

ISSN 1390-6712

MASKAY



ELECTRÓNICA

Vol. 3, No. 1

Revista MASKAY Electrónica

Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas – ESPE

DIRECTOR DEL DEEE

Ing. Evelio Granizo

EDITOR GENERAL

Dr. Enrique V. Carrera

INFORMACIÓN DE CONTACTO

Revista MASKAY-Electrónica
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas – ESPE

Av. Gral. Rumiñahui (S/N)
P. O. Box 17-15-243B
Sangolquí, Pichincha
Ecuador

El contenido de los artículos aquí publicados es responsabilidad exclusiva de sus correspondientes autores. Mayor información en <http://maskay.espe.edu.ec/>.

Presentación

El Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas – ESPE, consciente de su gran responsabilidad con la sociedad, busca proponer e impulsar iniciativas tendientes a fomentar el desarrollo integral de la electrónica en el Ecuador. Como resultado de este compromiso, nace hace tres años la revista Maskay Electrónica. Esta revista sintetiza de una forma documentada los esfuerzos en investigación y desarrollo que viene realizando el Departamento de Eléctrica y Electrónica.

En esta edición, la revista Maskay Electrónica pone a consideración de la comunidad los trabajos realizados por los investigadores de la Universidad de las Fuerzas Armadas – ESPE y otros centros de educación superior en la región. Muchos de los trabajos presentados son el resultado de Proyectos de Fin de Carrera llevados a cabo por estudiantes de las Carreras de Ingeniería en Electrónica y Telecomunicaciones, Ingeniería en Electrónica, Automatización y Control, e Ingeniería en Electrónica, Redes y Comunicación de Datos.

La consecución de esta importante publicación se debe en gran parte al apoyo desinteresado de un sinnúmero de revisores altamente calificados que han desempeñado la difícil tarea de evaluar con objetividad cada uno de los artículos presentados a la revista. Finalmente, como Editor General de la revista deseo dejar constancia de mi sincero agradecimiento a todos los autores que gracias a su empeño y perseverancia han logrado la publicación de sus correspondientes artículos.

Enrique V. Carrera
Editor

Contenido

Mapas de entornos mediante navegación difusa y sistema de teleoperación de una plataforma Pioneer P3-DX <i>Daniel Granda, Danny Vásconez</i>	1
Desempeño de algoritmos gusano en la navegación de una plataforma robótica móvil <i>Luis Álvarez, Juan Figueroa, Rodolfo Gordillo, Luis Chacón</i>	12
Teleoperación plataforma móvil Irobot Create <i>William Quishpe, Andrés Salas, Rodolfo Gordillo, Carlos Altamirano</i>	20
Diseño y análisis de desempeño de un inversor de voltaje utilizando controladores inteligentes <i>Juan José Salazar</i>	24
Diseño e implementación del sistema de monitoreo, supervisión y control automático del proceso de destilación de agua en el área de inyectables de Life C.A. <i>Alan Cuenca, Milton León</i>	33
Diseño e implementación de un Multiprocessor Systems-on-Chip (MPSoC) interconectado por una Networks-on-Chip (NoC) <i>Wilson Chicaiza, Daniel Verdezoto</i>	40
Dispositivo para medición de la presión arterial sobre la plataforma Android para dispositivos móviles <i>Mauricio Vinuesa, Darwin Ahulema, Flavio Pineda</i>	49
Monitorización de signos vitales mediante una red de dispositivos móviles <i>Daniel Cilio, Katherine Herrera, Enrique V. Carrera</i>	58

Comité Técnico

Daniel Altamirano
Diego Arcos
Diego Benítez
Enrique V. Carrera
Danilo Corral
Rodolfo Gordillo
Hugo Ortiz
Carlos Romero
Patricio Vizcaíno

Darwin Alulema
Paúl Ayala
Paúl Bernal
Alejandro Chacón
Marco Flores
Luis Orozco
Víctor Proaño
Danny Sotomayor

Mapas de Entornos Mediante Navegación Difusa y Sistema de Teleoperación de una Plataforma Pioneer P3-DX

Daniel Granda Gutiérrez, Danny Vásconez Chimbo

Resumen—El presente proyecto describe el diseño e implementación de aplicaciones de Teleoperación, Adquisición de Datos, Control Difuso de Velocidad y Mapeo de Entornos en 2D, para la plataforma móvil Pioneer P3-DX mediante el uso de sonares, odometría y software libre GNU/Linux.

El proyecto brinda una guía para utilizar los conceptos de programación en Python, que permite crear aplicaciones de manera versátil mediante el uso de librerías como: GTK para el desarrollo del entorno gráfico, PYFUZZY para el desarrollo del controlador difuso de velocidad y OPENCV para mostrar los mapas del entorno

Index Terms—Robótica, ARIA, OpenCV, Python, GNU/Linux, Controlador Difuso

I. INTRODUCCIÓN

En la actualidad uno de los mayores problemas en las plataformas móviles es la planificación de trayectorias y la navegación.

El mapeo de entornos se puede resolver de distintas maneras, generando mapas de entorno que pueden ser realizados por descripción geométrica, rejilla de ocupación, hitos (landmarks) y otras.

La navegación con visión artificial a veces no es posible ya sea por costos o por características del entorno, una alternativa es la utilización de sonares que son una de las opciones más económicas.

