5 \sin(q_1 - q_2) \\
 &\quad - l_1 l_{c5} m_5 \sin(q_1 - q_2 - q_3) \\
 &\quad - l_1 l_{c4} m_4 \sin(q_1 - q_2)) \dot{q}_2 \\
 &\quad + (-2l_1 l_{c5} m_5 \dot{q}_3 \sin(q_1 - q_2 - q_3)) \\
 C_{13} &= (-l_1 l_{c5} m_5 \sin(q_1 - q_2 - q_3)) \dot{q}_3 \\
 C_{21} &= (l_1 l_{c5} m_5 \sin(q_1 - q_2 - q_3) + l_1 l_4 m_5 \sin(q_1 - q_2) \\
 &\quad - l_2 l_{c3} m_3 \sin(q_1 - q_2) \\
 &\quad + l_1 l_{c4} m_4 \sin(q_1 - q_2)) \dot{q}_1 \\
 C_{22} &= (-2l_4 l_{c5} m_5 \dot{q}_3 \sin(q_3)) \\
 C_{23} &= (-l_4 l_{c5} m_5 \sin(q_3)) \ddot{q}_3 \\
 C_{31} &= (l_1 l_{c5} m_5 \sin(q_1 - q_2 - q_3)) \dot{q}_1 \\
 C_{32} &= (l_4 l_{c5} m_5 \sin(q_3)) \dot{q}_2 \\
 C_{33} &= 0 \\
 G_1 &= g \cos(q_1) (l_1 m_4 + l_1 m_5 + l_{c1} m_1 + l_{c3} m_3) \\
 G_2 &= -g (l_4 m_5 \cos(q_2) - l_2 m_3 \cos(q_2) - l_{c2} m_2 \cos(q_2) \\
 &\quad + l_{c4} m_4 \cos(q_2) + l_{c5} m_5 \cos(q_2 + q_3)) \\
 G_3 &= -g l_{c5} m_5 \cos(q_2 + q_3)
 \end{aligned}$$

2.2 Modelo cinemático.

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función de tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares [7], sin tomar en cuenta las fuerzas que originan estos movimientos. De acuerdo a la Fig. 1 la posición cartesiana del efecto final $[X_E, Y_E, \phi_E]$ con respecto a los valores angulares de las articulaciones (q_1, q_2, q_3) (cinemática directa) está definida por las ecuaciones (6)-(8).

$$X_E = l_1 \cos(q_1) + l_4 \cos(\pi + q_2) + l_5 \cos(\pi + q_2 + q_3) \quad (6)$$

$$Y_E = l_1 \sin(q_1) + l_4 \sin(\pi + q_2) + l_5 \sin(\pi + q_2 + q_3) \quad (7)$$

$$\phi_E = \pi + q_2 + q_3 \quad (8)$$

Por otra parte, si consideramos que conocemos la posición y orientación del efecto final $[X_E, Y_E, \phi_E]$ entonces por el desacople cinemático sabemos la

posición y orientación de la muñeca $[X_m, Y_m, \phi_m]$ esta dada en (9)-(11).

$$X_m = X_E - l_5 \cos(\phi_E) \quad (9)$$

$$Y_m = X_E - l_5 \sin(\phi_E) \quad (10)$$

$$\phi_m = \phi_E \quad (11)$$

Así, los valores articulares necesarios para lograrla estarán definidos por las ecuaciones (12)-(14) (cinemática inversa).

$$q_1 = \text{Atan2}\left(\frac{Y_m}{X_m}\right) - \gamma \quad (12)$$

$$q_2 = q_1 + q'_2 + \pi \quad (13)$$

$$q_3 = \phi_4 - \pi - q_2 \quad (14)$$

Donde:

$$q_2' = \text{Atan2}\left(\frac{\sqrt{4l_1^2 l_4^2 - (X_m^2 + Y_m^2 - l_1^2 - l_4^2)^2}}{X_m^2 + Y_m^2 - l_1^2 - l_4^2}\right)$$

$$\gamma = \text{Atan2}\left(\frac{l_4 \sin(q_2')}{l_1 + l_4 \cos(q_2')}\right)$$

2.3 Jacobiano.

En el campo de la robótica, la matriz Jacobiana J relaciona las velocidades en el espacio articular $\dot{q} \in R^3$ a las velocidades en el espacio cartesiano $\dot{\xi} \in R^3$ y viceversa. Puede definirse una matriz Jacobiana de cualquier dimensión, dado que el número de filas de la matriz será el número de grados de libertad en el espacio cartesiano que se estén considerando y el número de columnas será el número de articulaciones actuadas en el manipulador. Derivando las ecuaciones (6),(7) y (8) con respecto al tiempo, obtenidas en el análisis de la cinemática directa, se obtienen las ecuaciones de velocidad, donde $\dot{\xi} = J(q)\dot{q}$ donde $\dot{\xi} = [\dot{X}_E, \dot{Y}_E, \dot{\phi}_E]^T$ y $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$, entonces la matriz Jacobiana queda como se muestra en la ecuación (15) donde $J_{11} = -l_1 \sin(q_1)$, $J_{12} = l_4 \sin(q_2) + l_5 \sin(q_2 + q_3)$, $J_{13} = l_5 \sin(q_2 + q_3)$, $J_{21} = l_1 \cos(q_1)$, $J_{22} = -l_4 \cos(q_2) - l_5 \cos(q_2 + q_3)$, $J_{23} = -l_5 \cos(q_2 + q_3)$, $J_{31} = 0$, $J_{32} = 1$ y $J_{33} = 1$.

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \quad (15)$$

3. Control por dinámica inversa.

Sea $q \in R^3$, $\dot{q} \in R^3$, $\ddot{q} \in R^3$ el vector de coordenadas, velocidades y aceleraciones generalizadas en el espacio articular, $\tau \in R^3$ el vector de entrada,

considerando la estrategia de control mostrada en (16), el origen del sistema en lazo cerrado es asintóticamente estable, donde $q_d \in R^3$, $\dot{q}_d \in R^3$, $\ddot{q}_d \in R^3$, son los vectores de posición, velocidad y aceleración deseados en el espacio articular, el error articular se define como $e_q = q_d - q \in R^3$, la velocidad del error como $\dot{e}_q = \dot{q}_d - \dot{q} \in R^3$ y la aceleración del error como $\ddot{e}_q = \ddot{q}_d - \ddot{q} \in R^3$

$$\begin{aligned}\tau_c &= M(q)V + C\dot{q} + G \\ V &= (\ddot{q}_d + K_d\dot{e}_q + K_p e_q)\end{aligned}\quad (16)$$

Demostración.

El sistema en lazo cerrado está dado por (20).

$$M\ddot{q} + C\dot{q} + G = M(q)V + C\dot{q} + G \quad (17)$$

$$M\ddot{q} = M(q)V \quad (18)$$

$$\ddot{q} = \ddot{q}_d + K_d\dot{e}_q + K_p e_q \quad (19)$$

$$\ddot{e}_d + K_d\dot{e}_q + K_p e_q = 0 \quad (20)$$

En las coordenadas $z = [e, \dot{e}]^T$, el sistema en lazo cerrado descrito en (20) se puede representar como en (21) el cual presenta un único equilibrio en el origen $z = [0, 0]^T \in R^{2n}$.

$$\dot{z} = Az \quad (21)$$

$$\text{Donde: } A = \begin{bmatrix} 0 & 1 \\ -K_p & -K_d \end{bmatrix}$$

Considerando la función candidata de Lyapunov $V(z)$ mostrada en (22), la derivada con respecto al tiempo de $V(z)$ será (23) donde $A^T P + PA = -Q$.

$$V(z) = \frac{1}{2} z^T P z \quad (22)$$

$$\dot{V} = -z^T Q z \quad (23)$$

Considerando que $K_p \in R^{3x3}$ y $K_d \in R^{3x3}$ son matrices diagonales con términos positivos y utilizando el teorema generalizado de Krasowski [5], se puede obtener 2 matrices simétricas definidas positivas $P \in R^{nxn}$, $Q \in R^{nxn}$ de tal manera que para todo $z \neq 0$ la matriz $F(z) = A^T P + AP + Q$ es definida negativa, por lo que $\lim_{t \rightarrow \infty} z(t) = 0$.

La transformada de Laplace del sistema en lazo cerrado está dada como $s^2 + K_d s + K_p = 0$ en donde se observa que para sintonizar las ganancias del sistema de control se debe considerar $K_d = 2\xi W_n$ y $K_p = W_n^2$, donde $\xi \in (0, 1]$ es el factor de amortiguamiento y $W_n > 0$ es la frecuencia natural del sistema.

4. Resultados Experimentales.

Los resultados experimentales consisten en implementar el sistema de control paralelo en el manipulador para el problema de seguimiento. El sistema de control se programa en una PC con el programa de Matlab-Simulink y con una tarjeta de adquisición de datos Sensoray 626 se obtienen las salidas del sistema así como le proporciona la acción de control al sistema.

Se generó la trayectoria de un círculo expresada en (24)–(26), donde: $X = [X_d \ Y_d \ \dot{\phi}_d]^T$ es el vector de posición Cartesiana deseada. Al derivar con respecto al tiempo X se obtiene $\dot{X} = [\dot{X}_d \ \dot{Y}_d \ \dot{\phi}_d]^T$ que es el vector de velocidad Cartesiana deseada y $\ddot{X} = [\ddot{X}_d \ \ddot{Y}_d \ \ddot{\phi}_d]^T$ es el vector de aceleraciones Cartesianas deseada. La trayectoria tiene un radio de $r = 1m$ y una frecuencia de $f = 1/20$, la cual determina cuánto tiempo se tarda el manipulador en realizar un ciclo.

$$X_d = .32 + r(\cos(2 * pi * f * t)) \quad (24)$$

$$Y_d = .32 + r(\sin(2 * pi * f * t)) \quad (25)$$

$$\dot{\phi}_d = 0 \quad (26)$$

Debido a que el sistema de control trabaja en el espacio articular, se necesita mapear la posición, velocidad y aceleración Cartesiana al espacio articular con el uso de la cinemática inversa, Jacobiano. Las ganancias del controlador son calculadas como se mostró en la sección anterior tomando en cuenta una frecuencia natural W_n independiente para cada articulación, quedando los valores como $W_{n1} = 50$, $W_{n2} = 120$ y $W_{n3} = 650$.

En la Tabla 1 se muestran los parámetros cinemáticos y dinámicos del manipulador obtenidos a partir de Solidworks.

Parámetro	Valor	Parámetro	Valor
m_1	0.32224 Kg	l_4	0.25 m
m_2	0.18940 Kg	l_5	0.072 m
m_3	0.16919 Kg	l_{c1}	0.05249 m
m_4	0.95876 Kg	l_{c2}	0.01570 m
m_4	0.13494 Kg	l_{c3}	0.100 m
l_1	0.2 m	l_{c4}	0.06437
l_2	0.05 m	l_{c5}	0.01859233 m
l_3	0.2 m		

Tabla 1. Parámetros del Manipulador Paralelo.

La Fig. 2 muestra la trayectoria deseada que se generó para este experimento. La primera parte del movimiento es la trayectoria que sigue el robot manipulador para salir de su posición de reposo y colocarse en su posición inicial.

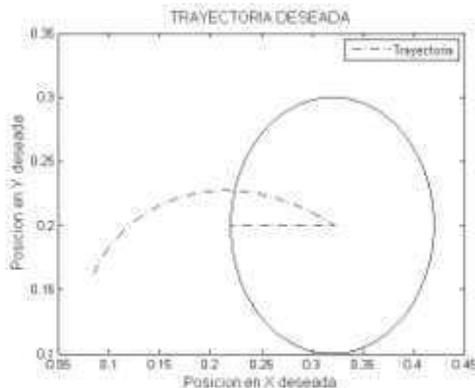


Figura 2. Trayectoria Deseada

En la figura 3 se muestra el comportamiento de cada una de las articulaciones con respecto a la trayectoria deseada y se puede observar que el controlador hace que ese movimiento deseado se consiga. En la figura 4 se muestra el error que presenta cada articulación con el control por dinámica inversa. La primera parte del movimiento que se muestra en la Fig. 4 corresponde al movimiento que hace el manipulador para salir de su posición de reposo (totalmente retraído), dicha trayectoria para salir de su estado de reposo dura hasta $t=10$ s. Después de este movimiento comienza a realizar la trayectoria del círculo que se generó, es por eso que en ese instante se ve un pico en el error de posición de cada articulación, después de este pico se observa que el sistema se estabiliza.

5. Conclusiones.

Se validó de manera experimental el sistema de control por dinámica inversa para el manipulador paralelo, dando como resultado que el modelo dinámico obtenido es acorde a la realidad. Sin embargo los errores en el espacio articular originados por el sistema de control se debe principalmente a las incertidumbres que existen en los parámetros cinemáticos y dinámicos así como a que no se incluyeron todas las dinámicas involucradas en el movimiento dinámico del robot, como por ejemplo, la fricción.

Se mostró de manera experimental el buen desempeño del control por dinámica inversa para el problema de “seguimiento”, recordando que este desempeño dependerá directamente del modelo dinámico obtenido del sistema y de los parámetros del sistema. Por otro lado resulta no trivial la sintonización de las ganancias del controlador.

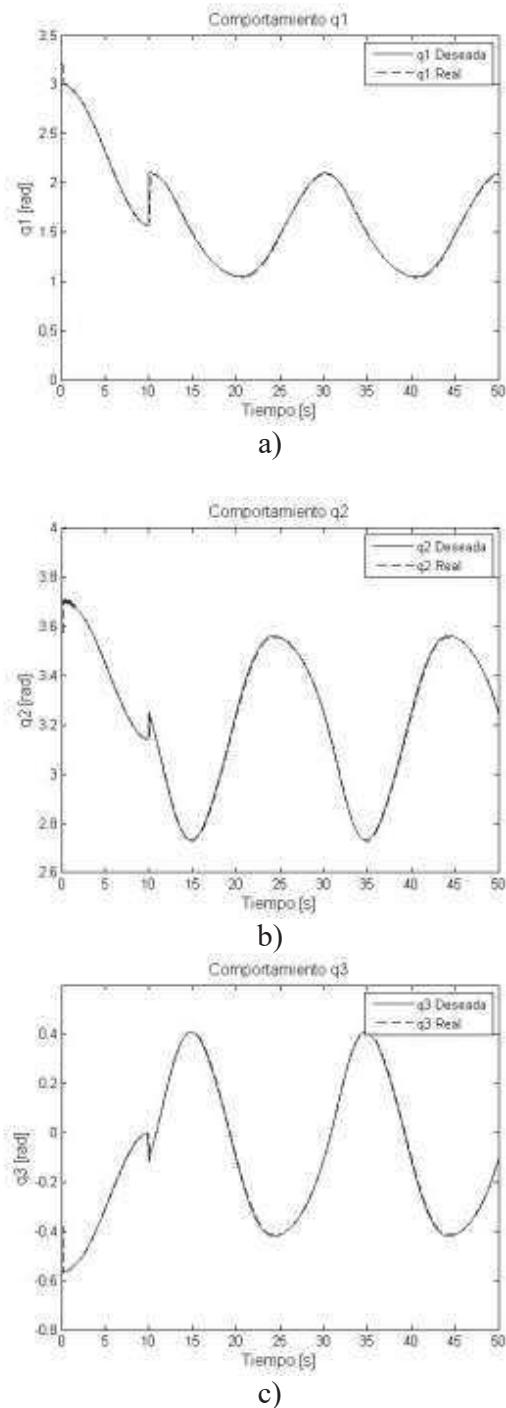


Figura 3. a) Comportamiento q_1 . b) Comportamiento q_2 . c) Comportamiento q_3 .

Trabajo futuro involucra implementar estrategias de control adaptable en el manipulador paralelo, así como la sintonización óptima de los parámetros del controlador.

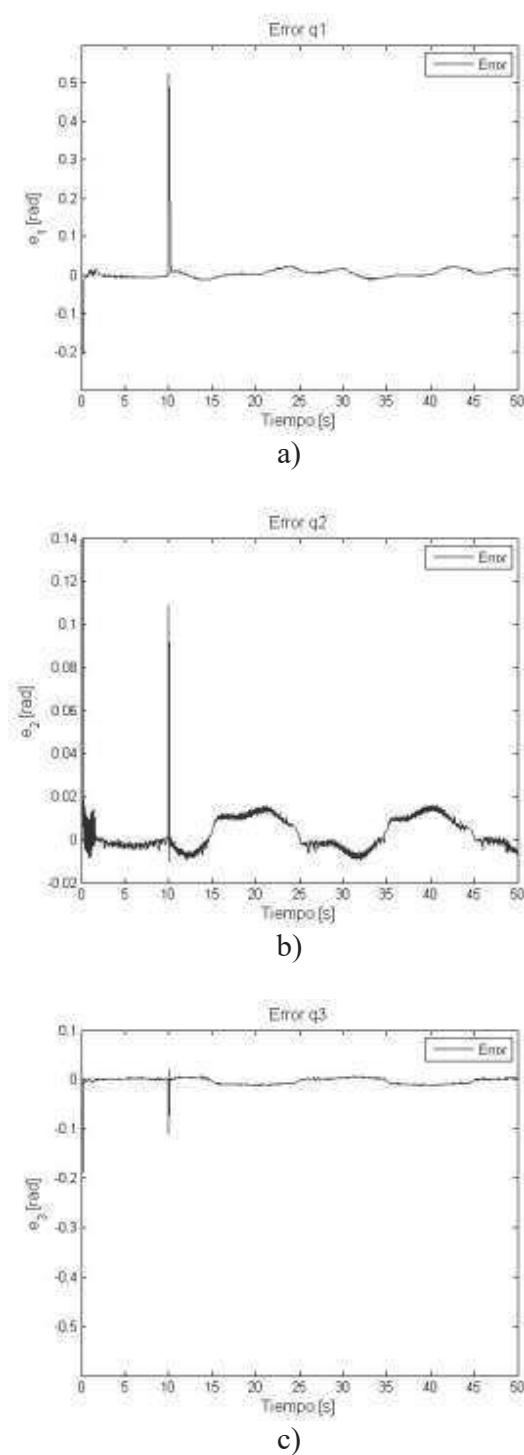


Figura 4. a) Error q_1 . b) Error q_2 . c) Error q_3 .

Referencias

- [1] M. Spong, S. Hutchinson, M. Vidyasagar. “*Robot Modelling and Control*”, First Edition, E.U.A. 2002
- [2] D. Yang, Y. Yamane, X. Zhang, Rongyan Zhu. “*A Sliding Mode Control of Robot Manipulator with Variable Payload*”, International Conference on Intelligent Processing Systems. Beijing, China, 1997.
- [3] C. Zheng, Y. Su. “*PID Control of Robot Manipulators in Task Space*”. Proceedings of the 8th World Congress on Intelligent Control and Automation. Jinan, China. Julio 2010.
- [4] S. Karunakar, A. Goldenberg. “*Contact Stability in Model-Based Force Control Systems of Robot Manipulators*” International Conference on Intelligent Processing. 1988.
- [5] M. G. Villareal-cervantes, C. A. Cruz-Villar, S. A. Saldívar Bárcenas, L. E. García Sosa, “*Dynamic Coupling Between a Parallel Manipulator and an Omnidirectional Mobile Platform*”, XV Latinamerican Control Conference, Lima, Perú, Octubre 2012.
- [6] J. Craig, “*Robótica*”, Prentice Hall, Tercera Edición. 2006
- [7] A. Barrientos, “*Fundamentos de Robótica*”, S.A. McGraw Hill, Interamerica de España, 2007.

Metodología para la Detección del Desalineamiento en Motores de Inducción Alimentado con Variador de Velocidad

Mejía-Barrón A.¹, García-Ramírez A.G.¹, Osornio-Rios R.A.¹ y Romero-Troncoso R.J.²

HSPdigital CA-Mecatrónica,¹Universidad Autónoma de Querétaro, Facultad de Ingeniería, Campus San Juan del Río, Rio Moctezuma 249, Col. San Cayetano, San Juan del Río, Querétaro, 76807. ² Universidad de Guanajuato, División de Ingenierías, Campus Irapuato-Salamanca. Carretera Salamanca –Valle de Santiago Km 2.5+1.8

Comunidad de Palo Blanco Salamanca, Gto. C.P. 36885.

{amejia, aggarcia, raosornio, troncoso}@hspdigital.org

Resumen

El objetivo de este artículo es presentar una metodología que colabore en el diagnóstico de desalineamiento en motores de inducción alimentados con variador de velocidad. La metodología propuesta se basa en el análisis de impedancia estimada a través de las envolventes de las transformadas Park de voltaje y corriente de fase medidas en el estator del motor. Por medio de la metodología propuesta se observan las variaciones de la envolvente de impedancia del motor desalineado contra la impedancia del motor en estado sano. La experimentación se llevó a cabo con un motor de 1hp, alimentado por variador de velocidad, acoplado a una carga a través de una banda.

Palabras clave: Análisis de impedancia, diagnóstico de fallas, motor de inducción, variadores de velocidad.

1. Introducción

Los motores de inducción son elementos clave en la industria debido a su robustez, fácil construcción, bajo costo y versatilidad, representan el consumo de más de la mitad de la energía producida en el mundo. Las aplicaciones típicas de los motores de inducción en la industria son muy amplias, se emplean en bombas, máquinas de transporte, procesos metalúrgicos, elevadores, máquinas textiles, robótica, sistemas de ventilación, etc. En consecuencia, la detección temprana de fallas en motores de inducción es muy importante para la industria [1], debido a que estas fallas pueden producir interrupciones no previstas en las líneas de producción, que representan pérdidas económica debido a los mantenimientos no previstos. Además, los motores de inducción son

fuentes de armónicos para la red eléctrica [2], que se deben generalmente a las fallas que pueda presentarse en el motor. Los armónicos inducidos a la red eléctrica pueden ser causa de fallas de otros dispositivos, de ahí la importancia de la detección temprana de fallas en motores de inducción. En general, entre un 40% a 50% de las fallas en motores se debe a fallas en baleros (falla en la cual se enfocan la mayoría de los trabajos de investigación y que está extensamente reportada), las fallas en rotor representan de un 5% a 10% y aproximadamente el 12% se debe a fallas de desbalance y desalineamiento [3]. Por otra parte, la alimentación de motores de inducción a partir de variadores de velocidad, permite extender su vida útil y ahorro de energía, pero hace más difícil la detección de fallas debido a los armónicos inducidos por la operación del variador de velocidad [4], lo que genera nuevos retos para la detección de fallas.

Una de las principales causas de daños en máquinas rotativas se atribuye directamente a condiciones de desalineamiento [5] y hay pocos trabajos de investigación reportados. La principal causa de este daño se atribuye a las excesivas fuerzas que inciden en los rotores, baleros, sellos y carcásas. De acuerdo a lo reportado en la literatura, los métodos más utilizados en el diagnóstico de condiciones de desalineamiento se llevan a cabo principalmente a partir de análisis de la firma de corriente del motor (MCSA), análisis de vibraciones y análisis de ruido. El MCSA extrae las características específicas de diferentes condiciones de falla a partir del procesamiento del espectro de corriente del estator usando transformada rápida de Fourier (FFT), transformada wavelet discreta (DWT) o diversas combinaciones [6], [7]. También, se han encontrado efectos significativos de la carga en el desalineamiento del eje del motor, dando lugar a la falta de componentes de frecuencia

importantes en el espectro de corrientes [8]. Técnicas a partir del análisis de vibraciones o combinación de parámetros como vibraciones y corrientes han sido empleadas para la detección de fallas en motores de inducción [3], [9], [10]. Técnicas basadas en inteligencia artificial son consideradas en el monitoreo y diagnóstico de fallas en máquinas eléctricas. Técnicas como redes neuronales artificiales (ANN), y mapas auto-organizados (SOM) se han empleado para la detección de desalineamiento y otras fallas en rotor a partir de MCSA [11], [12]. A su vez, se han empleado técnicas basadas en análisis estadístico para la detección desalineamiento a partir de análisis de corriente basado en el modelo oculto de Markov [13]. Cabe mencionar que en los trabajos anteriormente citados los motores son alimentados directamente de la línea y no utilizan variador de velocidad. Por otra parte, se han desarrollado técnicas para el análisis de fallas en motores alimentado por variador de velocidad, [14]–[16]. Se ha detectado la falla de desalineamiento por variador de velocidad a partir de un sistema de inferencias neuro-difuso adaptativo, por medio del procesamiento de la velocidad, corriente y el torque del motor [16]. Finalmente las metodologías para el análisis de la falla de desalineamiento en motores, la mayoría se basa en análisis de corriente, y existen pocas metodologías basadas en la impedancia del motor para el diagnóstico de dicha falla, siendo una posibilidad para la mejora del diagnóstico de la falla mediante la propuesta de metodologías para el cálculo de dicha impedancia y los efectos que esta muestra al presentarse la falla.

En este artículo se presenta una metodología para la detección de desalineamiento mecánico de un motor de inducción alimentado por variador de velocidad. La metodología propuesta está basada en el análisis de envolventes de las componentes de la transformada Park de la impedancia obtenida a partir de la corriente y el voltaje del estator del motor. La metodología propuesta se evalúa con un motor de 1hp, y es probada para distinguir entre las condiciones de motor sano y motor desalineado.

2. Fundamentación Teórica

2.1 Transformada Park

La transformada Park se emplea en análisis de sistemas trifásicos, donde las corrientes y voltajes trifásicos son representados en un marco de referencia rotatorio (d, q). Para obtener la transformada Park de un sistema trifásico se deben obtener una transformación para un espacio

vectorial estacionario (α, β) en donde dichas transformaciones se obtienen a partir de (1) y (2) para voltajes y corrientes respectivamente.

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \times \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \times \begin{bmatrix} i_A \\ i_B \\ i_C \end{bmatrix} \quad (2)$$

Donde v_A, v_B y v_C son los voltajes instantáneos de fase, y i_A, i_B e i_C son las corrientes instantáneas de fase. Teniendo las componentes estacionarias (α, β), la transformada Park se obtiene aplicando sólo una rotación en el eje z la cual se define como:

$$[Matriz\ Park] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (3)$$

Siendo:

$$\theta = \omega t \quad (4)$$

Donde ω es la velocidad de rotación (rad/s) del marco rotativo. Finalmente, se obtiene la transformación del marco (α, β) a (d, q), de corriente y de voltaje como se muestra en la ecuación (5).

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (5)$$

2.2 Transformada Hilbert

La transformada Hilbert (HT) es una herramienta matemática usada para el seguimiento de envolventes de señales, la cual es definida para señales reales como:

$$x_{HT}(t) = x(t) \times \frac{1}{\pi t} \quad (6)$$

Una manera útil de calcular la HT de $x(t)$ es a través de la señal analítica $z(t)$ que se muestra en (5), donde $A(t)$ y $\theta(t)$ se denominan las envolventes de la señal o la amplitud instantánea y fase de $z(t)$, respectivamente. Esta tiene parte real $x(t)$ y parte imaginaria $\overline{x(t)}$ las cuales son los datos originales y su HT, respectivamente. La $\overline{x(t)}$ es una versión de los datos originales con un desplazamiento de fase de 90° ; por lo tanto, los senos son transformados en cosenos y viceversa.

$$z(t) = x(t) + j\overline{x(t)} = A(t)e^{j\theta(t)} \quad (6)$$

Está claramente en términos de $x(t)$ y $\overline{x(t)}$ que:

$$A(t) = \sqrt{x^2(t) + \overline{x^2(t)}} \quad (7)$$

$$\theta(t) = \tan^{-1}\left(\frac{\overline{x(t)}}{x(t)}\right) \quad (8)$$

3. Metodología

La metodología se basa en la transformada Park, que permite la representación de un sistema trifásico de un motor de inducción a un sistema de dos dimensiones, la transformada Hilbert, de la cual se obtiene la envolvente de la señal obtenida a partir de (9) y (10) para cada muestra k , la envolvente de la impedancia es estimada a partir de (11) donde v_{HT} e i_{HT} , corresponden a las envolventes de v_{DQ-k} e i_{DQ-k} , respectivamente.

$$v_{DQ-k} = \sqrt{v_{D-k}^2 + v_{Q-k}^2} \quad (9)$$

$$i_{DQ-k} = \sqrt{i_{D-k}^2 + i_{Q-k}^2} \quad (10)$$

$$z_{HT} = \frac{v_{HT}}{i_{HT}} \quad (11)$$

La estimación de la impedancia se lleva a cabo por medio de software, a partir del análisis de señales de corriente y voltaje de fase, medidas en la salida del variador de velocidad, alimentado de la línea a 220V (F_A , F_B , F_C), como se muestra en la Fig. 1(a) a partir de un sistema de adquisición en plataforma FPGA. En la Fig. 1(b) se muestra el diagrama a bloques de la metodología propuesta para la estimación de la impedancia, donde los voltajes y corrientes de fase trifásicos son las entradas para la estimación de las componentes de la transformada Park, la salida de los bloques de la transformada Park está dada por el módulo de las componentes (d , q), dada por las ecuaciones (9) y (10). Posteriormente, se estima la envolvente de las magnitudes, obtenidas del bloque de la transformada Park, a través de la transformada Hilbert. Entonces, se realiza la estimación de la impedancia a partir de (11) donde se obtiene z_{HT} , por último, se aplica un filtro de diezmado en tiempo por un factor de 1024 para reducir el número de muestras sin perder las características de la señal, obteniendo la impedancia estimada (z) por la metodología propuesta.

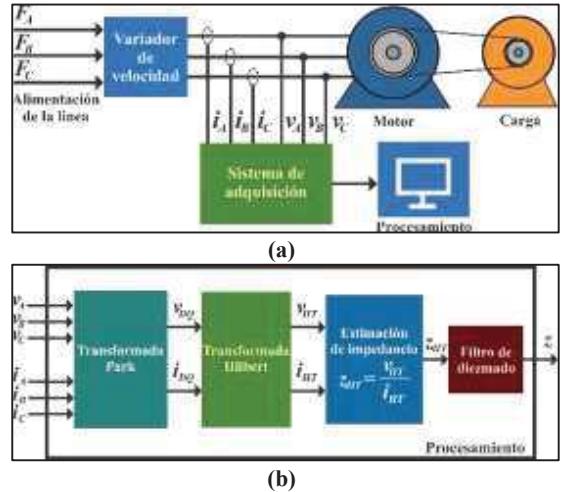


Fig. 1. a) Metodología para la medición de parámetros. b) Metodología para la estimación de impedancia.

A partir de la metodología propuesta se realiza el análisis bajo condiciones de motor sano y motor desalineado para obtener las características que puedan emplearse para la detección del desalineamiento.

4. Experimentación y resultados

4.1 Puesta del experimento

Las señales de corriente y voltaje son proporcionadas por un variador de velocidad (modelo WEG CFW08) conectado a un motor de inducción. El variador de velocidad opera en un rango de 0 a 100Hz usando una resolución de 0.1Hz. Se emplea un motor de inducción (modelo WEG 00136APE48T) para las pruebas de la metodología propuesta para el análisis de motor sano y desalineado. La velocidad del motor se controla por el variador de velocidad a 3Hz, 30Hz y 60Hz. El motor de inducción consta de dos polos, veintiocho barras y recibe una alimentación de 220VCA. La carga mecánica aplicada consta de un alternador, el cual representa un cuarto de la carga del motor. Las tres fases de señales de corriente son adquiridas usando tres sensores de efecto hall modelo L08P050D15, de la Tamura Corporation. Por otra parte, las señales de voltaje son monitoreadas por medio de un divisor de voltaje y un amplificador de aislamiento de precisión modelo ISO124PND con el objetivo de obtener un aislamiento entre el sistema de alimentación y el sistema de monitoreo. El sistema de adquisición para las señales de corriente y voltaje es proprietario y consiste en convertidores analógico digital de 16-

bit y 4 canales con salida serial modelo ADS8341 con una frecuencia de muestreo de 4kHz. El sistema de instrumentación fue calibrado por medio del Fluke 435 como en [14], este sistema usa una frecuencia de muestreo $f_s = 4\text{kHz}$ obteniendo 120,000 muestras de cada fase de corriente y voltaje durante 30 segundos desde el arranque hasta el estado estable de la operación del motor. El motor se controla a través de un relevador para realizar las pruebas de manera automática. Para la adquisición se utilizó una plataforma en FPGA Spartan 3EXC3S1600 a 48 MHz de operación. Para la adquisición se realizan pruebas de 30 segundos por cada velocidad propuesta que conlleva el estado transitorio y el estado estable de la velocidad del motor, los datos se adquieren a partir del sistema basado en FPGA, donde posteriormente son enviados a la computadora vía USB, y almacenados para el análisis por software a través de la metodología propuesta. En la Fig. 2, se muestran los componentes descritos para el montaje de la experimentación.

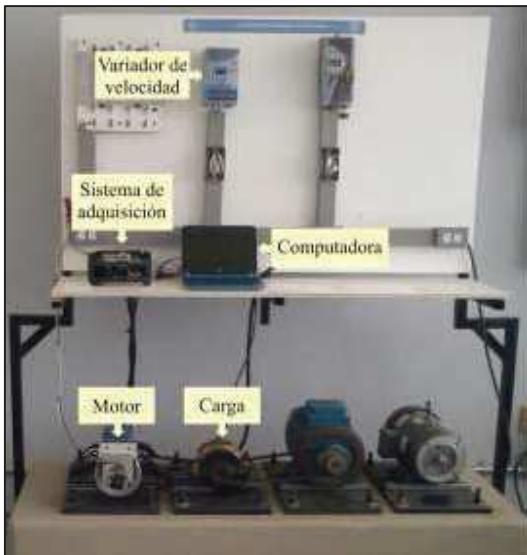


Fig. 2. Puesta del experimento.

La condición de desalineamiento fue llevada a cabo a través del desplazamiento hacia delante de la banda en la polea del alternador, de manera que los ejes transversales de rotación del motor no estén alineados. En la Fig. 3 se muestra el motor en estado sano, y en la Fig. 4 se muestra el motor con desalineamiento aproximado de la banda de 5° .



Fig. 3. Motor Alineado (estado sano).



Fig. 4. Motor desalineado.

4.2 Resultados

Se realizaron pruebas de 30 segundos a diferentes tres diferentes velocidades (3Hz, 30Hz y 60Hz), analizando el estado transitorio y el estado estable de la impedancia estimada a partir de los voltajes y corrientes de fase, en Fig. 5, se muestra la envolvente de la impedancia estimada a una frecuencia de variador de velocidad de 3Hz, para motor sano y desalineado.

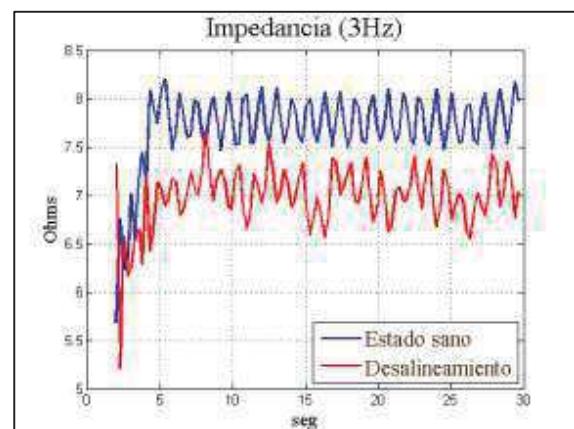


Fig. 5. Envolvente de impedancia de motor a 3Hz.

Se realizaron pruebas a una frecuencia de 30Hz, como se observa en Fig. 6 la envolvente de la impedancia en motor sano y motor con desalineamiento aumenta, sin embargo, se sigue conservando una diferencia en el estado estable para el motor sano y el motor desalineado.

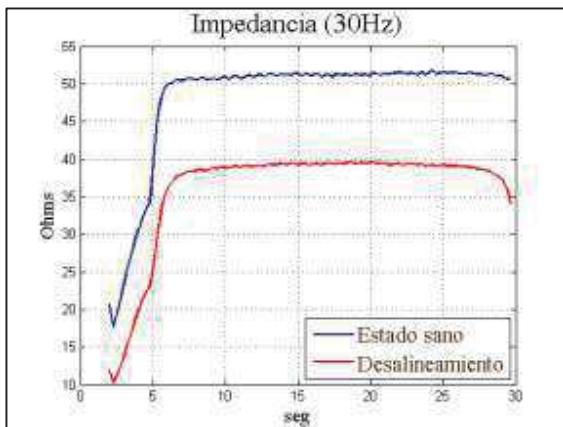


Fig. 6. Envolvente de impedancia de motor a 30Hz

Finalmente, se realizaron pruebas a una frecuencia de 60Hz como se muestra en Fig. 7 la magnitud de la impedancia aumenta conforme lo hace la frecuencia de operación del variador de velocidad y en todos los casos la envolvente de la impedancia del motor sano es siempre mayor a envolvente de la impedancia en el motor desalineado.

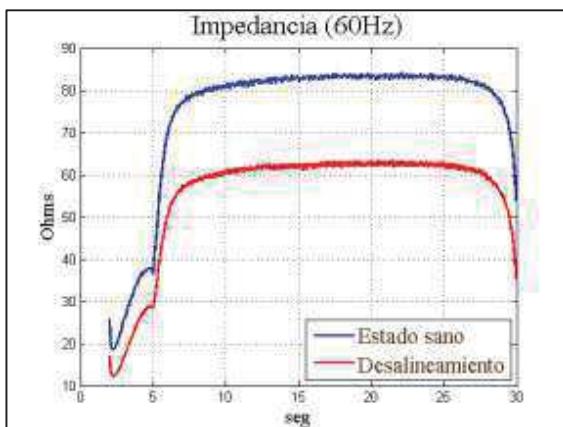


Fig. 7. Envolvente de impedancia de motor a 60Hz.