En el proyecto se usó técnicas de lógica difusa, para controlar la velocidad de la plataforma y programación para realizar navegación y estimación del mapa del entorno.

A pesar de que la plataforma Pioneer 3-DX dispone del software MOBILE-EYES que permite realizar teleoperación, navegación y generación de un modelo del entorno, es una solución rígida a la hora de utilizarlo por ser un sistema propietario. Por esta razón se realiza aplicaciones con interfaz gráfica para la visualización y operación de la plataforma Pioneer 3-DX con software abierto.

Se utilizó la plataforma Pioneer 3-DX y su simulador, utilizando los sonares y un programa para generar un modelo de entorno realizando localización y mapeo simultáneo o SLAM por sus siglas en inglés, sin tener que recurrir a MOBILE-EYES.

Daniel Granda Gutiérrez, Ingeniero electrónico en automatización y control graduado en la Escuela Politécnica del Ejercito, (email: degranda87@gmail.com).

Danny Vásconez Chimbo, Ingeniero electrónico en automatización y control graduado en la Escuela Politécnica del Ejercito, (email: dannyvasconez@gmail.com).

La trayectoria se genera mediante la teleoperación de la plataforma móvil al tiempo que se realiza el mapeo del entorno. La plataforma móvil utiliza un control de velocidad difuso para evitar choques con posibles obstáculos que se encuentren en la trayectoria.

Se realizó una interfaz gráfica para realizar la teleoperación, adquisición de datos de los sonares y para mostrar el modelo de entorno. El sistema operativo para realizar el proyecto es una distribución de GNU/Linux, perteneciente a la familia Debian, por lo tanto las librerías ARIA son utilizadas para ese sistema operativo. El lenguaje de programación usado para la creación de las aplicaciones es Python.

II. MARCO TEÓRICO

II-A. Plataforma Pioneer 3-DX

Las plataformas Pioneer son una familia de robots inteligentes móviles, utilizados para educación e investigación.



Figura 1. Plataforma Robótica Pioneer 3-DX

Se define a una plataforma móvil como un dispositivo formado por componentes físicos y computacionales, divididos en cuatro subsistemas:

- Locomoción
- Percepción
- Razonamiento
- Comunicación

Las plataformas MOBILEROBOTS contienen componentes básicos para el sensamiento y navegación en un entorno real.

El desarrollo de software incluye: ARIA (Interfaz Robótica Avanzada para Aplicaciones) y ARNetworking, ambas publicadas bajo licencia pública GNU/Linux, con librerías en C, Java y Python.

ARIA con ArNetworking es una plataforma que permite la integración de software de control propio, ya que maneja los

detalles de bajo nivel de interacción cliente-servidor, incluidas las comunicaciones de red y serial, comandos y la información del servidor de procesamiento de paquetes, tiempo de ciclo, y multihilo, así como el apoyo de una variedad de accesorios y controles, tales como giroscopios, sonares, y otros. Estas librerías vienen con código fuente, lo que permite examinar el software y modificarlo para sensores y aplicaciones propias.

II-B. Odometría

Es el método mediante el cual se determina la posición del vehículo (plataforma móvil) tomando la información de los encoders. La odometría se basa en poder obtener el desplazamiento realizado por la rueda asociada a un motor a partir de la medición de las vueltas realizadas por el mismo.

II-C. PYTHON

Python es un lenguaje de programación que tiene una sintaxis clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, hacen que desarrollar una aplicación en Python sea sencillo.

La sintaxis en Python es cercana al lenguaje natural, por este motivo es una de los mejores lenguajes para empezar a programar.

II-D. CONTROL DIFUSO

La teoría de conjuntos difusos permite representar el ser miembro de un conjunto como una distribución de posibilidades. La lógica difusa usa expresiones que no son ni ciertas ni falsas, es decir la lógica aplicada a conceptos que pueden tomar cualquier valor de veracidad dentro de un conjunto de valores intermedios.

La lógica difusa trabaja con conjuntos difusos, los cuales están definidos por sus funciones de pertenencia, la cual expresa la distribución de verdad de una variable.

Un conjunto difuso se puede definir matemáticamente al asignar a cada posible individuo que existe en el universo de discurso, un valor que representa su grado de pertenencia en el conjunto difuso. El grado de pertenencia indica cuando el elemento es similar o compatible con el concepto representado por el conjunto difuso.

La función de pertenencia se establece de una manera arbitraria, basándonos en la experiencia del usuario y contexto del problema.

II-D1. Controlador Difuso: Un sistema de control es un arreglo de componentes físicos conectados de tal manera que el arreglo se pueda manipular, dirigir o regular a sí mismo o a otro sistema.

El siguiente diagrama de bloques se muestra un controlador difuso, donde se muestra un control difuso en un sistema de control de bucle cerrado.

El controlador difuso tiene 4 componentes principales:

- Etapa de Fuzificación, que convierte las entradas del controlador en la información que el mecanismo de

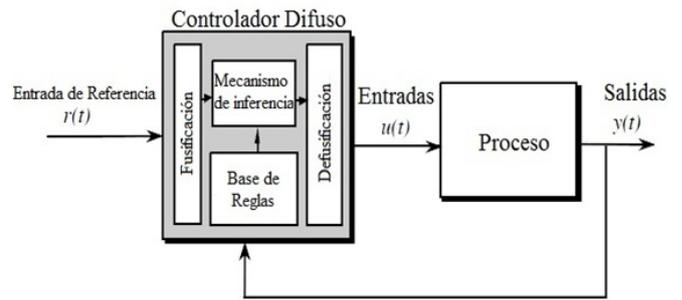


Figura 2. Arquitectura del controlador difuso

inferencia puede utilizar para activar y aplicar reglas. Como resultado de la fuzificación se obtiene valores lingüísticos medidos.