5. Conclusiones

En este trabajo se propone una metodología que pueda servir para extraer características provenientes de señales de corriente y voltaje que puedan servir como herramienta para la detección de desalineamiento. En este caso se propone un análisis de impedancia, la cual es estimada a partir

de las envolventes de la transformada Park de corrientes y voltajes, como se observa en los resultados una variación en la impedancia obtenida debida al desalineamiento la cual puede ser empleada para el diagnóstico de la falla. Finalmente, la metodología propuesta en este trabajo puede ser empleada, en conjunto con otras técnicas reportadas, para mejorar el diagnóstico de desalineamiento.

Agradecimiento

Este trabajo se realizó con el apoyo parcial del proyecto SEP-CONACyT 222453-2013 y la beca CONACyT 296868.

Referencias

- [1] A. Pilloni, A. Pisano, M. Riera-Guasp, R. Puche-Panadero, and M. Pineda-Sánchez, “Fault Detection in Induction Motors,” in *AC Electric Motors Control: Advanced Design Techniques and Applications*, 2013, pp. 275–309.
- [2] S. Sengupta, S. Chattopadhyay, and M. Mitra, *Electric Power Quality*. Springer Science + Business Media, 2011.
- [3] A. Garcia-Perez, R. de Jesus Romero-Troncoso, E. Cabal-Yepez, and R. A. Osornio-Rios, “The Application of High-Resolution Spectral Analysis for Identifying Multiple Combined Faults in Induction Motors,” *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 5. pp. 2002–2010, 2011.
- [4] O. Duque-Perez, L. A. Garcia-Escudero, D. Morinigo-Sotelo, P. E. Gardel, and M. Perez-Alonso, “Condition monitoring of induction motors fed by Voltage Source Inverters. Statistical analysis of spectral data,” *Electrical Machines (ICEM), 2012 XXth International Conference on*. pp. 2479–2484, 2012.
- [5] J. Piotrowski, *Shaft Alignment Handbook, Third Edition*. Taylor & Francis, 2006.
- [6] R. R. Obaid, T. G. Habetler, and R. M. Tallam, “Detecting load unbalance and shaft misalignment using stator current in inverter-driven induction motors,” *Electric Machines and Drives Conference, 2003. IEMDC’03. IEEE International*, vol. 3. pp. 1454–1458 vol.3, 2003.
- [7] E. Cabal-Yepez, R. A. Osornio-Rios, R. J. Romero-Troncoso, J. R. Razo-Hernandez, and R. Lopez-Garcia, “FPGA-Based Online

- [8] Induction Motor Multiple-Fault Detection with Fused FFT and Wavelet Analysis,” in *Reconfigurable Computing and FPGAs, 2009. ReConFig '09. International Conference on*, 2009, pp. 101–106.
- [9] J. Rastegar Fatemi, H. Henao, G.-A. Capolino, and S. Sieg-Zieba, “Load influence on induction machine torque and stator current in case of shaft misalignment,” in *Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE*, 2009, pp. 3449–3454.
- [10] L. M. C. Medina, R. de Jesus Romero-Troncoso, E. Cabal-Yepez, J. de Jesus Rangel-Magdaleno, and J. R. Millan-Almaraz, “FPGA-Based Multiple-Channel Vibration Analyzer for Industrial Applications in Induction Motor Failure Detection,” *Instrum. Meas. IEEE Trans.*, vol. 59, no. 1, pp. 63–72, 2010.
- [11] V. P. Raj, K. Natarajan, and T. G. Girikumar, “Induction motor fault detection and diagnosis by vibration analysis using MEMS accelerometer,” in *Emerging Trends in Communication, Control, Signal Processing Computing Applications (C2SPCA), 2013 International Conference on*, 2013, pp. 1–6.
- [12] M. Pawlak and C. T. Kowalski, “Low-cost Embedded System for the IM Fault Detection using Neural Networks,” no. 5, 2010.
- [13] J. M. Bossio, C. H. De Angelo, G. R. Bossio, and G. O. Garcia, “Fault diagnosis on induction motors using Self-Organizing Maps,” in *Industry Applications (INDUSCON), 2010 9th IEEE/IAS International Conference on*, 2010, pp. 1–6.
- [14] T. Wongswan, P. Tangamchit, C. Prapanavarat, and M. Pusayatanont, “Motor Misalignment Detection Based on Hidden Markov Model,” in *Communications and Information Technologies, 2006. ISCIT '06. International Symposium on*, 2006, pp. 422–427.
- [15] A. G. Garcia-Ramirez, R. A. Osornio-Rios, A. Garcia-Perez, and R. D. J. Romero-Troncoso, “FPGA-based smart-sensor for fault detection in VSD-fed induction motors,” in *Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED), 2013 9th IEEE International Symposium on*, 2013, pp. 233–240.
- [16] E. Cabal-Yepez, R. J. Romero-Troncoso, A. Garcia-Perez, R. A. Osornio-Rios, and R. Alvarez-Salas, “Multiple fault detection through information entropy analysis in ASD-fed induction motors,” in *Diagnostics for Electric Machines, Power Electronics Drives (SDEMPED), 2011 IEEE International Symposium on*, 2011, pp. 391–396.
- [17] J. Guzinski, H. Abu-Rub, A. Iqbal, and S. K. M. Ahmed, “Shaft misalignment detection using ANFIS for speed sensorless AC drive with inverter output filter,” in *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, 2011, pp. 2138–2143.

Aproximación Metodológica en la Enseñanza de Lenguajes de Programación Orientados a Sistemas Mecatrónicos

Curiel Aguirre Fernando¹ y Jiménez López Eusebio².

¹Universidad Estatal de Sonora, fher.curiel@gmail.com
²CIAAM de la Universidad Tecnológica del Sur de Sonora, ULSA Noroeste-IIMM, ejimenezl@msn.com

Resumen

El aprendizaje de la disciplina Mecatrónica se ha convertido en un reto para las universidades mexicanas. Algunas instituciones educativas les es difícil competir en equipamiento e infraestructura con las principales potencias en esta área, sin embargo, cabe mencionar que existen un sinnúmero de recursos libres tanto de hardware como de software que hacen que la tarea de enseñar-aprender la Ingeniería Mecatrónica sea asequible a la mayoría de los estudiantes y profesores. En este artículo se analizan y se proponen estrategias de enseñanza-aprendizaje (E-A) que encaminen a los estudiantes a la aplicación del paradigma de la programación orientado a sistemas embebidos relacionados con las áreas de automatización, robótica y mecatrónica, utilizando hardware y software de fácil acceso. Tales estrategias pretenden impactar la asignatura de Lenguajes de Programación con el enfoque propio de la ingeniería y que, por supuesto se puede extender su aplicación a otras asignaturas como lo son: Tecnologías de Información y de Comunicaciones, Microcontroladores y Robótica, que también integran el currículo de la mayoría de programas educativos de ingeniería mechatrónica, destacando así la importancia de los sistemas informáticos que dotan de cierta “inteligencia” a los prototipos desarrollados por los alumnos y profesores.

Palabras clave: Enseñanza de la Mecatrónica, Modelo Pedagógico, interdisciplinariedad, Lenguajes de Programación.

1. Introducción

La disciplina mechatrónica se remonta a los años 60 y nace en la empresa japonesa Yaskawa. En México se cuenta con poco más de 20 años de experiencia en este campo y por ende, representa un área de oportunidad para la enseñanza de esta fascinante carrera [1]. La robótica educativa ha cobrado mucho auge en los últimos años y es tema de

estudio en muchas universidades que cuentan con el Programa de Estudios (PE) de Ingeniería Mecatrónica (IM). Sin embargo, el estilo pedagógico y la forma de abordar los temas en la ingeniería mechatrónica distan de ser convencionales en comparación con el sistema tradicional educativo para las ingenierías puras. El origen, antecedentes y desarrollo de la ingeniería mechatrónica, es una guía en la aplicación de técnicas pedagógicas apropiadas, en las que se debe transitar por estrategias de Enseñanza-Aprendizaje (E-A) multidisciplinares a las interdisciplinares y por supuesto aspirar a la transdisciplinariedad.

En este artículo se analiza y discute acerca de la metodología de la enseñanza mechatrónica y de las técnicas E-A: los ciclos propedéuticos, el modelo holístico de la educación, el aprendizaje basado en competencias (ABC), el aprendizaje basado en problemas (ABP) y por supuesto el aprendizaje por proyectos.

2. Evolución de la Ingeniería Mecatrónica

La Ingeniería Mecatrónica surge como una propuesta de integración de conocimientos entre la Ingeniería Mecánica y la Ingeniería Electrónica. El concepto apareció en Japón el 12 de julio de 1969 en un reporte técnico realizado por Testuro Mori y Ko Kikuchi en la empresa del sector eléctrico Yaskawa Co. En dicho reporte se muestra la forma en cómo se aplicó la técnica conocida como Kaizen, cuyo término está relacionado con acciones de mejora continua. En enero de 1972 la empresa japonesa Yaskawa Co. obtiene el derecho de marca del concepto de Mecatrónica con número de registro 946594 (Japan Trademark Registration). Sin embargo, fue hasta principios de los 80s donde el concepto de mechatrónica se empieza hacer popular en forma mundial, retomado por el sector industrial y las universidades [2]. Por otro lado, las modernas tendencias que presenta la educación del ingeniero mechatrónico están orientadas a la formación de

personas con excelentes habilidades para resolver problemas de ingeniería bajo un trabajo en un equipo multicultural, o habilidades para formar líderes de proyectos con una alta integración tecnológica [3]. La tendencia muestra una educación para desarrollar un nuevo punto de vista enfocado en la creación de nuevos negocios derivados de los productos obtenidos de las investigaciones aplicadas. Se requieren nuevos ingenieros con una visión integral que responda a los cambios tecnológicos y a las necesidades sociales de nuestros tiempos. Existen pocas universidades en el mundo que definen el liderazgo o las nuevas tendencias de sus programas educativos.

Al observar con atención el desarrollo de la Ingeniería Mecatrónica desde sus inicios y hasta el presente, se puede comprender el cómo ha evolucionado y se ha convertido en lo que se conoce ahora. Se puede definir la mecatrónica como una disciplina que involucra de manera sinérgica los sistemas mecánicos, eléctricos y de control con las ciencias computacionales con el fin de producir elementos “inteligentes” y de muy altas prestaciones. El término “elementos “inteligentes” se refiere a dispositivos que tienen la capacidad de interactuar con el ambiente por medio de sensores y actuadores controlados por una plataforma de cómputo que se compone de hardware y software específico a la aplicación objetivo y que todo en su conjunto funciona sinérgicamente. Trascender la frontera de conocimiento de cada una de las disciplinas de que se compone la ingeniería mechatrónica no puede realizarse de manera simplista o en la fundación de un nuevo campo cognitivo [3]. Sin embargo, la transición de las distintas nociones disciplinares (de que se compone la mechatrónica) a la compleja sinergia transdisciplinaria que se obtiene en el producto final, debe forjarse como parte de un proceso transcultural, consistente en una transformación de hábitos prácticos y cognitivos que imprescindiblemente involucra una alteración profunda del proceso de enseñanza – aprendizaje.

Con esto se puede decir que la ingeniería mechatrónica es un estado sublime con respecto a las otras ingenierías, por así decirlo. Esto en el sentido de que la ingeniería mechatrónica “es lo más elevado en su género”, según definición de la RAE.

Existen diversas propuestas gráficas para ejemplificar a la Ingeniería Mecatrónica. La Figura 1, muestra una manera de varias posibles de representar la Ingeniería Mecatrónica [4].

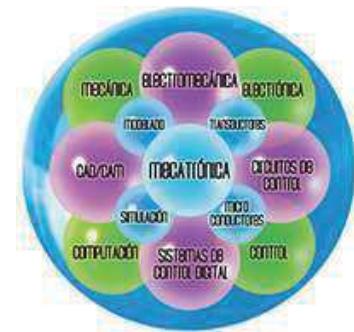


Fig. 1. Forma gráfica de ejemplificar la mechatrónica

3. Los ciclos propedéuticos y el currículo holístico

Las exigencias de los nuevos programas educativos existen en función de la demanda que ejerce la industria. En relación a esta demanda es cómo se planea cubrir las necesidades del mercado laboral de los futuros profesionales en mechatrónica. Por otro lado, existe otra demanda oculta o que pocas veces es tan obvia, y más que demanda es una necesidad de los estudiantes de obtener la capacitación y habilitación oportuna en las competencias relacionadas con su profesión y necesarias para ejercer lo más pronto posible. Tal es la razón de que algunas universidades desarrollaron currículos de Ingeniería Mecatrónica por ciclos propedéuticos [4]. El currículo por ciclos propedéuticos es una práctica que ha funcionado en otros países como Colombia y que en nuestro país ya se ha implementado tanto en carreras de licenciatura como de ingeniería que pertenece a universidades públicas y privadas, pero principalmente se ha desarrollado con mayor éxito en las universidades tecnológicas. La educación por ciclos propedéuticos consiste en diseñar un plan de estudios capaz de formar a los futuros profesionistas de manera integral y basada en certificaciones que acreditan en los estudiantes conocimientos a nivel de técnico, de tecnólogo o de ingeniero según dé seguimiento a sus estudios (a diferencia de la educación formal que ofrecen la mayoría de las universidades, que obedece a un patrón secuencial y de cierta manera particionado en segmentos independientes que difícilmente guardan alguna conexión con otras disciplinas del conocimiento).

Un currículo vertical genera resultados poco efectivos y no por que los contenidos sean de inutilidad para el aspirante a profesional, sino más bien, porque están comparto-mentalizados, aislados y sin alguna relación aparente o en ciertos casos la relación que guardan es demasiado sutil como para

que la generalidad de los estudiantes lo logre percibir. Un ejemplo puede ser el contenido de la asignatura de cálculo en la que se emplea como facilitador a un matemático puro. La lógica dicta que el matemático es la persona ideal para guiar el contenido de la asignatura, más esto depende directamente de la habilidad del profesor para relacionar los contenidos de su asignatura con la carrera o profesión futura de sus estudiantes. De esta manera es posible elaborar un currículo basado en ciclos propedéuticos, siempre y cuando, éstos se diseñen en base al modelo holístico que aporte a la enseñanza de la mecatrónica los elementos necesarios en donde los estudiantes adquieran competencias suficientes para desarrollarse en su campo laboral sin sacrificar la calidad de los contenidos. La Figura 2 muestra un ejemplo de los ciclos propedéuticos.

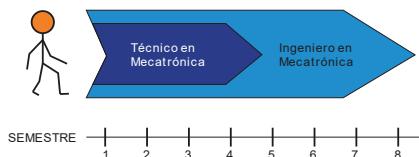


Fig. 2. Ciclos propedéuticos

La aplicación del modelo holístico en la educación permitirá que los estudiantes comiencen a aplicar sus conocimientos adquiridos en asignaturas de las ciencias básicas como son: matemáticas, cálculo, física, álgebra vectorial e incluso tecnologías de información e involucrarse en proyectos de integración que permitan el diseño, implementación y puesta en marcha de un sistema mecatrónico, motivando el interés de los estudiantes por la investigación aplicada, acercándolo al conocimiento de problemas reales [5]. Un ejemplo de este proceso puede verse en [6], en dicho trabajo se desarrolla un robot cartesiano con alta integración mechatrónica, y en donde para su concepción, se usaron diversas herramientas matemáticas, computacionales y de fabricación avanzada. La Figura 3 muestra dicho robot controlado por una Ipad.



Fig.3. Prototipo mechatrónico

El modelo holístico se relaciona con desarrollar un proceso de formación del profesional que consolide un paradigma educativo productivo, creativo e innovador, en contraposición con el informativo, vigente esencialmente en la enseñanza clásica, que deberá proporcionar la participación activa de estudiantes y profesores en su vínculo con los nuevos enfoques y desarrollo de la producción y los servicios teniendo en cuenta experiencias regionales e internacionales. Esto implica un profundo análisis, no sólo de las concepciones, sino de las condiciones reales de cómo implementar y ejecutar dicho proceso para lograr un cambio efectivo [7]. La Figura 4 muestra una propuesta del modelo holístico en la enseñanza de los lenguajes de programación.

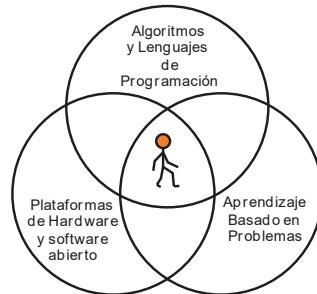


Fig. 4. Enseñanza de los lenguajes de programación basada en el modelo holístico de la educación.

En consecuencia, para el caso de la sinergia, el todo no se explica por las partes, se manifiesta a través de las partes: el todo recibe significado de las partes insertas en él. Esto es, cierta parte de una totalidad o de un sistema tiene significación distinta si está aislada o integrada a otra totalidad, ya que sus funciones dentro de otro sistema le confieren cualidades diferentes [8]. Esto es la esencia de la filosófica holística que se asemeja al concepto de sinergia que pocas palabras significa “acción de diversos elementos, de tal forma que cuando estos actúan conjuntamente, el rendimiento es mayor que la suma de los rendimientos de cada uno de ellos actuando por separado”.

4. Diseño holístico de la asignatura Lenguajes de Programación

Las estrategias de E-A deben tomar en cuenta múltiples dimensiones y múltiples niveles [9]. Por ello, hay que considerar los campos de acción de la mechatrónica. La Figura 1 muestra los campos disciplinarios de que se compone la Ingeniería Mecatrónica. Sin embargo, hay que ver más allá de lo que aparenta ser evidente. Es necesario entonces, cambiar la perspectiva desde donde se observa la

mecatrónica y destacar la sinergia de las disciplinas involucradas en un sentido holístico (ver Figura 5).

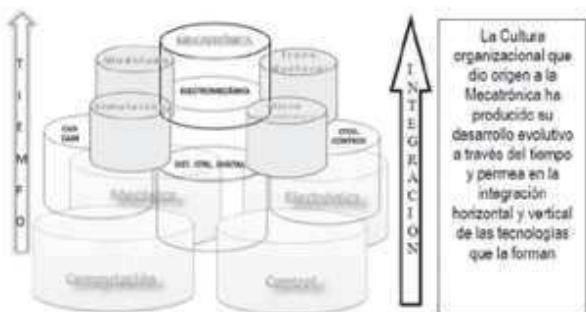


Fig. 5. Evolución de los campos disciplinares que dieron origen a la mecatrónica mencionado en [8].

El currículo es un término que se refiere a qué y cómo enseñar [10], por lo tanto, es necesario tener en cuenta las competencias que se deben desarrollar a los estudiantes de Ingeniería Mecatrónica en la asignatura de Lenguajes de Programación. Más adelante se presentarán propuestas de “cómo enseñar” dicha asignatura. Sin embargo, es importante aclarar que los diferentes modelos y teorías educativas distan de ser una receta que se sigue al pie de la letra. Por ésta razón, es necesario abordar los distintos momentos que se dan en el proceso educativo [11], dichos momentos, desde el punto de vista computacional, coinciden con algunas de las fases del ciclo de vida de un sistema (análisis, diseño, implementación y retroalimentación).

4.1 Análisis contextual del contenido de la asignatura de Lenguajes de Programación.

La asignatura Lenguajes de Programación se asemeja a las matemáticas, ya que son de naturaleza abstracta. Sin embargo, al igual que las matemáticas, la programación debe abordarse con un sentido útil y práctico, sin que los temas propuestos en su contenido queden aislados y carezcan de utilidad o aplicación en los sistemas mechatrónicos. Es importante diseñar instrumentos de aprendizaje basado en problemas propios de la profesión e incluso expresarlos en un prototipo que integre todos los saberes de la Ingeniería Mecatrónica sin importar el nivel alcanzado que obviamente en los primeros semestres resultará pobre (desde el punto de vista profesional) pero que de forma metódica proporciona a los estudiantes conocimiento y experiencia valiosísima para futuros trabajos. Un ejemplo de este caso se presenta en [6].

Existen varias alternativas para echar mano de los recursos didácticos apropiados para la enseñanza

de la programación. El orden de los ejes temáticos propuesto no exime la posibilidad de realizar proyectos desde los primeros pasos del aprendizaje, ya que incluso para Scratch, existe una versión habilitada para el uso de un Arduino. No se abordarán en este artículo la totalidad de los recursos didácticos expuestos en la Figura 6, simple y sencillamente porque son elementos importantísimos para la enseñanza de los lenguajes de programación y merecen un tratamiento especial ya que son base en trabajos existentes y para proyectos futuros.

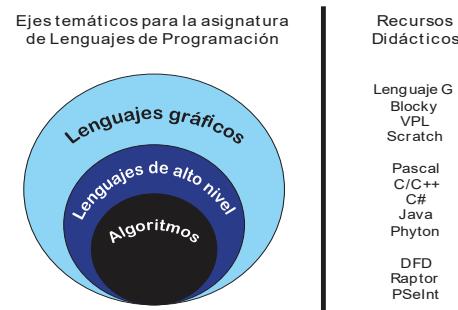


Fig. 6. Relación entre temas y recursos en la enseñanza de Lenguajes de Programación

4.2 Diseño e implementación de los temas centrales.

Es posible diseñar e implementar contenidos secuenciados, estructurados y alineados con las competencias que se deben lograr en los estudiantes de Lenguajes de Programación. Sin embargo, establecer relaciones entre otras asignaturas (disciplinas), es una historia distinta. Para lograr una transversalidad de contenidos en las asignaturas, es menester de las academias pero sobre todo la comunicación entre las distintas academias de que se compone el PE de Ingeniería Mecatrónica. Si la esencia de la mechatrónica es la interdisciplinariidad, entonces es responsabilidad de los académicos predicar con el ejemplo. Será necesario entonces conciliar proyectos, planes y prácticas de laboratorio que involucren múltiples disciplinas y con esto, ir llevando de la mano a los estudiantes con el fin de que alcancen ese tan preciado aprendizaje significativo.

En este artículo, se proponen que los temas regulares de la asignatura de Lenguajes de Programación sean exactamente los mismos, sólo es necesario cambiar el enfoque de las prácticas, problemas y proyectos que se resuelven (ver Tabla 1).

Tema	Plataforma	
	Software	Hardware

Algoritmos	DFD, Raptor, PSeInt, Scratch, Blockly	Arduino
Programación estructurada	CodeBlock C/C++, Borland C++, DevC/C++, Visual MRDS, LabView.	Arduino Raspberry Pic/DSPic Lego MindStorm C#,

Tabla 1. Relación entre temas, software y hardware propuestos.

Se recomienda para facilitar el aprendizaje de los algoritmos al estudiante, que se utilicen lenguajes didácticos exclusivos para este propósito como lo son: DFD (Data Flow Diagram), Raptor o PSeInt. También existen lenguajes gráficos como Blockly y Scratch que cumplen perfectamente con el propósito de introducir al estudiante a los conceptos básicos de la programación y al pensamiento algorítmico de manera gráfica. Cabe mencionar que Scratch es una plataforma gráfica para el desarrollo de algoritmos, similar a un rompecabezas en donde las estructuras secuenciales, condicionales y cíclicas se unen para formar luego una secuencia lógica de instrucciones que resuelven un problema específico, y que, gracias a su sencilla e intuitiva interfaz (GUI), es la plataforma más utilizada para promover el aprendizaje de la programación y la algoritmia en casi todos los niveles educativos, desde primaria y secundaria, incluso hasta nivel universitario ya que existe una versión S4A (Scratch for Arduino) capaz de controlar un Arduino y ser utilizado como una tarjeta de adquisición de datos y/o elemento de control para actuadores (ver Figura 7).

Es prioritario el aprendizaje del lenguaje de programación C/C++, ya sea en entornos de desarrollo (IDE) para MS-Windows o Linux. CodeBlocks es una buena opción, ya que este libra la compatibilidad en ambos sistemas operativos. La ventaja de trabajar con IDE's compatibles con Linux es que hacen más simple la transición de una plataforma de desarrollo a otra. Raspberry Pi (ver Figura 8) es una plataforma de desarrollo completa, una minicomputadora capaz de correr un sistema operativo por sí misma y además tiene la ventaja de contar con un puerto GPIO que la hace versátil al momento de diseñar cualquier prototipo mecatrónico. Existen plataformas comerciales distintas a las presentadas con anterioridad, pero poco asequibles a los estudiantes. La diferencia de precios entre unos y otros es considerable si se toma en cuenta que una tarjeta Arduino cuesta entre 20 y 30 U.S. dlls., un Raspberry Pi esta alrededor de los 45 y 60 U.S. dlls; mientras que las plataformas comerciales están entre los 400 y 1500 U.S. dlls.



Fig. 7. Shield para control de motores DC y servos acoplado a una tarjeta Arduino

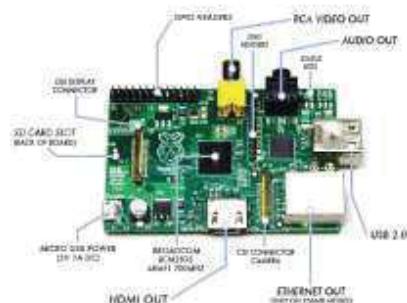


Fig. 8. Plataforma de cómputo Raspberry Pi

4.3 Evaluación de las competencias adquiridas.

Es necesario establecer las políticas de evaluación de las evidencias de desempeño en el desarrollo de los proyectos, resolución de problemas y competencias adquiridas de forma integral. Todo lo anterior va en relación a los resultados de aprendizaje que se pretenden obtener de los estudiantes. En el caso de la asignatura de Lenguajes de Programación, los resultados de aprendizaje deben ser: capacidad de análisis y un planteamiento efectivo de los problemas relacionados a la mecatrónica, destrezas en el diseño e integración de los distintos componentes de un sistema mecatrónico (desde el nivel conceptual hasta su ejecución); habilidades para la puesta en marcha de la solución, depuración y mantenimiento. Una excelente fuente de inspiración para los estudiantes de mecatrónica son los eventos extramuros y los concursos de robótica como: pelea de sumobots, sigue líneas, escaladores, luchadores, entre otros, y en los que se pone a prueba todos y cada uno de los resultados de aprendizaje antes mencionado e incluso, otros de igual importancia como lo son: la capacidad de trabajo en equipo, la disciplina, el orden, el liderazgo y sobre todo el espíritu de competencia, entre otros.

5. Conclusiones

Actualmente se están proponiendo cambios en el diseño curricular del PE de Ingeniería Mecatrónica de la Universidad Estatal de Sonora (UES). Los logros alcanzados en la inclusión de una metodología apropiada y el rediseño del plan de estudios, se medirán en el sentido de que se le dé una verdadera importancia a las ciencias computacionales y en específico a la algoritmia. Sin embargo, hay mucho camino por recorrer y paradigmas por construir. La Ingeniería Mecatrónica es la filosofía de la sinergia convertida en ciencia aplicada de forma integral y holística. La importancia de aplicar el modelo holístico en la preparación de nuestros estudiantes radica principalmente en que en el transcurso de la formación del ingeniero, se obtienen resultados integrales en relación con las competencias adquiridas y ciertamente el alumno tiene la oportunidad de poner en práctica sus conocimientos integrando proyectos mecatrónicos de todas las medidas y alcances.

Dichos proyectos actualmente se proponen en el currículo de la carrera de Ingeniería Mecatrónica de la UES, principalmente en el caso de la programación de computadoras, en donde se utilizan tecnologías actualizadas y se toman en cuenta hardware y software tanto propietario como abierto; tal es el caso del hardware: Arduino y Raspberry Pi; y software como: CodeBlocks C/C++, Scratch, LabVIEW ®. En la Universidad La Salle Noroeste se usan diversos paquetes computacionales para asistir las materias de la carrera de Ingeniería Mecatrónica, como por ejemplo, las plataformas Cocoa, X-CODE y OBJETIVE-C, Solidworks2014 y MatLAB, entre otros, con su respectivo hardware propietario. Es fácil inferir que la manera en que se enseña la mecatrónica no puede ser la convencional. Es decir, las estrategias para enseñar química, física o matemáticas no pueden ni deben ser las mismas para la enseñanza de la Ingeniería Mecatrónica. En consecuencia, la enseñanza de las disciplinas de que se compone la mecatrónica debe trascender y formar vínculos hasta lograr la transdisciplinariedad de la misma manera en que los sistemas mecatrónicos guardan esa sinergia entre sus componentes.

Referencias

- [1] Vargas E. Metodología aplicada al desarrollo de máquinas mecatrónicas. Congreso Latinoamericano de Instrumentación y Control de Procesos Universidad Autónoma de Querétaro. México. (2012)
- [2] Reyes F. Cid J., Vargas E. "Mecatrónica: Control y Automatización", Editorial: Alfaomega Grupo Editor S.A. de C.V., 1º edición, México, ISBN: 9786077075486, 2013.
- [3] Gutiérrez M., "La teoría del aprendizaje significativo en la perspectiva de la teoría cognitiva", Octaedro, España, 1ra. Edición, 2008.
- [4] Vargas E., Aceves M., Sotomayor A., Tovar S., Rodríguez J. Perspectivas de la Ingeniería Mecatrónica. *VIII Congreso Internacional de Ingeniería Mecánica y Mecatrónica*. Abril 10-12 y Mayo 29-31, Puerto Vallarta, Jalisco. 2014.
- [5] Arveláez O., Mendoza J. "La ingeniería mecatrónica por ciclos en Colombia", Scientia et Technica, Universidad Tecnológica de Pereira, XIII, No 35, páginas 421-426, 2007.
- [6] Rodríguez M., Ricardo L.. "El Modelo Holístico para el proceso de Enseñanza-Aprendizaje de geometría en Arquitectos de la escuela cubana", *Revista Latinoamericana de Investigación en Matemática Educativa*, 10 (3), páginas 421-481, 2007.
- [7] Jiménez E., Ruiz A., Meléndez J., Pereo A., Carbajal I., Martínez V., López F., Luna G. *Integración mecatrónica en el desarrollo de un robot cartesiano*. 13º Congreso Nacional de Mecatrónica. Octubre 30 a Noviembre 1. Querétaro, Qro. (Por presentarse). 2014.
- [8] Nieto M. "El papel de las ciencias básicas en la enseñanza de la ingeniería", 1er. Congreso de la enseñanza de la ingeniería, Quetzaltenango 2004.
- [9] Aquino J., Corona L., Trujillo J "Tendencia en la enseñanza de la Ingeniería Mecatrónica y su campo disciplinar", Colección Ciencia y Tecnología. Universidad de Palermo, Volumen XIII, Número 13, p. 233-250 2013.
- [10] Castellanos J., Lara L., Lapido M., Iglesias M., Balbis E., Navales M., Alvarez A. "Diseño Conceptual para la formación Holística del Ingeniero Mecánico", Biblioteca Virtual Universidad de Cienfuegos, URL: http://www.bibliociencias.cu/gsdl/collect/evento_s/import/Formacion_holistica_ingeniero.pdf. Consultado julio de 2014.
- [11] Díaz F. "Aproximaciones metodológicas al diseño curricular hacia una propuesta integral", *Tecnología y comunicación educativa*. ILCE, Número 21, 1993.

Controlador PID de una Celda Termoeléctrica: Sintonización por medio de un Algoritmo Evolutivo

García Mejía Juan Fernando, Carlos Eduardo Torres Reyes y José Antonio García Mejía

Universidad Autónoma del Estado de México
 Centro Universitario UAEM Atlacomulco
 Carretera Toluca Atlacomulco km 60
 Atlacomulco Edo. De Méx

Resumen

Una celda termoeléctrica es un dispositivo semiconductor de dos capas que basa su funcionamiento en los efectos Thompson y Seebeck, por lo tanto es factible su uso en la refrigeración móvil, con esto en mente en este trabajo se documenta el diseño de un controlador PID para regulación de temperatura cuyos parámetros proporcional, integral y derivativo se calculan se obtienen mediante una técnica de inteligencia de enjambre denominada enjambre de partículas con codificación real, lo que permite la reducción del error cuadrático medio. Cabe destacar que la presente propuesta se basa en Scilab un software de licencia GNU

Palabras clave: Celda Termoelectrica, enjambre de particula, Scilab

1. Introducción

Una celda termoeléctrica (TEC, por sus siglas en inglés) es un dispositivo semiconductor que tiene la capacidad de generar calor ó frio en función de la magnitud y polaridad, de una corriente eléctrica de alimentación. Las aplicaciones de los TEC se encuentran en micro refrigeración y refrigeradores móviles; presentan ventajas como su tamaño reducido, así como largo tiempo de vida útil, amigable con el ambiente [8] Una celda termoeléctrica de manera típica está caracterizada por la función de transferencia descrita en (1) [6]

$$\tilde{T}_L(s) = G_1(s)\tilde{I}(s) + G_Q(s)\tilde{Q}_L(s) + G_a(s)\tilde{T}_a(s) \quad (2)$$

Donde

$\tilde{T}_L(s)$ representa la temperatura de la cara de enfriamiento

$\tilde{I}(s)$ es la corriente de entrada
 $\tilde{T}_a(s)$ es la temperatura ambiente

Además

$$G_I(s) = \frac{N(s)}{sD(s)} \quad G_a(s) = \frac{AA_F h k q}{D(s)}$$

$$G_Q(s) = \frac{E_H \sinh(qL) + Akq \cosh(qL)}{D(s)}$$

Donde

$$N(s) = \{Akq[\alpha_L \bar{T}_L \cosh(qL) - \alpha_H \bar{T}_H] + \alpha_L \bar{T}_L E_H \sinh(qL)\}s + \frac{Akq\beta}{C\gamma} [E_H(1 - \cosh(pL)) - Akpsinh(pL)]$$

y

$$D(s) = AkqE_L \cosh(qL) + E_H E_L \sinh(qL) + AkqE_H \cosh(pL) + A^2 k^2 p q \sinh(pL)$$

El modelo matemático se reduce a la expresión mostrada en (2) dado que se considera a la temperatura ambiente como constante. Al aplicar constantes especificadas en [6] se obtiene (2), cuya respuesta a una entrada de 1.55 ampers se muestra en la figura 1, se puede observar que el tiempo de establecimiento, es decir donde no se presentan cambios en el eje del tiempo t es alrededor de 325 segundos, en un valor de $-9.9^\circ C$ en el eje y , que representa la temperatura de enfriamiento de la TEC, con una corriente de entrada de 1.55 Ampers

$$\tilde{T}_L(s) = G_I(s)\tilde{I}(s) \quad (2)$$

$$G(s) = -6.4061 \left(\frac{0.064s + 0.00854}{s^2 + 0.5964s + 0.00855} \right) \quad (3)$$

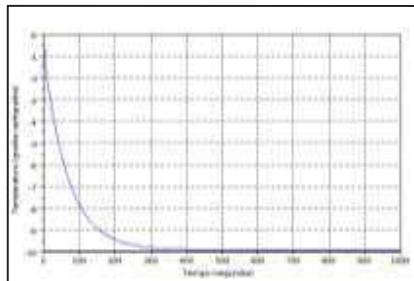


Fig. 1. Respuesta de la Celda Termoeléctrica

El desempeño de la TEC mostrado en la figura 1 puede mejorarse mediante el uso de técnicas que se encuentran enmarcadas dentro de la ingeniería de control, la cual se encarga del estudio de esquemas que manipulen las respuestas de fenómenos físicos mediante actuadores. Uno de los esquemas más usados en la ingeniería de control se denomina controlador Proporcional, Integral, Derivativo (PID) cuya expresión matemática se muestra en la ecuación 4. La expresión matemática del TEC controlado por medio de un PID se muestra en la ecuación 5, donde $R(s)$ es la referencia de temperatura que se desea obtener

$$C(s) = k_p + \frac{k_i}{s} + k_d s = \frac{k_p s + k_i + k_d s^2}{s} \quad (4)$$

$$\tilde{T}_L(s) = \frac{G_I(s)C(s)}{1 + G_I(s)C(s)} R(s) \quad (5)$$

$$\tilde{T}_L(s) = \frac{G_I(s)C(s)}{1 + G_I(s)C(s)} R(s) \quad (6)$$

Como puede deducirse de las expresiones anteriores, $\tilde{T}_L(s)$ depende de los valores k_p, k_i, k_d , los cuales se calculan de forma habitual por medio realiza por medio del criterio de Zigler-Nichols. En este trabajo se propone como alternativa a este un enfoque basado en optimización numérica, el cual se abordara en secciones subsecuentes

2. Fundamentación teórica

La optimización es una de las áreas de la ingeniería, es un conjunto de ideas que encuentran el mejor resultado bajo ciertas consideraciones denominadas restricciones [2]. Un caso particular de esta es la optimización numérica la cual es un proceso donde el óptimo global (valor máximo o mínimo) de una función objetivo es buscado mediante la elección de una combinación de variables, esto se puede expresar en (7). [5]

$$\min(\max) f(x), x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n \quad (7)$$

Las restricciones en un problema se suelen expresar como

$$\begin{aligned} g_j(x) &\leq 0, & j &= 1, 2, \dots, m \\ h_j(x) &= 0, & j &= 1, 2, \dots, r \end{aligned}$$

Donde $f(x), g_j(x)$ y $h_j(x)$ son funciones escalares del vector x . Los componentes de $x = [x_1, x_2, \dots, x_n]^T$ son denominados variables, $f(x)$ es la función objetivo, $g_j(x)$ y $h_j(x)$ son funciones que describen las condiciones de inequidad e igualdad respectivamente. El vector óptimo x que resuelve la expresión (7) se denota por x^* con el correspondiente valor de optimización $f(x^*)$. Las posibles formas de resolver el problema general descrito en (7) son las siguientes [7]:

- i. analíticamente
- ii. emulándolo y midiendo físicamente las variables de interés
- iii. resolviéndolo mediante técnicas computacionales, como los algoritmos evolutivos

Los algoritmos evolutivos (EA, por sus siglas en inglés) son series de pasos definidos con aplicación en la optimización o de aprendizaje con la capacidad de evolucionar. Estos tienen tres características principales [1]:

- i. Basado en la población. Un EA mantienen un conjunto de soluciones, llamada población.
- ii. Fitness-orientada. Todas las soluciones de una población se llama individuo, tiene una representación llamada código y la evaluación de desempeño se denomina aptitud (o fitness).
- iii. Variación de motor. Los individuos de una población que se someten a una serie de operaciones con la finalidad introducir variaciones en su código.