- La Base de reglas (conjunto de reglas si-entonces), que contiene una cuantificación lógica difusa de la descripción lingüística del experto de cómo lograr un buen control.
- El mecanismo de inferencia, que emula la decisión del experto en la interpretación y aplicación del conocimiento sobre la mejor manera de controlar la planta.
- Defuzificación, que convierte las conclusiones del mecanismo de inferencia en las entradas reales para el proceso.

II-E. LOCALIZACIÓN Y MAPEO DE ENTORNOS SIMULTÁNEOS (SLAM)

La localización y el mapeo simultáneo SLAM (Simultaneous Localization And Mapping), plantea la posibilidad para una plataforma móvil de situarse en un posición desconocida dentro de un sistema desconocido y levantar de forma incremental un mapa del entorno mientras simultáneamente determina su posición utilizando el mapa.

El esquema del problema del SLAM consta de los siguientes pasos:

1. Adquirir la información sensorial.
2. Detectar los puntos de referencia marcados para identificar los puntos de interés del entorno.
3. Establecer correspondencias entre lo observado y lo esperado.
4. Cálculo de la posición.

II-E1. Tipos de Mapas: El tipo de mapa que se va a levantar mediante la resolución de la localización y el mapeo simultáneo es muy importante, tanto como el método que se lleva a cabo para la resolución del mismo.

El SLAM (Localización y Mapeo Simultáneo) es un sistema de navegación basado en la construcción de mapas. Una buena representación debe ser lo suficientemente detallada para que la plataforma móvil pueda localizarse y navegar de forma autónoma.

II-E1a. Mapas Métricos: Sirven para localizar la plataforma robótica con una gran precisión y delinear un camino en presencia de obstáculos, pero requieren mucho espacio de memoria. Se dividen en dos categorías principales:

- Mapas creados a partir de hitos: Describen el entorno como un conjunto de marcas localizadas espacialmente.

Las ventajas de esta representación es su compacidad (compacto), lo que hace que sea muy útil para entornos de tres dimensiones y sistemas de dos dimensiones de gran extensión

- Mapas de densidad o de malla: Se usan cuando se necesita una alta resolución para planificar una ruta con exactitud, o el entorno tiene una estructura libre. Este tipo de mapas se desarrollan para celdas de dos o tres dimensiones, donde se guarda la probabilidad de estar libres u ocupadas.
 - Una de las principales características es la capacidad de representar entornos no estructurados, y se puede incrementar la exactitud cambiando la resolución de la celda.
 - Las celdas son independientes debido a esto no se puede expresar una relación entre las mismas.
 - Se construyen habitualmente ocupando sensores laser y sonares ya que es necesario obtener la medida de la distancia y la orientación.

Para este proyecto se desarrolló una variación de los mapas de densidad o de malla

III. DESARROLLO DE LA APLICACIÓN DE TELEOPERACIÓN Y ADQUISICIÓN DE DATOS

III-A. Diagrama del Sistema

La (Fig. 3.) muestra el diagrama de conexión entre cliente y servidor, además se observa los programas utilizados por cada uno de estos para el posterior desarrollo de las aplicaciones de Teleoperación, Adquisición de Datos, Control Difuso de Velocidad y Mapeo de Entornos.

El lenguaje de programación utilizado para el desarrollo de las aplicaciones es Python. Para el desarrollo de la interfaz gráfica de usuario de las diferentes aplicaciones se utiliza GTK (The GIMP Toolkit) y para el Mapeo de Entornos OpenCv.

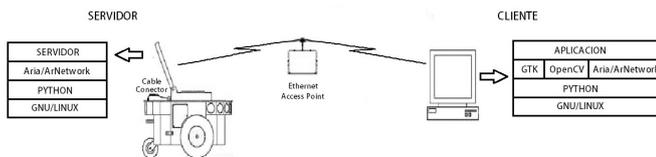


Figura 3. Diagrama General del Sistema

III-B. SERVIDOR

Al implementar las aplicaciones de Control Difuso de Velocidad y Mapeo de entornos, se produjeron errores. Se debe tomar en cuenta que muchos de los errores que se producen no son producidos por mal programación de las aplicaciones, sino por problemas de que ArNetwork no está correctamente implementado para funcionar con Python, uno de estos ejemplos es el comando ArMutex que a pesar de utilizar los ejemplos proporcionados por el fabricante no funcionaban en Python mostrando un error, es por esta razón que se implementó una librería de comunicación propia y por ende

se realizó dos programas de servidores uno implementado utilizando la librería de comunicación de Aria (ArNetworkingPy) y otro servidor que usa una librería de comunicación propia.

III-B1. SERVIDOR CON ARNETPACKET: La aplicación servidor permite utilizar la plataforma móvil de manera remota a través de los paquetes ArNetPacket, que son transmitidos y recibidos por el cliente, los paquetes ArNetPacket contienen información transmitida por el cliente o por el servidor. Las librerías y funciones del servidor son mostradas por la (Fig. 4.)