Un grupo especial de algoritmos evolutivos es La inteligencia de enjambre (SI, por sus siglas en inglés) es una propiedad que poseen algunos sistemas biológicos constituidos por individuos sencillos — los bancos de peces, las colonias de hormigas o las bandadas de aves — que tienen un comportamiento colectivo. Estos modelos biológicos se estudian con la finalidad de generar técnicas computacionales que

permitan la solución de problemas de optimización numérica o combinatoria, lo cual fue concretado en los trabajos de Marco Dorigo que desarrollo la optimización por colonia de hormigas (ACO, por sus siglas en inglés) y de James Kennedy y Russell Eberhart que propusieron la optimización por enjambre de partículas (PSO, por sus siglas en inglés) [3].

Con base a lo anterior, es posible considerar a la inteligencia de enjambres como un conjunto de métodos estocásticos basados en poblaciones o un enfoque meta-heurístico para resolver problemas de optimización

La optimización por enjambre de partículas (PSO) es un método de optimización global donde las soluciones de un determinado problema se representan como puntos en un espacio N dimensional. Cada partícula se representa por un vector de posición y un vector de velocidad. Fue propuesto en 1995 y se basa en un modelo psicosocial de aprendizaje social e influencia. Computacionalmente un enjambre es un conjunto de agentes móviles que se comunican entre sí actuando sobre su entorno local. Los individuos (partículas) de un enjambre tienen un comportamiento simple el cual consiste en emular el éxito de los individuos vecinos. El comportamiento colectivo del conjunto de partículas se refleja en el descubrimiento de regiones óptimas en un espacio de búsqueda de alta dimensión [4]

3. Método

El algoritmo desarrollado para resolver el problema de optimización mencionado en este trabajo, utilizando un enjambre de partículas se muestra a continuación

1. Se genera la población inicial de manera aleatoria, para los valores de K_p, K_i, K_d (posición de partículas) distribuidos a lo largo del espacio de búsqueda de manera uniforme.
2. Generar los vectores de velocidad correspondientes a K_p, K_i, K_d
3. Con esta población de controladores se evalúan la función objeto que se considera antígeno
4. Actualizar velocidad mediante la siguiente formula

$$v_i(k+1) = \gamma v_i(k) + c_1 r_i(pbest - p_i(k)) + c_2 r_2(gbest(k) - p_i(k)) \quad (1)$$

Donde

v_i es velocidad de partículas
 γ es aceleración, que decrece conforme se incrementan las generaciones
 c_1 es constante [0,1]
 r_i es constante [0,1]
 $pbest$ es mejor local
 p_i es posición actual de partícula
 c_2 es constante [0,1]
 r_2 es constante [0,1]
 $gbest(k)$ es el mejor global
 k generación

5. Actualizar posición de partícula

$$p_i(k+1) = p_i + v_i(k+1) \quad (2)$$

6. Obtener $pbest$ y $gbest(k)$
7. Verificar criterio de terminación, si no saltar a 3

Para obtener la función objetivo del presente trabajo se retoman las expresiones (5) y (6) se puede observar que la salida $\tilde{T}_L(s)$ depende de los valores k_p, k_i, k_d . Con base a la definición de error propia de los sistemas de control se tiene

$$e(s) = R(s) - \tilde{T}_L(s) \quad (8)$$

De (8) puede derivarse una función objetivo que se muestra en (9)

$$f_{obj}(k_p, k_i, k_d) = \max \left(\frac{1}{1 + \sqrt{\frac{1}{T} \int_0^T e(k_p, k_i, k_d)^2}} \right) \quad (9)$$

4. Resultados

Al similar el comportamiento de la celda termoeléctrica en Scilab y ejecutar el algoritmo propuesto se obtuvieron los siguientes valores del error cuadrático medio así como de las ganancias k_p, k_i, k_d (Tabla 1 y figura 3), por otra parte la figura 2 muestra el ajuste por medio del criterio de Zigler-Nichols

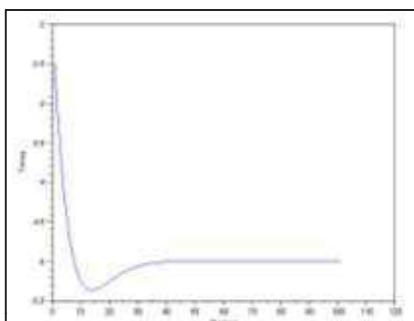


Fig. 2 PID sintonizado por ajuste de Zigler-Nichols

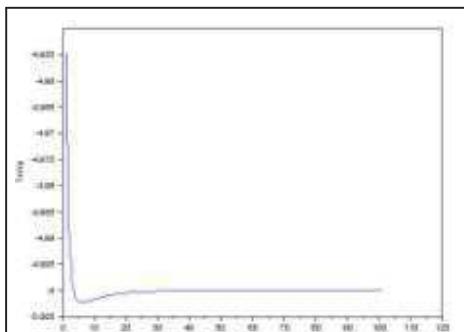


Fig. 3 PID sintonizado por un enjambre de partículas

Tabla 1 Resultados

Parámetro	Ajuste de Zigler-Nichols	Ajuste basado por enjambre de partículas
k_p, k_i, k_d	-2.4, -0.6-2.4	- 19.9, - 21.2, - 8.2
Tiempo de establecimiento	40seg	25seg
Valor de sobre impulso	-5.4	-5.004
Valor error	0.3379797	0.066

5. Conclusiones

Como los muestran los resultados mostrados en la tabla 1 se observa que el tiempo de establecimiento, así como el error cuadrático medio se reducen, en relación al ajuste de Zigler-Nichols, este ultimo parámetro es el objetivo del algoritmo genético, se observa el sobre impulso presenta un incremento, el cual puede ser disminuido mediante el uso de restricciones o convirtiendo el algoritmo de uno

simple a uno de múltiples objetivos, se observa que la convergencia del algoritmo es alta. Con esto en mente es posible afirmar que el ajuste de parámetros de un controlador PID es más efectivo mediante técnicas de optimización basada en inteligencia de enjambre en comparación con métodos analíticos como el ajuste de Nichols. Además la codificación real permite una manipulación de los datos de manera más exacta en relación a codificaciones binarias

Referencias

- [1] Burger, C.. “Propeller performance analysis and multidisciplinary optimization using a genetic algorithm.” Auburn University. 2007
- [2] Chow Lee K. “The success of information security projets: An investigation of the projet management process, projet cost, project risks and user acceptance 2008”
- [3] Duarte Muñoz Abraham “Metaheurísticas” Librería-Editorial Dykinson, 2007
- [4] Muñoz Pérez José “Inteligencia computacional inspirada en la vida” Servicio Publicaciones UMA, 2010
- [5] Ponstein J.P “Approaches To The Theory Of Optimization” Cambridge University Press 2004
- [6] Song Shaojing Temperature “Control of Thermolectric cooler Based on Adaptive NN-PID” International Conference on Electrical and Control Engineering 2010
- [7] Snyman Jan A. “Practical Mathematical Optimization: An Introduction To Basic Optimization Theory and Classical And New Gradient-Based Algorithms”. Birkhäuser 2005.
- [8] Tarter R., “Solid-state power conversion handbook”. United State of America: John Wiley and Sons 1993
- .

Diseño, Modelación y Simulación de un Brazo Robótico de cinco grados de libertad Utilizando LabVIEW y SolidWorks

Ramírez Escobar Raúl Alonso¹, Martínez Soto Luis Ángel¹, Vázquez Cuevas Ignacio Javier¹, Meranza Castillón Manuel Omar¹, González Ochoa Sara Bertha¹ y Jiménez López Eusebio²

¹Instituto Tecnológico de Nogales, raul_itnogales@outlook.com, luisangelmtz20@hotmail.com, ivazquezitn@gmail.com, manmeran@hotmail.com, sarabgo@hotmail.com

²Universidad La Salle Noroeste – CIAAM-UTS-IIIMM, ejimenezl@msn.com

Resumen

El presente trabajo describe los resultados de un proyecto realizado por estudiantes y profesores de la carrera de Ingeniería Mecatrónica del Instituto Tecnológico de Nogales (ITN). El proyecto consistió en el desarrollo de un simulador de un robot antropomorfo de 5 GDL. Se construyeron las ecuaciones cinemáticas de posicionamiento del robot usando la metodología de Denavit-Hartenberg. Con los modelos matemáticos obtenidos fueron planteados los modelos cinemáticos inversos y directo, respectivamente. Dicho modelo fue programado en el software LabVIEW. El robot fue dibujado en el paquete Solidworks y, posteriormente, sus partes fueron importadas a LabVIEW en donde se visualizan los movimientos del robot. Se construyó un prototipo con el objetivo de probar la funcionalidad el simulador.

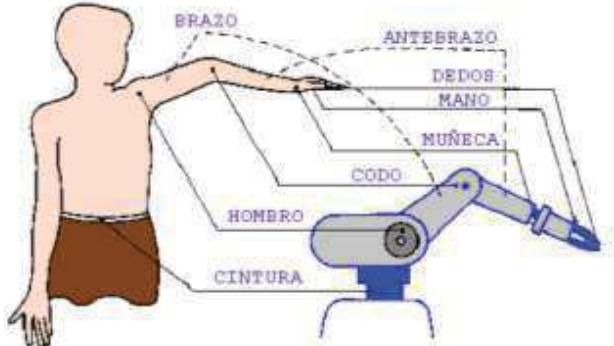
Palabras clave: Cinemática, Robots, LabVIEW, Solidworks.

1. Introducción

Un brazo manipulador o brazo robótico se puede definir como el conjunto de elementos electromecánicos que propician el movimiento de un elemento terminal (gripper o herramienta). La constitución física de la mayor parte de estos manipuladores guarda cierta similitud con la anatomía de las extremidades superiores del cuerpo humano, por lo que, en ocasiones, para hacer referencia a los distintos elementos que componen al robot, se usan términos como: cintura, hombro, brazo, codo, muñeca, etc. [1]. La Figura 1 muestra la similitud de un robot con un brazo humano.

La robótica ha abierto una nueva etapa en los procesos de mecanización y automatización de los procesos de producción, ya que gracias a ella se ha podido sustituir máquinas o sistemas capaces de automatizar operaciones concretas, por dispositivos mecánicos de uso general dotados de varios grados de libertad que son capaces de adaptarse a la automatización de un número muy variado de procesos u operaciones [2].

Fig. 1. Similitud entre un brazo humano y uno robótico



Una especificación general de un brazo robótico comprende: sus grados de libertad, su configuración y su cinemática directa e inversa. Estas especificaciones son dadas desde el diseño propio de cada robot y su aplicación. El grado de libertad de un sistema mecánico se define como un número entero correspondiente al número mínimo de coordenadas requeridas para especificar una configuración geométrica de un sistema dado. Para ubicar un cuerpo rígido en el espacio se necesitan seis grados de libertad, es decir, una traslación a lo largo de cada una de las tres direcciones del espacio y una rotación alrededor de un eje tridimensional [3].

Para la modelación cinemática y dinámica de multicuerpos rígidos, diversas herramientas y métodos se han utilizado. La metodología de Denavit-Hartenberg es un método de modelación de robots más popular y uno de los más utilizados [4]. Para el caso de la simulación computacional de robots y mecanismos se han utilizado diversos paquetes computacionales, entre ellos el Solidworks [5] y el LabVIEW [6]. En este artículo se describe el desarrollo de un simulador de un robot antropomorfo de 5GDL. Para la modelación cinemática se utilizó la metodología de Denavit-Hartenberg y para la simulación

computacional se usaron los paquetes Solidworks para el dibujado del prototipo y LabVIEW para la animación y el control.

2. Metodología de Denavit-Hartenberg

Según Denavit-Hartenberg se puede hacer un análisis de un multicuerpo utilizando únicamente 4 parámetros implícitos en la geometría del robot, estos parámetros son necesarios en la matriz de transformación homogénea la cual representa el movimiento de un eslabón, por lo tanto para representar el movimiento de las 5 articulaciones serán necesarias 5 matrices de transformación homogénea [4]. La matriz de transformación homogénea propuesta por Denavit – Hartenberg para la solución de este modelo se muestra a continuación [7]:

$$A_i^{i-1}(q_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

De esta matriz a_i representa lo largo del eslabón al que se esté haciendo referencia en el subíndice i, d_i es la distancia entre articulaciones, es decir, es cuando el eslabón no está unido exactamente a la base del motor, existe una distancia representada por este parámetro, si este perno o distancia está desfasado un determinado ángulo este se representa mediante α_i , finalmente θ_i representa el ángulo de giro de la articulación, es decir, el ángulo que gira el motor del eslabón i. Por ejemplo, supóngase que se conocen los parámetros de un eslabón de un robot. Entonces la matriz homogénea asociada es:

$$A_i^0 = \begin{bmatrix} \cos(\theta_i) & 0 & \sin(\theta_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & 0 & -\cos(\theta_i) & a_i \sin(\theta_i) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

La teoría de Denavit-Hartenberg sugiere que una vez que se tienen estas matrices la forma de obtener la cinemática directa para el total de los grados de libertad basta con hacer una post-multiplicación entre las matrices relacionadas con cada eslabón; es decir; multiplicar la primera por la segunda y al resultado de esto por la tercera y así sucesivamente hasta hacerlo para las cinco matrices si fuera el caso de un robot de 5 GDL [3]. El modelo final para un robot de 5 GDL antropomorfo es el siguiente:

$$T_n^0(q_i) = A_1^0 * A_2^1 * A_3^2 * A_4^3 * A_5^4 \quad (3)$$

3. Desarrollo del simulador

Con el propósito de motivar a los alumnos y profesores de la carrera de Ingeniería Mecatrónica del Instituto Tecnológico de Nogales al desarrollo de prototipos robóticos, se planteó un proyecto en el que fuera posible integrar diversas herramientas matemáticas y computacionales para concebir un robot didáctico y mostrar que la sinergia es una de las características del diseño de sistemas mecatrónicos. Un robot es un sistema mecatrónico, pues integra sistemas mecánicos, electrónicos y computacionales [4]. Después de algunas discusiones se propuso el diseño, la fabricación y el control de un robot de 5 GDL tipo antropomorfo como proyecto de arranque y que fuera la base para futuros trabajos de desarrollo e investigación. En este sentido los alumnos realizaron una búsqueda sobre robots y prototipos en las diferentes universidades. En el Instituto Tecnológico de Chihuahua fue desarrollado un brazo robótico SPIDER como el mostrado en la Figura 2 [8].

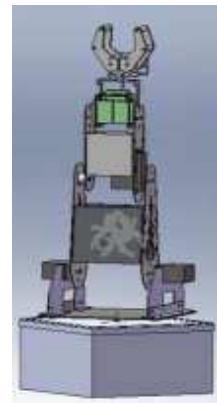


Fig. 2. Robot SPIDE de 5 GDL

El prototipo está basado en una configuración de un robot articulado de 5 GDL, cada pieza fue diseñada en SolidWorks® y al finalizar el modelado de las piezas se realizó un ensamblaje virtual para hacer las correcciones y modificaciones que fueran necesarias hasta que dieran un resultado satisfactorio. Una vez que se completó el ensamblaje de manera satisfactoria utilizando SolidWorks® se realizaron varias simulaciones como un análisis de colisiones entre piezas con el cual se definió la envolvente de trabajo de SPIDER, además se realizó un análisis de propiedades físicas de cada pieza (densidad, masa, área de superficie, momentos de inercia, etc.), esto con la finalidad de tener una clara idea del funcionamiento del diseño al ser construido y así evitar pérdidas de material y tiempo.

En el artículo científico llamado “Robot M5_EDU_HL” elaborado por Audevedes S.A. de C.V.; se describe el desarrollo de un robot M5_EDU_HL. Este robot es un manipulador didáctico de 5 grados de libertad y tiene diferentes tipos de terminales intercambiables que van desde un gripper hasta un trazador. Posee un entorno de programación amigable mostrando información sobre los

cálculos cinemáticos y un entorno de simulación en 3D que facilita e incrementa el aprendizaje de los robots manipuladores. Como parte opcional del brazo robótico, se le puede adosar al mismo una cámara digital (tipo webcam o similar) que permite implementar algoritmos de visión artificial (reconocimiento de figuras) para efectuar prácticas de robótica avanzada [9].



Fig. 3. Robot M5_EDU_HL

3. 1 Descripción del prototipo

Después de haber estudiado varias configuraciones de robots de 5 GDL se optó por generar uno propio. El prototipo fue dibujado en la plataforma Solidwork, como se muestra en la Figura 4.

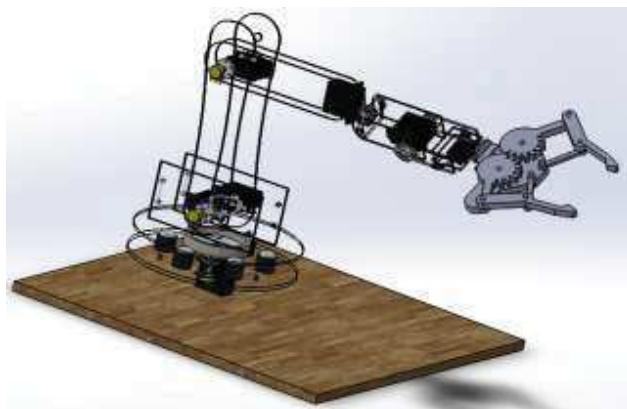


Fig.4. Ensamble mecánico del robot 5 GDL en Solidworks

La Figura 5 muestra la pantalla principal del simulador del robot considerado. Dicho simulador está desarrollado en la plataforma LabVIEW.

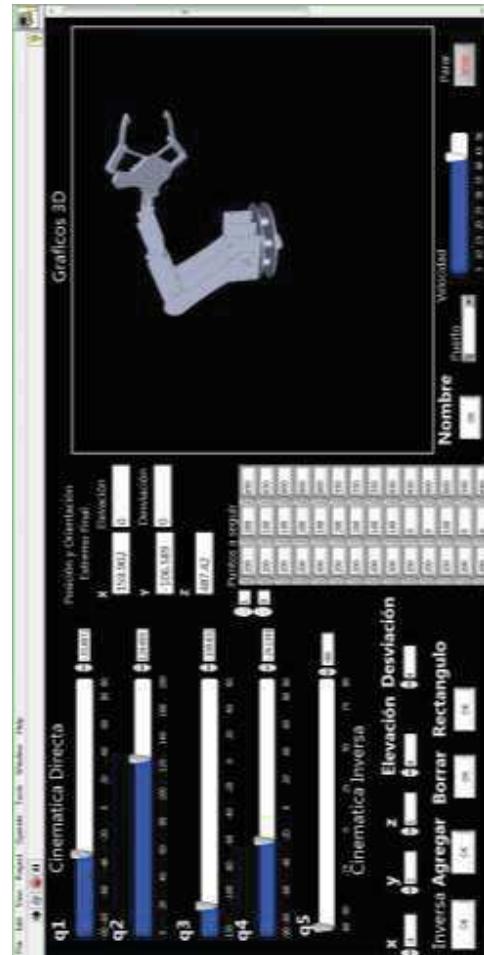


Fig. 5. Robot diseñado en el ITN.

El simulador permite controlar los movimientos de los eslabones que componen al robot. Dichos movimientos fueron modelados con la metodología de Denavit-Hartenberg usando los parámetros geométricos necesarios del robot. Con la información geométrica de los eslabones y usando las expresiones (1), (2) y (3) se conforma el modelo cinemático del robot por medio del cual se planteó los problemas cinemáticos directo e inverso, respectivamente. Para el caso del problema directo se conocen todos los ángulos que los eslabones deben usar para rotar y las incógnitas son la posición y la orientación en el espacio del efecto final del robot. Para el caso del problema inverso son conocidas la posición y orientación del órgano terminal y se desconocen los ángulos que los eslabones deben rotar. En la Figura 5 se observa que la pantalla principal del simulador está dividida en tres partes: 1) Parte donde se ingresan los datos para los problemas directo e inverso, respectivamente, 2) Parte donde se tiene o se puede grabar una matriz de puntos y 3) Parte

3.2. Descripción de los VI (Virtual Instruments)

El simulador mostrado en la Figura 5 fue hecho en LabVIEW. La construcción de dicho simulador se realiza por medio del diseño de instrumentos virtuales o VI. En esta sección se describirán algunos VI relacionados con el simulador del robot en estudio.

En la Figura 6 se muestra el VI relacionado con el cálculo de la cinemática directa e inversa. Este algoritmo opera a través de la diferencia entre el valor actual contra el valor deseado, el algoritmo arroja un incremento en cada uno de los puntos (X,Y,Z) para así ir agregando a la nueva posición. Este VI tiene programadas las ecuaciones cinemáticas generadas por la aplicación de la metodología de Denavit-Hartenberg. La Figura 7 muestra un VI el cual relaciona los dibujos del robot a través de primitivas con LabVIEW. Se realiza una llamada de cada uno de los sub-ensambles hechos en Solidworks para armar el robot y se crean objetos con el fin de sólo llamar una vez a todos y, posteriormente, se arma el robot y se transforma, es decir se rotan y se trasladan cada uno de los sub-ensambles y a través del desplazamiento de registros se recuerda la última transformación. Los VI mostrados en las Figuras 6 y 7 son dos de los más importantes en el desarrollo del simulador, pues representan el modelado cinemático y la construcción gráfica del robot.

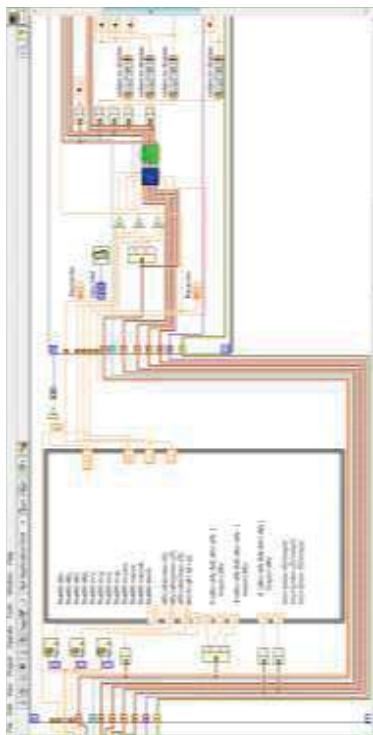


Fig. 6. VI del modelo matemático

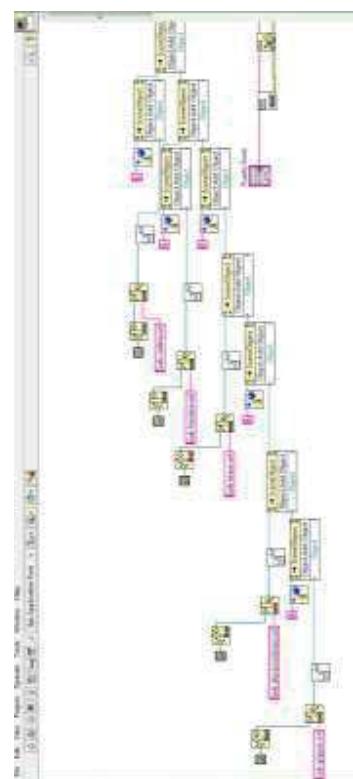


Fig. 7. VI de conexión de Solidworks

3.3. Prototipo

Para probar la funcionalidad del simulador se diseñó y construyó un prototipo del robot el cual se muestra en la Figura 8. Para mover los eslabones del robot se utilizaron servomotores de aeromodelismo. Se utilizó Acrílico para la fabricación de los eslabones y se colocó un mecanismo de agarre mecánico en el extremo final del robot. Cabe mencionar que se buscaba que el prototipo fuera de bajo costo y, en consecuencia, los materiales debían ser baratos.



Fig. 8. Prototipo del robot

En la Figura 9 se muestra el VI relacionado con la comunicación entre LabVIEW y los servomotores. En dicho VI se observa la comunicación serial con una tarjeta arduino desde LabVIEW. Se realizó un protocolo de comunicación entre LabVIEW y arduino para mover cada uno de los servomotores dependiendo de valor enviado en LabVIEW. Las pruebas realizadas en el prototipo mostraron que el simulador fue representativo y que el robot efectivamente respondía a los movimientos virtuales de los eslabones en el simulador. Esta representatividad física de los movimientos del robot permite determinar la eficiencia del modelado matemático y de la metodología de Denavit-Hartenberg.

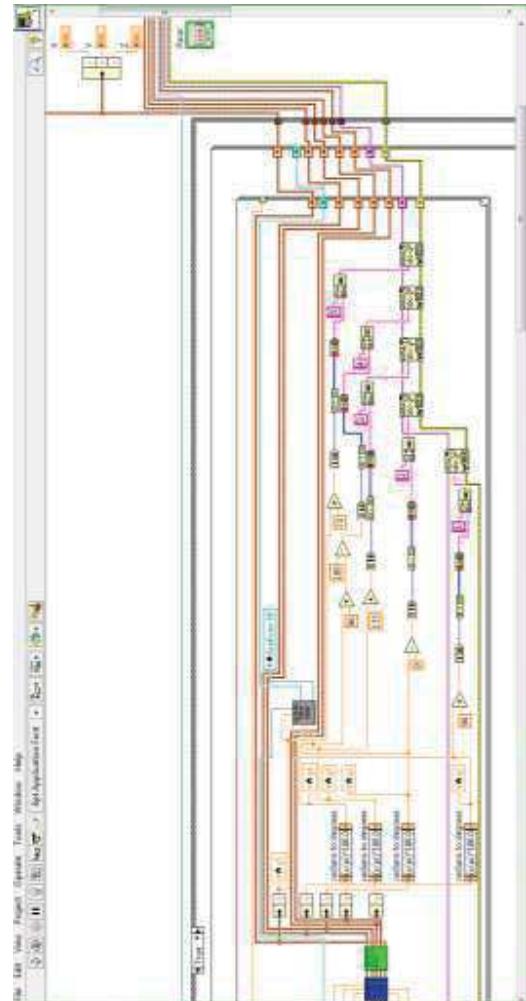


Fig. 9. VI del control

4. Conclusiones

En este artículo se ha descrito el desarrollo de un simulador de un robot antropomorfo de 5 GDL. Las conclusiones se describen a continuación:

-El desarrollo del simulador permitió integrar diversas herramientas computacionales y matemáticas. La metodología de Denavit-Hartenberg y los softwares Solidworks y LabVIEW se integraron para el diseño del simulador, con lo que se demuestra la sinergia como característica de los sistemas mecatrónicos, como es el caso del robot estudiado en este artículo.

-El prototipo del robot fue útil para probar la funcionalidad del simulador. El software LabVIEW es de gran utilidad para el control de los actuadores y para simular los movimientos.

-La metodología de Denavit-Hartenberg permitió construir los modelos cinemáticos del robot, mismos que fueron

fácilmente programados en LabVIEW y con ellos se logró generar los VI para manipular la cinemática directa e inversa.

-La integración Solidworks-LabVIEW permite desarrollar simuladores y animaciones de robots de forma simple y con una gran visualización gráfica tanto del robot como de sus movimientos, pues por un lado se aprovecha el potencial gráfico del Solidwork y por otro la facilidad de programación del LabVIEW.

-Es importante impulsar la formación de grupos de trabajo integrados por profesores y alumnos, pues los proyectos mecatrónicos no pueden realizarse por una sola persona, pues estos requieren de amplios conocimientos y del uso de diversas tecnologías integradas. El diseño, la fabricación y la simulación del robot descrito en este artículo requirió de la participación de alumnos y profesores del ITN.

Referencias

[1] Delfín J. Análisis cinemático de un robot paralelo planar de tres grados de libertad tipo RRR. México. (Tesis de Maestría en Ingeniería Mecánica). UNAM, Facultad de Ingeniería, División de Estudios de Posgrado, Sección Mecánica. 2004.

[2] Arias M., "Robótica," Politécnico Colombiano "Jaime Isaza, Medellín. 2004

[3] Jiménez E. Simulación de un proceso de manufactura con obstáculo en la línea de producción. México. (Tesis de Maestría en Ingeniería Mecánica). UNAM, Facultad de Ingeniería, División de Estudios de Postgrado, Sección Mecánica. 1998.

[4] Jorge A. "Fundamentals of Robotic Mechanical System. Theory, Methods and Algorithms". Springer- Verlag. New York. 1997.

[5] Meléndez J., Jiménez E., Ruiz A., Urbalejo A., Reyes L., Luna N., Vázquez I. Desarrollo de un simulador de un robot cartesiano en dispositivos móviles iOS utilizado Solidworks y Open GL ES 2.0. *13º Congreso Nacional de Mecatrónica*. Octubre 30 a Noviembre 1. Querétaro, Qro. (Por presentarse). 2014.

[6] Jiménez E., Reyes A., Islas M., Ávila F., Ochoa F., Barceló B. A 2-DOF Robot Modelation using Quaternions and LABVIEW Simulations. VI International Conference on Engineering and Computer Education. Pp. 1252-1256. March. Buenos Aires, Argentina. 2009.

[7] Modeling and control of Robot Manipulators: Lorenzo Sciavico, Bruno Siciliano, Ed. Mc. Graw Hill. 1996.

[8] Martínez A., Jáquez O. Sonia A., Rivera M., Sandoval R. Diseño propio y Construcción de un Brazo Robótico de 5. RIEE&C. Revista de ingeniería eléctrica, electrónica y computación, Vol. 4 No. 1, Julio. ISSN 1870 – 9532. 2008

[9] Mellodge P. Brazo Robótico de 5 grados de libertad; Robótica al alcance de todos., www.edudevices.com.ar

Desarrollo de un Simulador de un Robot Cartesiano para dispositivos móviles utilizado Solidworks y Open GL ES 2.0

Meléndez Campos Javier¹, Jiménez López Eusebio², Ruiz Domínguez Alejandra¹, Urbalejo Contreras Arturo³, Reyes Ávila Luis⁴, Luna Acosta Noé³ y Vázquez Cuevas Ignacio⁵

¹ Universidad La Salle Noroeste, jmelendezcampos@hotmail.com, alejandra_ruiz92@hotmail.com
² ULSA Noroeste - CINNTRA de la Universidad Tecnológica del Sur de Sonora-IIIMM, ejimenezl@msn.com
³ CIAAM de la Universidad Tecnológica del Sur de Sonora, urbalejoa@hotmail.com, nluna@uts.edu.mx

⁴ Instituto Mexicano del Transporte-IIIMM, lreyesa@imt.mx
⁵ Universidad Tecnológica de Nogales-ITN, ing.ignacio.javier@gmail.com

Resumen

En este artículo se presenta el desarrollo de un simulador cinemático de un robot cartesiano de dos grados de libertad (2GDL) usando el software Solidworks 2014 y gráficos en OpenGL ES 2.0. El modelo cinemático del robot fue construido haciendo un análisis de cuerpo rígido y un análisis de trayectoria rectilínea conformada por lugares geométricos rectilíneos y perfiles de trayectoria trapezoidales suavizados con funciones polinomiales de grado 5. El simulador fue diseñado para ser visualizado y operado en dispositivos móviles (Tablets) bajo el sistema operativo iOS. El robot fue diseñado y construido virtualmente en Solidworks y posteriormente los archivos fueron transferidos a OpenGL ES 2.0. Los modelos cinemáticos del robot y la trayectoria fueron importados del software de cálculo simbólico MatLAB. El simulador puede ser fácilmente operado desde una Tablet.

Palabras clave: Robot cartesiano, SolidWorks, Open GL, Dispositivos Móviles.

1. Introducción

El diseño de cualquier diseño mecatrónico hoy en día implica necesariamente la incorporación de la simulación computacional en cualquiera de los sistemas que componen el desarrollo de un producto, un proceso o un sistema. En forma general, el término simular es sinónimo de imitar el comportamiento de un sistema con algún propósito específico [1]. Para realizar las simulaciones numéricas y gráficas de robots se han utilizado diferentes programas de uso específico, entre estos destacan los siguientes: a) aquellos que se basan en el diseño mecánico, por ejemplo Autocad®, SolidWorks®, Mechanical Desktop®, etc., y b) aquellos que se basan en el

modelo dinámico, por ejemplo Matlab®-Simulink®, Simmon, etc [2]. El desarrollo de una simulación depende de las necesidades del cliente, de la selección del software y de las plataformas de programación. En este proceso no se aplica una metodología específica, más bien el integrador debe explorar las distintas posibilidades que se tienen en cuanto a tecnologías disponibles y recursos computacionales.

Existen diversos trabajos que reportan desarrollos de simuladores robóticos, por ejemplo en [3] se presenta el desarrollo de un robot hexápodo usando CATIA y un software para el cálculo y el modelado dinámico. En [4] se utiliza el software Solidworks para el análisis y modelado cinemático y cinético de un robot de rescate. En este artículo se presenta el desarrollo de un simulador de un robot cartesiano usando Solidworks 2014, Open GL ES 2.0, MatLAB y bibliotecas de Apple. El simulador se opera desde un dispositivo móvil. Se describen el modelo cinemático y de trayectoria relacionados con el robot y un método de interpolación polinomial.

2. Descripción del problema

Se busca desarrollar un robot cartesiano para que forme parte de un proceso productivo de una celda de manufactura didáctica. Para la concepción y fabricación del robot, se requiere de la formación de un grupo multidisciplinario conformados por alumnos y profesores de la Universidad la Salle Noroeste y la Universidad Tecnológica del Sur de Sonora (UTS) bajo el convenio de la RED ALFA. Los requerimientos técnicos para el desarrollo del robot fueron los siguientes:

- 1) Diseño en CAD de las partes y ensamble del robot.

- 2) Desarrollo de un simulador del robot integrando trayectorias rectilíneas con perfiles de trayectoria trapezoidales.
- 3) Uso de técnicas de manufactura aditiva para la fabricación del robot.
- 4) Debe incluir motores paso a paso.
- 5) El control y manejo del robot debe realizarse por medio de un dispositivo móvil (Tablet o celular).
- 6) El robot formará parte de una celda de manufactura para propósitos didácticos del proyecto PROMEP de la UTS y para hacer pruebas de sincronización en celdas de manufactura de las universidades que integran la red ALFA.

3. Propuesta de solución para el desarrollo del simulador

El desarrollo de cualquier sistema mecatrónico requiere del diseño o la utilización de un simulador computacional. Para el caso del robot solicitado por la empresa el diseño del simulador debe ser de tal forma que se pueda operar desde un dispositivo móvil, puesto que el control del robot físico se hará desde dicho dispositivo. Después de haber discutido algunas ideas se propuso el siguiente desarrollo del simulador:

- 1) Dibujar el robot y sus componentes en un paquete CAD.
- 2) Construir el modelo cinemático del robot y la trayectoria en un paquete de cálculo simbólico.
- 3) Usar métodos gráficos y software para dispositivos móviles y migrar la información del dibujo CAD y los modelos del robot.
- 4) Probar el simulador en el dispositivo móvil elegido.

4. Propuesta del software

Una vez elegido el camino y las ideas principales para el desarrollo del simulador se tomó la decisión de utilizar los siguientes paquetes y plataformas computacionales:

- 1) Para el desarrollo del robot en CAD se usó Solidworks 2014.
- 2) Para modelar y obtener las funciones y/o datos numéricos del robot y de la trayectoria se usó MatLAB.

- 3) Sistema operativo iOS para dispositivos móviles y X-CODE para la programación.
- 4) Lenguaje de programación OBJECTIVE-C.
- 5) Para la simulación gráfica se usó OPEN GL ES 2.0.

La Figura 1 muestra el software utilizado.



Fig. 1. Software utilizado para el desarrollo del robot cartesiano.

5. Desarrollo del simulador

En esta sección se presenta el desarrollo del simulador motivo de estudio en este artículo.

5.1 Modelado cinemático del robot.

El modelado cinemático del robot se dividió en dos partes: 1) Modelo de cuerpo rígido para el robot y 2) Modelo de trayectoria. La Figura 2 muestra la configuración del robot [5].