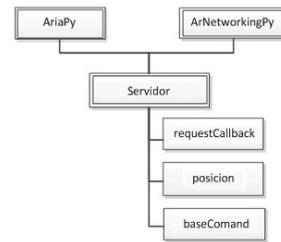


Figura 4. Esquema de funciones del programa Servidor

El servidor a pesar de ser exclusivo de las librerías ArNetworkingPy se necesita utilizar AriaPy para poder conectar la plataforma móvil al servidor. El servidor dispone de dos comandos adicionales del listado base de ArNetPacket que son requestCallBack y posición estos sirven para probar la conexión con el servidor y adquirir los valores del sonar en coordenadas X,Y,Th respectivamente.

El funcionamiento del servidor se puede describir como dos procesos que funcionan en hilos, estos dos procesos son el núcleo base del servidor y el objeto robot, por lo tanto el programa al final debe mantenerse en espera para poder trabajar con el servidor, esto se puede apreciar en la (Fig. 5.)

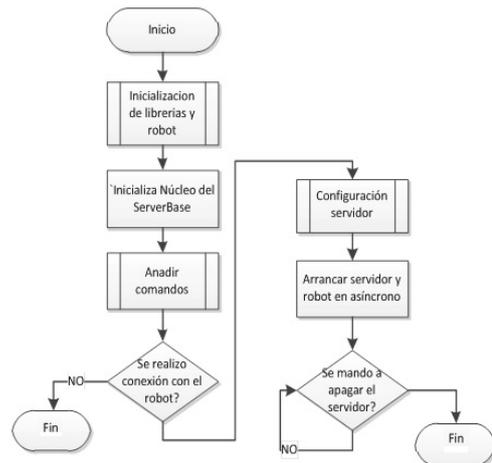


Figura 5. Diagrama de Flujo del programa Servidor

Como aprecia en la (Fig. 5.) se realiza la inicialización de las librerías y el objeto robot, después se inicializa el

núcleo base del servidor que activa los parámetros básicos (comandos, parámetros de conexión), después se añaden comandos propios para poder realizar funciones adicionales, se verifica la conexión con el robot, configuración del servidor con las variables del robot, se ejecuta el servidor y el robot en hilos para que funcionen al unísono, al final se mantiene en ejecución la aplicación para poder trabajar con esta durante toda la operación de la plataforma móvil.

La primera subfunción es la inicialización de los elementos básicos y necesarios para poder comenzar a trabajar con la plataforma móvil como se puede apreciar en la (Fig. 6.)



Figura 6. Inicialización de librerías y robot

Lo primero es inicializar Aria esto nos permitirá poder trabajar con el robot y todas sus funciones, luego se debe crear el objeto robot y el objeto dispositivo sonar a este último se le debe añadir el objeto robot para que pueda realizar y adquirir medidas de los sonares de la plataforma móvil. Lo siguiente es crear los comandos y estructuras para los paquetes ArNetPacket.

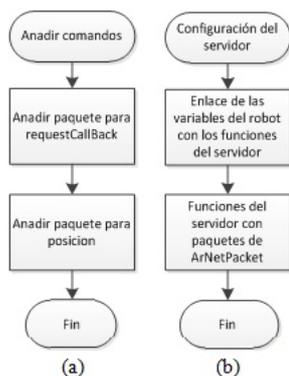


Figura 7. Comandos y paquete ArNetPacket

En la (Fig. 7a.) el diagrama de flujo es para explicar que se debe crear el esqueleto del paquete ArNetPacket y enlazarlo con una función en el programa. En Añadir el paquete para requestCallBack se incluye la información de requestCallBack al paquete, el mismo que sirve para probar la conexión con el robot, de manera similar se hace en Añadir paquete para

posición que se utiliza para adquirir los valores del sonar. Este caso es un ejemplo de cómo se realizó las funciones del servidor con ArNetPacket.

En la figura (Fig. 7b.) en cambio se enlazan las variables y valores de sensores del objeto robot a la función que esta enlazada al esqueleto del paquete.

III-B1a. PAQUETE ARNETPACKET: El paquete ArNetPacket sirve para comunicar el cliente con el servidor, dentro de este paquete se encuentran los parámetros y comandos para dar las órdenes a la plataforma móvil como los valores devueltos por esta.

El servidor y cliente funcionan a través de dichos paquetes. Al ser todos estos datos realizados para C al trabajarlos en Python es necesario darles un valor antes de utilizarlos como es el caso del parámetro String, ya que para Python no es necesario declarar el tipo de una variable solo se tiene que darle un valor para que este sepa a que pertenece, por lo tanto al ARIA y ARNETWORKING al ser realizados en C se tiene que acomodar la manera en cómo se utiliza Python con variables para no tener problemas ya que todas las funciones de estas librerías se encuentran escritas en C y se realiza una importación desde C a Python.

III-B2. SERVIDOR SIN ARNETPACKET: Como se puede ver en la (Fig. 8.) lo primero que se realiza es llamar a la clase NoPyArNetworking que es la encargada de tomar los datos de la plataforma robótica como son la odometría, sonares, motores y funciones de movimiento, después de esto se leen los datos adquiridos. Luego de esto se verifica la Dirección IP del servidor y el número de puerto, si los parámetros son los correctos se ejecuta las funciones NoPyArNetworking.robot y NoPyArNetworking.server, que son parte de la clase NoPyArNetworking.

Si la Dirección IP o el número de puerto no son los correctos, se muestra un mensaje y luego se setea la Dirección IP como localhost (de la máquina donde corre el programa) y el número de puerto con 74744 y se ejecuta las funciones NoPyArNetworking.robot y NoPyArNetworking.server.

A continuación se muestra la clase NoPyArNetworking con sus funciones respectivas ver (Fig. 9.).

La función `__init__` inicia las variables que van a contener los valores que envíen las funciones de *sonares* y *odometria*. La función *motores* es la encargada como su nombre mismo lo indica de prender o apagar los motores de la plataforma robótica. La función *mover* sirve para el movimiento y velocidad de la plataforma. La función *sonares* se usa para la triangulación de objetos. La función *odometria* sirve para adquirir los datos de posición de la plataforma en coordenadas (x,y,Th).

III-C. CLIENTE

El cliente fue realizado con funciones para: iniciar la conexión, enviar paquetes, leer paquetes y cerrar conexión, con la finalidad de que pueda ser utilizado por todas las aplicaciones como son: Teleoperación, Adquisición de Datos,

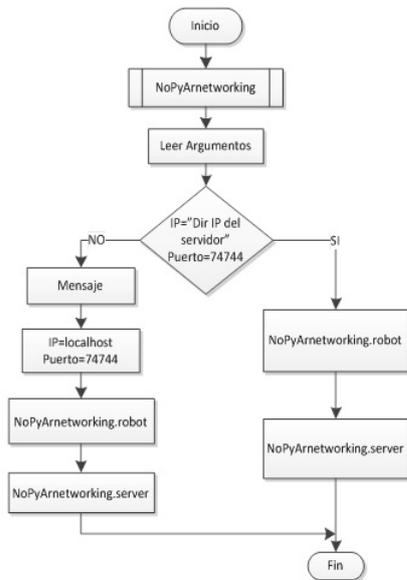


Figura 8. Diagrama de flujo del programa servidor sin ArNetWorking

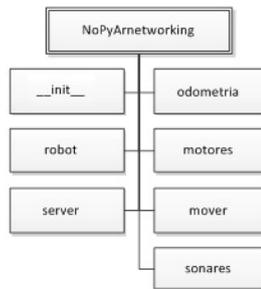


Figura 9. Esquema de funciones de NoPyArNetworking

Control Difuso de Velocidad y Mapeo de Entornos, sin tener que realizar modificaciones en el código. Por lo tanto las aplicaciones solo deben importarlo y llamar a la función que se va a utilizar, ver (Fig. 10.)

III-D. APLICACIÓN DE TELEOPERACIÓN

Las aplicaciones están divididas en dos archivos el archivo xml de gtk2 que es el que contiene la información de la interfaz gráfica y el script de Python, al realizar la aplicación de esta manera se facilita el script tanto en la comprensión como en su desarrollo ya que todo se vuelve señales de aviso tomadas de la interfaz gráfica al script.

La aplicación de Teleoperación permite realizar teleoperación básica de la plataforma móvil, tiene una interfaz simple ya que para el caso propuesto es necesario utilizar solo el movimiento.

La aplicación está estructurada en la siguiente forma:

- Clase prueba_teleoperacion
- Ejecución de gtk

La clase *prueba_teleoperacion* tiene la función `__init__` que es la encargada de cargar el archivo xml de gtk, las señales

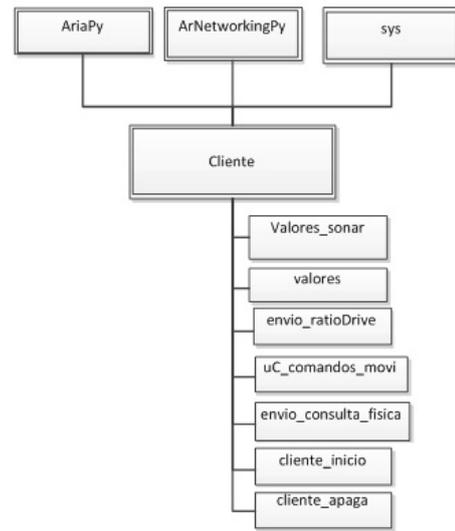


Figura 10. Esquema de funciones del programa Cliente

y widgets de la interfaz gráfica y el resto de funciones de la clase son solo para trabajar con la activación de las señales de gtk y realizar una acción según la señal, ver (Fig. 11.)

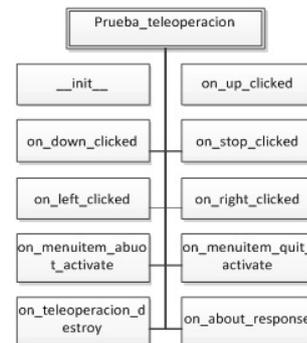


Figura 11. Esquema de funciones de la aplicación de Teleoperación

Por ejemplo para el caso de la función `on_stop_clicked` si el programa detecta la activación de esta función se procede a enviar un paquete `ArNetPacket` con el comando stop que realiza un paro de emergencia y así ocurre con el resto de funciones.

III-E. APLICACIÓN DE ADQUISICIÓN DE DATOS

La aplicación de adquisición de datos permite realizar lectura de los sonares de la plataforma móvil, es necesario tener estos valores para que puedan ser utilizados por el controlador difuso y el generador de mapas.