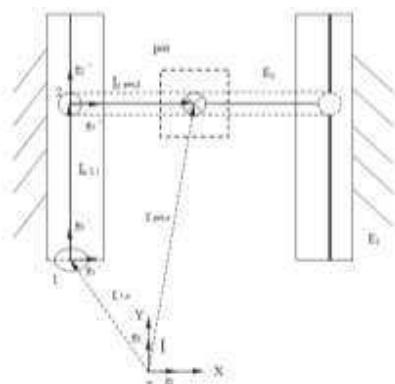


Fig. 2. Configuración del robot.

De acuerdo con la Figura 2, el robot está constituido por dos eslabones (E_1 y E_2). El objetivo del modelado cinemático es construir las

ecuaciones de posición, desplazamiento, velocidad y aceleración relacionadas con el punto del órgano termina (pot) con referencia al origen del sistema cartesiano inercial (X,Y).

El modelo de posición es:

$$\underline{r}_{\text{pot},0} = \underline{r}_{1,0} + l_{2,1} \bullet \underline{e}_2' + l_{\text{pot},2} \bullet \underline{e}_1'' \quad (1)$$

Aquí, $l_{2,1}$ y $l_{\text{pot},2}$ son las longitudes de los eslabones y los sistemas móviles \underline{e}_2' y \underline{e}_1'' se desplazan con los eslabones.

La expresión (1) se puede escribir en términos de coordenadas de la manera siguiente:

$$\underline{r}_{\text{pot},0} = \underline{r}_{1,0} + (y_2 - y_1) \bullet \underline{e}_2 + (x_{\text{pot}} - x_2) \bullet \underline{e}_1 \quad (2)$$

Aquí, \underline{e}_1 y \underline{e}_2 son las bases canónicas localizadas en el origen del sistema de referencia mostrado en la Figura 1. Por otro lado, el modelo de desplazamiento es el siguiente:

$$\dot{\underline{r}}_{\text{pot},0}(t) = \underline{r}_{1,0} + \dot{L}'_{2,1}(t) + \dot{L}'_{\text{pot},2}(t) \quad (3)$$

Los modelos de velocidad y aceleración son:

$$1) \quad \dot{\underline{r}}'_{\text{pot},0}(t) = \dot{L}'_{2,1}(t) + \dot{L}'_{\text{pot},2}(t) \quad (4)$$

$$2) \quad \ddot{\underline{r}}'_{\text{pot},0}(t) = \ddot{L}'_{2,1}(t) + \ddot{L}'_{\text{pot},2}(t)$$

O, en forma equivalente:

(5)

$$1) \quad \dot{\underline{r}}'_{\text{pot},0}(t) = \dot{l}'_{2,1}(t) \bullet \underline{e}_2 + \dot{l}'_{\text{pot},2}(t) \bullet \underline{e}_1$$

$$2) \quad \ddot{\underline{r}}'_{\text{pot},0}(t) = \ddot{l}'_{2,1}(t) \bullet \underline{e}_2 + \ddot{l}'_{\text{pot},2}(t) \bullet \underline{e}_1$$

Por otro lado, el modelo de trayectoria del robot estudiado se compone de lugares geométricos rectilíneos y de perfiles de trayectoria trapezoidales. La Figura 3 muestra en líneas punteadas los perfiles trapezoidales de la trayectoria y en línea continua los perfiles suavizados con interpolación polinomial [6].

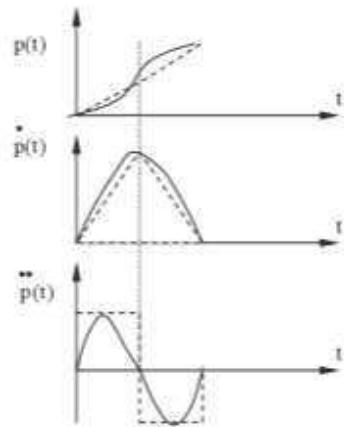


Fig. 3. Perfiles de trayectoria trapezoidales.

Para el caso del simulador motivo de estudio fue usada interpolación polinomial grado 5 para suavizar los perfiles. El modelo de trayectoria final del movimiento del robot del punto **a** al punto **b** en el espacio-tiempo (en este caso en el plano-tiempo) es el siguiente:

$$1) \quad \underline{R}_{b,o}(t) = \underline{R}_{a,o} + p(t) \bullet \frac{1}{d_{b,a}} \begin{bmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \end{bmatrix} \quad (6)$$

$$2) \quad \dot{\underline{R}}_{b,o}(t) = p(t) \bullet \frac{1}{d_{b,a}} \begin{bmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \end{bmatrix}$$

$$3) \quad \ddot{\underline{R}}_{b,o}(t) = p(t) \bullet \frac{1}{d_{b,a}} \begin{bmatrix} x_b - x_a \\ y_b - y_a \\ z_b - z_a \end{bmatrix}$$

Nótese que el vector $\underline{R}_{b,o}(t)$ localiza el punto **b** (punto extremo del lugar geométrico rectilíneo) desde un origen **o** en el espacio tiempo, $\underline{R}_{a,o}$ localiza el punto inicial del lugar geométrico desde el origen **o**, $d_{b,a}$ es la distancia entre dos puntos y $p(t)$ es una función polinómica de grado 5, esto es:

$$p(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \alpha_3 t^3 + \alpha_4 t^4 + \alpha_5 t^5 \quad (7)$$

El paso final para completar el modelado cinemático del robot es ensamblar las ecuaciones (3) y (5) con las ecuaciones (6).

5.2 Programación en MatLAB.

Con la finalidad de determinar la posición, velocidad y aceleración del efecto final (pot) del robot motivo de estudio, se desarrolló un algoritmo el cual tomaba como argumentos: la posición inicial y final del robot, así como la velocidad a la cual el usuario quiere que se mueva el dispositivo. El polinomio de grado cinco consiste en la multiplicación de matrices en las cuales intervienen las ecuaciones de posición, velocidad y aceleración del objeto. Para facilitar la tarea al algoritmo desarrollado en Objective-C, se realizaron las multiplicaciones de matrices en el software MATLAB, el cual dio como resultado una ecuación general del movimiento la cual se transfirió a código C para integrarse con el algoritmo.

De esta manera mediante controles en la interfaz, el usuario puede indicar la posición en el espacio a la que el robot se situará, así como la velocidad en la que se desplazará de un punto a otro. La Figura 4 muestra los perfiles de trayectoria suavizados en MatLAB.

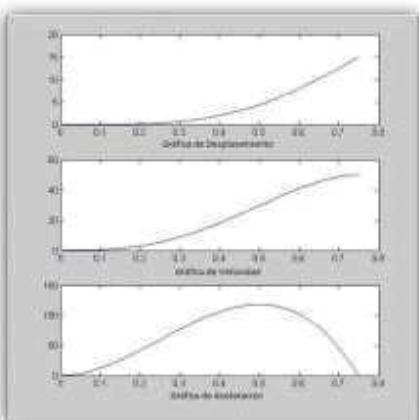


Fig. 4. Perfiles de trayectoria suavizados en MatLAB.

A) Dibujo en sólidos del robot.

El prototipo del robot fue dibujado en la plataforma Solidworks 2014. La figura siguiente muestra la configuración del robot:

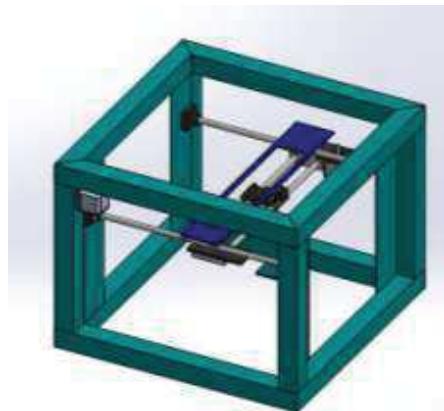


Fig. 5. Modelado en sólidos del robot cartesiano.

El modelado del robot en Solidworks es clave para el desarrollo y la animación del simulador, ya que la información geométrica de los componentes del robot serán migrados a OpenGL ES 2.0.

5.3 Desarrollo gráfico.

Para mejorar la interacción con el usuario se realizó una simulación gráfica en OpenGL la cual tendría que ser lo más representativamente posible al diseño real del robot. Apple, mediante el paquete de XCode, provee una serie de bibliotecas y ejemplos que encapsulan y facilitan el uso de muchas cosas de interés para los desarrolladores, de manera que estos puedan adaptarlos a su plataforma de manera más fácil. De esta forma, se pueden desarrollar aplicaciones más complejas reduciendo de manera significativa el tiempo de programación. Dentro de estas bibliotecas se encuentra una llamada GLKit, la cual encapsula, entre otras cosas, todo lo correspondiente a OpenGL ES 2.0. En dicha biblioteca se encuentran **Objetos** los cuales facilitan la generación y configuración de ventanas (en este entorno llamadas **Vistas**) para renderizado en 2D y 3D. Como se mencionó anteriormente, OpenGL ES 2.0 grafica figuras a partir de primitivas básicas como lo son el triángulo y la línea. Es por eso que el robot motivo de estudio se dibujó usando estas primitivas de manera recurrente, siendo el dibujado vértice por vértice un poco tedioso. Se buscó implementar una forma de integrar los gráficos desde otra plataforma.

Como recurso inicial se contaba con un diseño final del robot construido en Solidworks 2014, (ver Figura 5). Este software es conocido por ser altamente utilizado en la industria, siendo

uno de los líderes a nivel mundial en lo que se refiere a software de diseño asistido por computadora. Sin embargo, Solidworks y los archivos generados por este, no permiten la transferencia directa de gráficos a OpenGL ES 2.0. Se tuvo que diseñar una metodología de migración de gráficos de las piezas y ensambles de Solidworks a información ordenada de vértices, la cual puede ser adoptada e interpretada por OpenGL ES 2.0, para posteriormente ser graficada triángulo por triángulo en pantalla.

Para realizar la migración se utilizó como intermediario el formato OBJ usado en algunos programas de Diseño 3D que a su vez son implementados con regularidad en la industria de los videojuegos. Este formato guarda de manera ordenada y de manera entendible información básica de las figuras, como los vértices, caras, colores y materiales. Solidworks tampoco es capaz de exportar sus figuras y ensambles en este formato por sí solo, sin embargo en internet se pueden encontrar múltiples programas llamados “Solidworks Macros” que al sumarse a Solidworks, entre otras tareas son capaces de realizar exportaciones en este formato.

Como ya se mencionó con anterioridad, el formato OBJ sólo se utilizó como intermediario entre Solidworks y OpenGL ES 2.0, es por eso que se tuvo que buscar y modificar un programa que tomara la información del archivo OBJ y lo estructurara en archivos de formato C (.h y .c) de tal manera que fuera posible acceder a esta información desde OBJECTIVE-C. Una vez generados estos archivos, son importados a XCode e instanciados desde el Objeto de la clase correspondiente a GLKit. Si se abren los archivos generados por el programa se encontrarán con una serie de arreglos simples de C en el que se enumeran de manera ordenada todos los vértices, incluso se le da un formato para que sea más fácil visualizar los grupos de tres vértices correspondientes a cada uno de los triángulos. De esta manera la figura del robot será mostrada en pantalla justo como se mostraba en Solidworks.

Cabe mencionar que está en manos del usuario aplicar técnicas de programación de OpenGL para realizar las transformaciones que él desee a la figura, es decir, escalar, desplazar o en su defecto rotar la figura.

Para el caso del robot motivo de estudio, se usó una rotación general para que el usuario fuera capaz de manipular la perspectiva en la que miraba la figura del robot. Para esto se usó un

método que se volvió muy popular con el aumento de las pantallas táctiles en la vida cotidiana de las personas. Consiste en encerrar la figura en una esfera imaginaria, el usuario se encarga de tocar la pantalla y arrastrar la esfera, de manera que la figura dentro de esta también se rota. Se puede decir que es muy similar a la manera en que se interactúa con un globo terráqueo, mientras este gira. Tal vez sea muy común para todos realizar este tipo de gestos en pantalla ya que se utilizan en la mayoría de las aplicaciones para teléfonos inteligentes y tablets que se encuentran en el mercado en las cuales se utilizan gráficos.

Además de las rotaciones se utilizaron desplazamientos, regularmente llamados traslaciones. Es de vital importancia realizar un seccionado de las partes móviles del ensamblaje de Solidworks, así como realizar la exportación separada de cada una de ellas, de tal manera que las partes del robot puedan ser manipuladas de manera independiente al momento de las transformaciones de OpenGL. Es en esta parte donde se determinó una posición (0,0) para el efecto final del robot, y desde ahí se realizan los trasladados correspondientes a cada eje de la máquina. La distancia de traslado es indicada por el algoritmo en el cual se realizó el polinomio de grado 5. En resumen, se pretende que se enlacen las variables de forma que cada vez que haya un cambio y se decida mover la posición del robot, se actualiza la variable de traslado gráfico para que el usuario se percate del comportamiento del robot de manera inmediata. La Figura 6 muestra el simulador final del robot cartesiano.

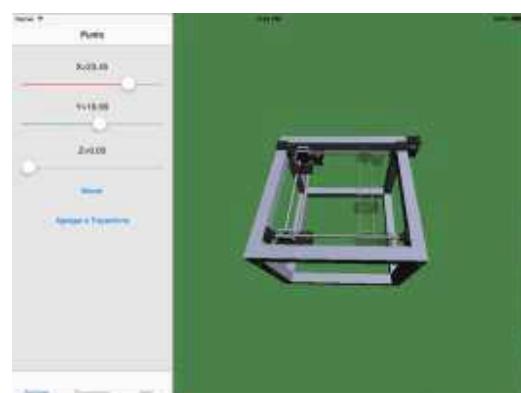


Fig. 6. Simulador final del robot cartesiano.

6. Conclusiones

En este artículo se ha descrito el desarrollo de un simulador cinemático de un robot cartesiano, el cual es operado desde un dispositivo móvil. Las principales conclusiones se resumen en los puntos siguientes:

-Los diseños mecatrónicos requieren necesariamente de la simulación computacional. Para el caso de robot motivo de estudio, la simulación cinemática fue hecha integrando diferentes softwares y plataformas de programación lo que resultó un sistema fácil de operar en un dispositivo móvil.

-La simulación cinemática requiere del desarrollo de modelos matemáticos operacionales y de paquetes de cálculo formal. Para el caso del robot estudiado en este artículo, el modelado se realizó en forma separada (robot y trayectoria) y, posteriormente, se integraron los modelos. Este proceso facilitó la programación en MatLAB.

-El entorno de desarrollo XCode y las bibliotecas que incluye, hicieron más fácil el diseño e implementación del simulador para el sistema operativo iOS, incluyendo cálculos avanzados y gráficos de alta calidad, teniendo como producto final una aplicación amigable y fácil de usar por el usuario.

-La metodología usada para la migración de gráficos redujo el tiempo de programación y dio como resultado un modelo detallado del diseño original hecho en Solidworks.

Agradecimientos

Los autores de este trabajo agradecen a las universidades y empresas que integran la RED Interinstitucional ALFA, a la RED de Manufactura y Mantenimiento, a la empresa Spin – OFF Innovación en Ingeniería de Manufactura y Mantenimiento S. de R.L. MI, al Cuerpo Académico CIAAM de la UTS, al Programa del Mejoramiento del Profesorado (PROMEP) bajo el proyecto Desarrollo de un Prototipo de Robot Industrial para Aplicaciones en Celdas de Producción Automatizadas Clave UTSSON-CA-4, por el apoyo brindado para la realización de este trabajo.

[1] Jiménez E., Navarro J., Reyes L., Luna N., Urbalejo A., Islas M., Castro J. Simulación de una celda de manufactura usando Quaterniones y Mathematica. *11º Congreso Nacional de Mecatrónica*. Villahermosa, Tabasco. 2012.

[2] Maldonado H., Silva R., Ramos E., Hernández V., Rivera J. Modelado y simulación de un robot rígido de dos grados de libertad. *Lat. Am. J. Phys. Educ.* Vol.5, No. 1. 2011.

[3] Mahapatra A., Shekhar S., Kumar D. Modeling and Simulation of Wave Gait of a Hexapod Walking Robot: A CAD/CAE Approach. *International Journal of Robotics and Automation (IJRA)*. Vol. 2, No. 3, pp. 104~111. 2013.

[4] Hassanzadeh E., Shahmohammadi M., Khamseh N. Kinematic and Kinetic Study of Rescue Robot by SolidWorks Software . *Research Journal of Applied Sciences, Engineering and Technology* 5(21): 5070-5076. 2013.

[5] Sobarso J. Modelo y simulación de las trayectorias de un robot cartesiano usando la integración LabVIEW-SolidWorks para aplicaciones industriales. Tesis de Maestría en Ciencias de la Ingeniería Mecatrónica. Instituto Tecnológico Superior de Cajeme. 2012.

[6] Jorge A. “Fundamentals of Robotic Mechanical System. Theory, Methods and Algorithms”. Springer- Verlag. New York. 1997.

Referencias

Integración Mecatrónica en el Desarrollo de un Robot Cartesiano Didáctico

Jiménez López Eusebio¹, Ruiz Domínguez Alejandra², Meléndez Campos Javier², Pereo Arvayo Gerardo², Carvajal Lavandazo Ivanna², Martínez Molina Víctor Manuel³, López Figueroa Francisco⁴ y Luna Sandoval Gabriel⁵.

¹Universidad La Salle Noroeste – CINNTRA-UTS-IIMM, ejimenezl@msn.com

²Universidad La Salle Noroeste, alejandra_ruiz92@hotmail.com, jmelendezcampos@hotmail.com,
ivanna_carvajal@hotmail.com, gerardopereo@gmail.com

³CINNTRA de la Universidad Tecnológica del Sur de Sonora-ULSA Noroeste-CECATI 94,
ymartinez@uts.edu.mx

⁴CIAAM de la Universidad Tecnológica del Sur de Sonora, flopez@uts.edu.mx

⁵Universidad Estatal de Sonora, gabriel.luna@ues.mx

Resumen

En este artículo se presenta el desarrollo de un robot cartesiano didáctico utilizando integración mecatrónica. El robot fue dibujado en Solidworks, tanto en modelado de sólidos, como en planos de fabricación. Para la fabricación de algunos de los componentes se usó una impresora 3D de extrusión de plástico, la cual interpretó los archivos en formato (.STL) importados desde SolidWorks 2014 y, posteriormente, se procedió a la fabricación de algunas piezas de geometrías tanto simples como complejas. El sistema de actuación y transmisión del movimiento se realizó por medio de motores a pasos controlados con tarjetas electrónicas (Arduino UNO), sistema de bandas síncronas y poleas dentadas. Fue desarrollado un simulador cinemático integrado por una trayectoria rectilínea y el modelo del robot, el cual es operado desde un iPAD. Para el caso del simulador se usaron archivos de MatLAB, SolidWorks, Xcode, Migrador SolidWorks-OBJ-VertexData y Open GL ES 2.0. La manipulación del robot se realiza desde un dispositivo móvil (Tablet) mediante el sistema Bluetooth 4.0.