El funcionamiento de esta aplicación es similar al de la Teleoperación excepto por los paquetes de `ArNetPacket` que envía, ver (Fig. 12.)

Por ejemplo la función `on_boton1_clicked`, utiliza la librería `cliente_lib`. Toma los valores de los sonares, los acomoda en

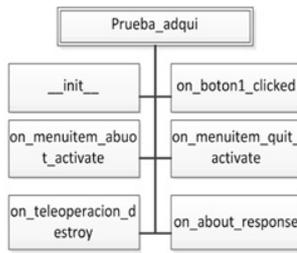


Figura 12. Esquema de funciones de la aplicación de Adquisición de Datos

una estructura y los convierte en texto para poder visualizarlos en la interfaz de usuario.

Ambas aplicaciones pueden funcionar al unísono por que el servidor no se bloquea al tener una conexión activa, permitiendo probar al mismo tiempo el movimiento y adquisición de datos para pruebas.

IV. CONTROLADOR DIFUSO DE VELOCIDAD

IV-A. DISEÑO DEL CONTROLADOR DIFUSO

Lo que se pretende lograr es que mientras la plataforma se encuentre a una distancia segura de un obstáculo vaya a una velocidad alta, teniendo en cuenta que la mayor velocidad a la que puede llegar la plataforma es de 1.2 m/s; a medida que la plataforma vaya acercándose al obstáculo esta irá disminuyendo la velocidad hasta que finalmente pare si es que se encuentra muy cerca del obstáculo.

El control difuso para la velocidad de la plataforma Pioneer 3-DX funciona en base al Set Point de entrada que va a ser la distancia con respecto al obstáculo más cercano a la que se quiera que la plataforma frene.

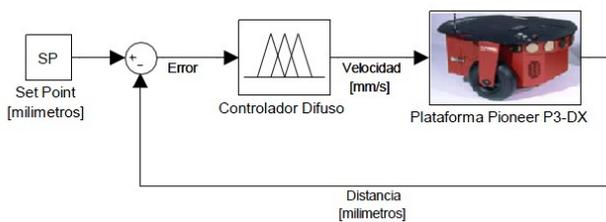


Figura 13. Diagrama del controlador difuso

IV-A1. Rango de operación de las variables de entrada y salida: Para realizar el control de velocidad se ha tomado en cuenta una variable de entrada y una variable de salida. La variable de entrada es: El error que está dado por el rango de funcionamiento de los sonares, los cuales tienen un alcance de 0,12 - 5 m. El error va a estar dado por el Set Point que es la distancia a la cual se quiera que pare el robot con respecto a un obstáculo menos la distancia a la que se encuentre la plataforma de un obstáculo. La variable de salida está dada por la velocidad máxima que puede alcanzar la plataforma: 0 - 1,2 m/s.

Variable de Entrada	Variable de Salida
Error	Velocidad

IV-A2. Definición de las funciones de pertenencia: Se definen los adjetivos y rangos para cada una de las funciones de pertenencia.

El Error se define en 5 valores lingüísticos:

Error	Rango
Muy Grande Negativo (MNG)	[-5000 3000]
Grande Negativo (GN)	[-4500 -3000 -2000]
Negativo (N)	[-3000 -2000 -1000]
Cero (Z)	[-2000 -1000 0]
Positivo (P)	[-1000 1000 5000]

En (Fig. 14.) se muestra la función de pertenencia de la variable de entrada Error, está dada en [mm].

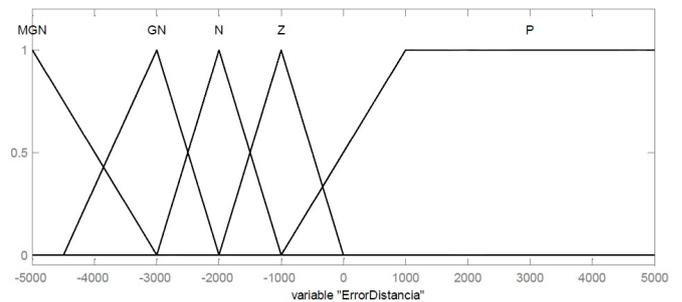


Figura 14. Función de pertenencia de la Variable Error

La variable de salida del controlador es la velocidad, la cual se define en 5 valores lingüísticos:

Velocidad	Rango
Velocidad Muy Baja (VMB)	[-120 255]
Velocidad Baja (VB)	[30 355 600]
Velocidad Media (VM)	[255 -600 950]
Velocidad Alta (VA)	[600 875 1130]
Velocidad Muy Alta (VMA)	[950 1300]

La (Fig. 15.) muestra la función de pertenencia de la variable de salida Velocidad, dada en [mm/s].

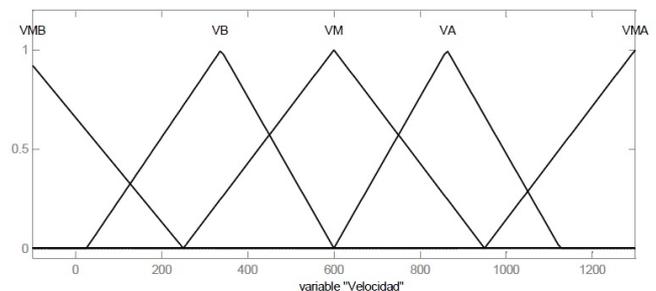


Figura 15. Función de pertenencia de la Variable Velocidad

IV-A3. *Desarrollo de la Base de Reglas:* En las reglas se tiene una representación implícita del modelo, por lo que de ellas se puede seguir el comportamiento aproximado del modelo. Por lo tanto se definen las reglas que determinan el comportamiento del sistema.

Nro	REGLA
1	Si(Error es MGN) Entonces (VMA)
2	Si(Error es GN) Entonces (VA)
3	Si(Error es N) Entonces (VM)
4	Si(Error es Z) Entonces (VB)
5	Si(Error es P) Entonces (VMB)

En la (Fig. 16.) se muestra una gráfica de Distancia Vs Velocidad para un Set Point de 1.5 metros.

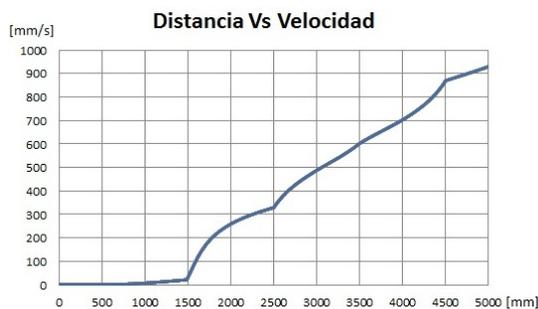


Figura 16. Distancia Vs Velocidad para un Set Point de 1.5 m

V. DESARROLLO DE LA APLICACIÓN PARA MAPEO DE LA PLATAFORMA MÓVIL

V-A. LOCALIZACIÓN

Para la localización se utiliza la odometría, es decir se leen los encoders que se encuentran en las llantas de la plataforma móvil, con esto se reciben datos que entregan el desplazamiento que ha tenido la plataforma con respecto al punto de inicio.

Los sensores miden la posición y orientación del robot. Donde X y Y representan la posición de la plataforma y Th la orientación del robot. Estas coordenadas del sistema odométrico están referenciadas a un sistema absoluto externo, fijo, coincidente con la posición y orientación del robot cuando se enciende.

Desde esas lecturas y conocidas las coordenadas de su posición anterior ($x(t - 1)$, $y(t - 1)$, $Th(t - 1)$) se calcula la nueva posición. Estos cálculos los hace el sistema operativo del microcontrolador, que realiza la conversión a coordenadas absolutas X, Y, Th.

V-B. MAPEO

Para realizar el mapeo del entorno se utilizan los sonares, los cuales entregan la información de la distancia a la que se encuentran los obstáculos, estas medidas son necesarias para construir los mapas del lugar en el que se encuentra la plataforma móvil. Para renderizar la imagen en dos dimensiones del entorno se ha utilizado OpenCV.

El área que se puede mapear es de 25 x 17.5 metros, es decir desde el punto donde fue encendida la plataforma 12.5 m en el eje Y positivo, 12.5m en el eje Y negativo, 8.75 m en el eje X positivo y 8.5m en el eje X negativo, lo que da un área total de 437.5 m², área suficiente para mapear una habitación o un pasillo. La (Fig. 17.) ilustra el sistema de coordenadas, inicio de la plataforma y medidas del mapa.

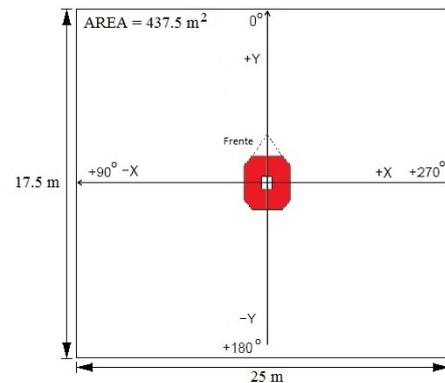


Figura 17. Sistema de coordenadas, inicio de la plataforma y medidas del mapa

V-C. MAPA

El tipo de mapa que este trabajo presenta es una variación del Mapa de Densidad o de Malla que a su vez es una subclase de un mapa Métrico; ya que captura las características geométricas del entorno, y las representa como un conjunto de objetos cuyas coordenadas están definidas en un plano cartesiano. Este método define una matriz de NxM celdas que representa una cierta región de igual tamaño, asociadas a su localización real según la posición en la matriz. Cada celda puede estar ocupada o vacía, según haya o no un obstáculo;

Para generar el mapa del entorno lo primero que se hace es crear una imagen en blanco del 500 x 250 píxeles en OpenCV, luego se grafican en el visor las coordenadas dadas por la plataforma como círculos de 2 píxeles de radio esto se hace con la finalidad de que se pueda visualizar con mayor facilidad, finalmente se guarda como una imagen.

V-D. APLICACIÓN DE MAPEO DE ENTORNOS

La aplicación para desarrollar la generación del entorno y la localización, necesita de las aplicaciones de Teleoperación y Adquisición de datos.

A continuación se detalla el proceso mediante el cual se logró obtener el mapa del entorno.