Palabras clave: Robot cartesiano, SolidWorks, Impresión 3D, Dispositivos Móviles, Bluetooth 4.0, iOS, Motores a pasos.

1. Introducción

La Mecatrónica es una disciplina de carácter científico y tecnológico, e integra áreas del conocimiento asociadas con la Ingeniería Mecánica, Ingeniería Electrónica y Sistemas Computacionales. Un sistema mecatrónico se

refiere a la integración de componentes mecánicos, electrónicos y de programación que incluye principalmente aspectos de control automático, y administración de proyectos para realizar la automatización de productos, procesos o servicios con bajos costos, flexibilidad y características de competitividad [1].

Los robots son sistemas mecatrónicos [2], pues se conforman de sistemas mecánicos, electrónicos y computacionales. Por ello, el diseño y la fabricación de estos productos, requiere de la utilización de una alta integración de tecnologías, conocimientos y metodologías disponibles, particularmente si se tratan de prototipos generados en proyectos de docencia e investigación, ya que los alumnos deben ser capaces de valorar y utilizar en la medida de lo posible la alta integración mecatrónica.

En la actualidad dos tecnologías se han incorporado al desarrollo de sistemas mecatrónicos, una es la manufactura aditiva [3] y otra es el control a distancia a través de dispositivos móviles como celulares o Tablets [4]. En este artículo se presenta el desarrollo de un prototipo de robot cartesiano usando integración mecatrónica. El software Solidworks 2014 fue utilizado para el diseño y la transferencia de archivos hacia otros sistemas para el desarrollo de un simulador y la fabricación de partes en impresión 3D. El prototipo es controlado por una tarjeta Arduino y gobernado desde una Tablet (iPad).

Algunos trabajos en donde se usa alta integración mecatrónica se describen a continuación. En [5] se desarrolla un control de un

robot usando tecnología Bluetooth y un microcontrolador 8051. El control es operado desde un teléfono móvil con sistema operativo Android. En [6] se presenta el diseño y la implementación de un robot usado para asistir a personas. Su aplicación gira en torno a un tablero chipKIT Arduino que interconecta un controlador robótico con un Tablet PC y varios sensores. El Tablet PC es Webcam, Wi-Fi y Bluetooth activado, ofreciendo una plataforma versátil que es capaz de procesar los datos y, al mismo tiempo, proporciona al usuario una interfaz amigable.

2. Descripción del problema

Se busca desarrollar un robot cartesiano para que forme parte de un proceso productivo de una celda de manufactura didáctica. Para la concepción y fabricación del robot, se requiere de la formación de un grupo multidisciplinario conformados por alumnos y profesores de la Universidad la Salle Noroeste y la Universidad Tecnológica del Sur de Sonora (UTS) bajo el convenio de la RED ALFA. Los requerimientos técnicos para el desarrollo del robot fueron los siguientes:

- 1) Diseño en CAD de las partes y conjunto de robot.
- 2) Desarrollo de un simulador del robot integrando trayectorias rectilíneas con perfiles de trayectoria trapezoidales.
- 3) Uso de técnicas de manufactura aditiva para la fabricación del robot.
- 4) Debe incluir motores a pasos.
- 5) El control y manejo del robot debe realizarse por medio de un dispositivo móvil (Tablet o celular).
- 6) El robot formará parte de una celda de manufactura para propósitos didácticos del proyecto PROMEP de la UTS y para hacer pruebas de sincronización en celdas de manufactura de las universidades que integran la red ALFA.

3. Propuesta de solución para el desarrollo del prototipo del robot.

Debido a la naturaleza didáctica del prototipo de robot cartesiano, se propone que la concepción y desarrollo del dispositivo integre tecnologías de la manufactura aditiva, control a distancia y por dispositivos móviles, uso de software CAD, paquetes de cálculo simbólico, control con tarjetas comerciales, sistema de actuación por actuadores convencionales y kits de bandas síncronas.

Después de haber discutido algunas ideas, se propuso el siguiente desarrollo del prototipo:

- 1) Dibujar el robot y sus componentes en un paquete CAD.
- 2) Construir el modelo cinemático del robot y la trayectoria en un paquete de cálculo simbólico.
- 3) Usar métodos gráficos y software para dispositivos móviles, migrar la información del dibujo CAD y los modelos del robot a otro software de más poder gráfico.
- 4) Probar el simulador en el dispositivo móvil elegido.
- 5) Utilizar un sistema de manufactura aditiva (Impresión 3D) para la fabricación de algunos componentes.
- 6) Usar sistemas tradicionales de manufactura para la construcción del robot.
- 7) Utilizar motores a pasos y control por tarjetas comerciales.
- 8) Usar dispositivos para la comunicación inalámbrica.

4. Propuesta de tecnologías

Una vez elegido el camino y las ideas principales para el desarrollo del prototipo se tomó la decisión de utilizar los siguientes paquetes y plataformas computacionales, tecnologías de la fabricación y control:

Para el simulador:

- 1) Solidworks 2014.
- 2) MatLAB.
- 3) Sistema operativo iOS para dispositivos móviles y X-CODE.
- 4) OBJETIVE-C.
- 5) OPEN GL ES 2.0.

Para el diseño y la fabricación:

- 1) Solidworks 2014 para el diseño y el desarrollo de archivos en extensión .STL.
- 2) Uso del sistema de impresión 3D Modelo: AW3D V5, para la fabricación de algunas partes del prototipo.
- 3) Maquinaria convencional de fabricación de arranque por viruta y soldadura industrial.

Para el sistema de actuación y control:

- 1) Kit de Bandas y Poleas MakeBlock – Dorado.
- 2) Motores a pasos 42x34 12v
- 3) Tarjeta Arduino UNO.
- 4) Ipad
- 5) Bluetooth 4.0.

La Figura 1 muestra las tecnologías usadas para el desarrollo del prototipo del robot cartesiano.



Fig. 1. Software utilizado para el desarrollo del robot.

5. Desarrollo del prototipo

El desarrollo del prototipo se sistematizó en tres partes: 1) Desarrollo del simulador, 2) Diseño y fabricación del dispositivo y 3) Control del robot.

5.1 Desarrollo del simulador

En un trabajo reciente se ha explicado el desarrollo del simulador del robot motivo de estudio [7]. Las tecnologías usadas para el diseño del simulador fueron: archivos de MatLAB

(funciones y cálculo de la cinemática del robot), SolidWorks (migración de partes gráficas del robot), Xcode, Migrador SolidWorks-OBJ-VertexData y Open GL ES 2.0. La Figura 2 muestra la pantalla final del simulador.

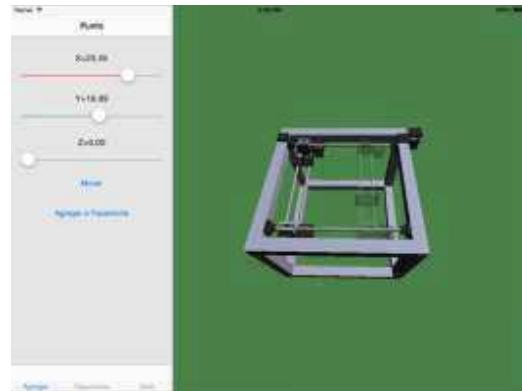


Fig. 2. Simulador del robot cartesiano en iPad

5.2 Diseño y fabricación del dispositivo.

Para el diseño del prototipo se utilizó Solidworks 2014. Este software tiene múltiples aplicaciones, entre ellas el modelado de sólidos y el desarrollo de planos de fabricación para la manufactura de partes y componentes. La Figura 3 muestra el prototipo del robot en modelo de sólidos.



Fig. 3. Robot en Solidworks 2014

La Figura 4 muestra un plano de fabricación del prototipo. Dichos planos se usan para la fabricación tradicional de componentes.

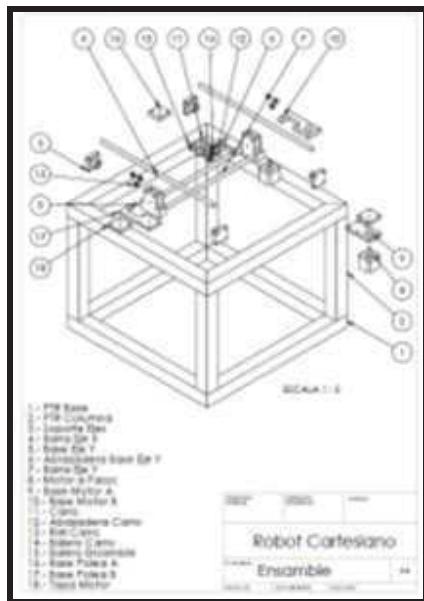


Fig. 4. Plano de fabricación.

Algunas piezas del prototipo se fabricaron con máquinas herramientas en el taller de fabricación del CECATI 94 y otras piezas en una máquina de impresión 3D Modelo: AW3D V5 como la mostrada en la Figura 5. Para la impresión de las piezas fue necesario exportar del Solidworks 2014 las piezas en extensiones de archivos .STL.

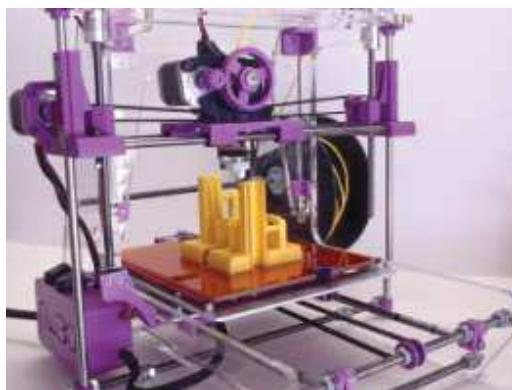


Fig. 5. Máquina para impresión 3D.

La Figura 6 muestra el prototipo fabricado, el cual consta de una estructura tubular rígida soldada, los sistemas de guías, acoplamientos de actuadores y bandas dentadas.



Fig. 6. Prototipo ensamblado.

5.3 Control del robot.

Se determinó que el robot se moviera mediante dos pequeños motores a pasos bipolares que trabajan con un voltaje de 12 V. A su vez, se requiere que el sistema electrónico sea capaz de aceptar órdenes desde una aplicación de iPad, desde donde se controlarán los desplazamientos en los diferentes ejes según lo requiera el usuario. Se consideró que el sistema requeriría de un microcontrolador, de tal manera que este sirviera como intermediario para controlar los motores a pasos recibiendo órdenes desde el dispositivo móvil.

Por otro lado, debido a la facilidad que los desarrolladores encuentran al programar en un entorno de programación como C, y al ahorro de tiempo que esto traería debido a la disponibilidad de recursos, bibliotecas, foros y soporte que esto tiene, se escogió una tarjeta Arduino como el microcontrolador que estaría en el centro del sistema. Sin embargo, la tarjeta Arduino por sí sola no puede recibir datos de una plataforma diferente como el sistema operativo iOS que rige a las aplicaciones de iPad, iPhone o iPod. Y debido a esto se escogió el Bluetooth 4.0 como la manera más óptima de comunicar iPad con Arduino. Una compañía estadounidense llamada RedBearLab ofrece un aditamento que incorpora un sistema Bluetooth LE, acompañado por un sistema de desarrollo basado en objetos que puede ser rápidamente adaptado por los desarrolladores en su aplicación para enviar datos como si se tratara de un puerto serial simple. Esta biblioteca cuenta con funciones básicas como Leer y Escribir sobre el puerto serial, de tal manera que las salidas de esta tarjeta son dos simples pines RX/TX los cuales se conectan al microcontrolador en el cual se leerán o en su defecto desde el cual escribirán los

datos de comunicación Serial. Arduino cuenta con bibliotecas y funciones desarrolladas para usar las capacidades de la UART del microcontrolador que contiene. De tal manera que la tarjeta Bluetooth se conecta en los pines RX/TX de Arduino, haciendo así posible la comunicación entre iPad y Arduino vía Bluetooth LE. De esta manera, la aplicación de iPad, rica en gráficos y controles amigables para el usuario, sería capaz de enviar la información de los pasos que requería moverse el Robot para ir de un punto a otro. En la Figura 7 se describe de forma muy general el diseño final del sistema de control.

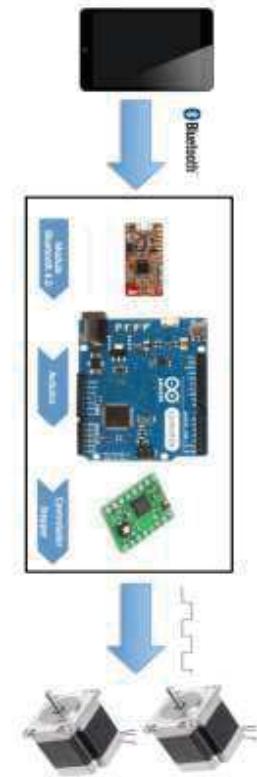


Fig.7. Diagrama general de etapa de control

Una vez realizada la comunicación entre la aplicación y la placa Arduino, se buscó una forma sencilla de poder controlar los motores a pasos según las órdenes recibidas por el puerto Serial. Así como Arduino cuenta con una biblioteca para facilitar la comunicación serial, también cuenta con una biblioteca que soporta el control de motores a pasos con su respectiva etapa de potencia.

El diseño de esta etapa de potencia es sugerida mediante puentes H, pero existen

múltiples tarjetas en el mercado las cuales integran toda esa circuitería en menor espacio y de manera más estética. Es por eso que se buscó una tarjeta capaz de controlar motores a pasos bipolares que pudiera soportar la corriente que los motores demandan. En la Figura 8 se observa la integración final del sistema de control electrónico el cual contiene las tarjetas Arduino con los controladores de motores a pasos, así como con la tarjeta Bluetooth 4.0.



Fig. 8. Integración de sistema de control

Finalmente, se seleccionó un A4988, el cual además de contar con la etapa de potencia necesaria, permite escoger entre 5 tipos de pasos diferentes mediante niveles lógicos, logrando así mayor exactitud. De esta manera el Arduino controlará dos tarjetas de este tipo usando 4 salidas digitales por cada una: tres para seleccionar el tipo de paso y una como señal de reloj. El resultado final se observa en la Figura 9 donde el usuario de manera inalámbrica controla los movimientos del robot.



Fig. 9. Control desde iPad

Conclusiones

En este artículo se ha descrito el desarrollo de un robot cartesiano didáctico usando integración mecatrónica. Las principales conclusiones se resumen en los puntos siguientes:

-Para el desarrollo de un sistema mecatrónico, como es el caso del robot cartesiano motivo de estudio, se requiere utilizar todos los recursos tecnológicos disponibles y las metodologías con el propósito de aumentar el grado de integración en el diseño del sistema. Para el desarrollo del robot se usaron tecnologías de software y hardware, de manufactura aditiva y convencional, tarjetas electrónicas, motores a pasos y dispositivos móviles.

-El Solidworks 2014 fue una herramienta computacional útil y de gran funcionalidad ya que permitió una alta integración mecatrónica, pues se desarrollaron dibujos planos para la fabricación de componentes, se generaron archivos con extensión .STL para su transferencia hacia dispositivos de manufactura aditiva (Impresión 3D) y archivos para el desarrollo del simulador en OPEN GL ES 2.0.

-Durante el desarrollo del simulador se observó otro proceso de alta integración mecatrónica al requerirse archivos de MatLAB, SolidWorks, Xcode, Migrador SolidWorks-OBJ-VertexData y Open GL ES 2.0.

-Se integraron tecnologías referentes a microcontroladores mediante el uso de la plataforma Arduino, lo cual permitió la comunicación serial con dispositivos externos al robot, así como la conexión con tarjetas controladoras de motores a pasos. Este procedimiento de integración redujo de manera significativa el tiempo de desarrollo e implementación de un sistema electrónico para el control de este tipo de motores

-Se utilizó lo último en comunicación Bluetooth para realizar la comunicación Serial usando Bluetooth 4.0, permitiendo así el control del dispositivo desde un mando a distancia.

Agradecimientos

Los autores de este trabajo agradecen a las universidades y empresas que integran la RED Interinstitucional ALFA, al CECATI 94, a la empresa Spin – OFF Innovación en Ingeniería de

Manufactura y Mantenimiento S. de R.L. MI, al Cuerpo Académico CIAAM de la UTS, al Programa del Mejoramiento del Profesorado (PROMEP) bajo el proyecto Desarrollo de un Prototipo de Robot Industrial para Aplicaciones en Celdas de Producción Automatizadas Clave UTSSON-CA-4, por el apoyo brindado para la realización de este trabajo.

Referencias

- [1] Vargas E., Aceves M., Sotomayor A., Tovar S., Rodríguez J. Perspectivas de la Ingeniería Mecatrónica. *VIII Congreso Internacional de Ingeniería Mecánica y Mecatrónica*. Abril 10-12 y Mayo 29-31, Puerto Vallarta, Jalisco. (2014).
- [2] Jorge A. "Fundamentals of Robotic Mechanical System. Theory, Methods and Algorithms". Springer- Verlag. New York. 1997.
- [3] Ramírez P., López J. (2011). Tecnologías aditivas, un concepto más amplio que el de prototipado rápido. *XV Congreso Internacional de Ingeniería de Proyectos*. Huesca, 6-8 de julio. 2011.
- [4] Selvam M. Smart Phone Based Robotic Control for Surveillance Applications. *International Journal of Research in Engineering and Technology*. Volume: 03 Issue: 03. (2014).
- [5] Pahuja R., Kumar N. Android Mobile Phone Controlled Bluetooth Robot Using 8051 Microcontroller. *International Journal of Scientific Engineering and Research*. Volume 2 Issue 7. 2004.
- [6] Alexan A., Osan A., Oniga S. AssistMe robot, an assistance robotic platform. *Journal of Electronic and Computer Engineering*. Vol 5 . pp 1-4. 2012.
- [7] Meléndez J., Jiménez E., Ruiz A., Urbalejo A., Reyes L., Luna N., Vázquez I. Desarrollo de un simulador de un robot cartesiano en dispositivos móviles iOS utilizado Solidworks y Open GL ES 2.0. *13º Congreso Nacional de Mecatrónica*. Octubre 30 a Noviembre 1. Querétaro, Qro. (Por presentarse). 2014.

Libro digital en CD

Ingeniería Mecatrónica. Proyectos y Aplicaciones

se terminó de imprimir el día 23 de Octubre del 2014
en los talleres de:

Asociación Mexicana de Mecatrónica A.C.

Prolongación Corregidora Norte No. 168, interior E
Col. El Cerrito, C.P. 76160, Querétaro, Qro.

Se tiraron 1000 ejemplares, en formato digital html, tipo CD,
Tamaño: 200 Mb.

ISBN: 978-607-9394-00-4