La (Fig. 18.) muestra como se comenzó desarrollando la aplicación para esto se utilizó el simulador de la plataforma móvil (4). Se va generando el mapa en un visor (3) a través de la librería OpenCv. Se puede ver que para mover la plataforma se necesita de la aplicación de Teleoperación (2) y se usa también la aplicación de Adquisición de Datos (1) para la lectura de sonares que sirve para tomar los puntos donde se encuentran los obstáculos que después son graficados en el

visor. Aquí únicamente se muestra un cuadrante del mapa ya que se asume que el origen del robot es el punto 0,0 es decir que como se mostró anteriormente en la (Fig. 17.) la plataforma puede trabajar con valores positivos o negativos en las coordenadas X y Y tomando en cuenta que el punto donde fue encendida la plataforma es el punto 0,0. El visor en cambio únicamente representa valores positivos de las coordenadas X y Y, es por esto que en esta parte del desarrollo de la aplicación de mapeo de entornos únicamente se grafican los puntos positivos que envía la plataforma, que vienen a formar el primer cuadrante.

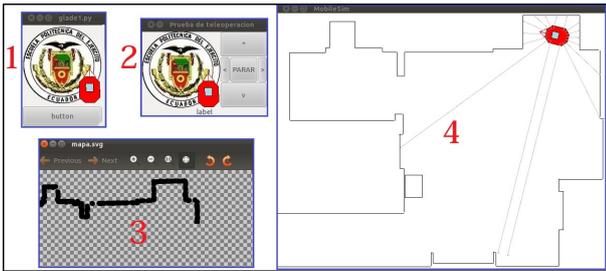


Figura 18. Sistema de coordenadas, inicio de la plataforma y medidas del mapa

La (Fig. 19.) muestra ya un mapa completo generado y la plataforma localizada en el mismo, esto se llevó a cabo con el simulador de la plataforma móvil, se puede observar que ya se genera el mapa completo con los cuatro cuadrantes tomando en cuenta que la plataforma móvil comenzará en el centro de la imagen para hacer esto se mueve el sistema de coordenadas, evitando de esta forma que existan coordenadas negativas y poder así graficar el mapa en todos sus cuadrantes.

Las ecuaciones utilizadas para esto son:

$$X_{pixel} = \frac{X_{real}}{50} + 500 \quad (1)$$

$$Y_{pixel} = \frac{Y_{real}}{50} + 350 \quad (2)$$

Dónde:

X_{pixel} = Número de pixeles del mapa creado en el eje X

Y_{pixel} = Número de pixeles del mapa creado en el eje Y

X_{real} = Medida en milímetros del entorno real en el eje X

Y_{real} = Medida en milímetros del entorno real en el eje Y

En estas ecuaciones se puede notar por tanto que la resolución del mapa creado es de 1 pixel = 50 mm. Para graficar el mapa en el centro del visor se hace un desplazamiento de 500 pixeles en X y de 350 pixeles en Y.

Para localizar la plataforma se utiliza los datos tomados de la odometría y se los plasma en el mapa, cada vez que la plataforma se mueve se toma los datos y se actualiza el mapa, lo cual da la idea que el robot se mueve dentro del mapa.

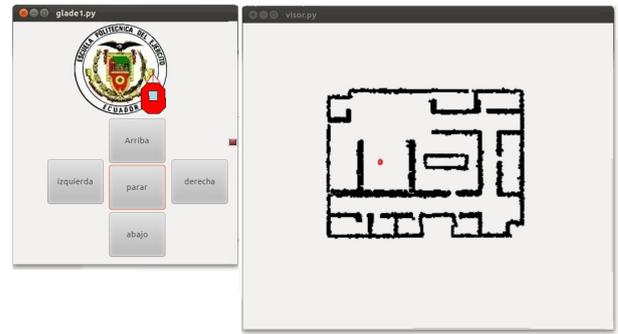


Figura 19. Etapa final para la elaboración del mapa (probado en simulador)

El funcionamiento de la aplicación de Mapeo de Entornos es similar al de la aplicación de Teleoperación, a continuación se muestra un esquema de funciones del programa.

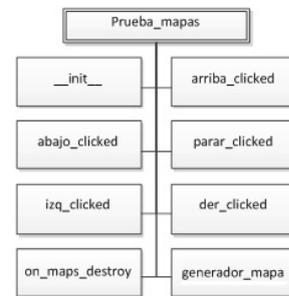


Figura 20. Esquema de funciones para la aplicación de Mapeo de Entornos

La función *generador_mapa* utiliza la librería *renderizado* y la librería *cliente_lib* para generar el mapa. La librería *renderizado* se la estableció para crear el mapa a través de OpenCv.

VI. APLICACIÓN FINAL TELVEMAP Y APLICACIÓN DE NAVEGACIÓN AUTÓNOMA

VI-A. Librería de Comunicación Propia

La librería de Comunicación se implementó con el afán de reemplazar la librería de comunicación ArNetworking de ARIA que presentó problemas al momento de ejecutar las aplicaciones creadas. Se la realizó para trabajar con el protocolo TCP/IP.

Cabe indicar que toda la comunicación se realiza a través de sockets, por lo que para esto es necesario definir una dirección IP y un puerto de comunicación TCP, tanto para el cliente como para el servidor

A continuación se muestra una tabla de los mensajes que envía el cliente al servidor y los acuses de recibo que envía el servidor al cliente.

Mensajes Cliente	Respuesta Servidor
adelante, 5	adelante
atras, 1200	atras
izquierda,456	izquierda
derecha, 763	derecha
sonares (x1,y1)	(x1,y1) Listo
(x2,y2)	(x2,y2) Listo
⋮	⋮
(x7,y7)	(x7,y7) Listo
odometria	(x,y,Th)

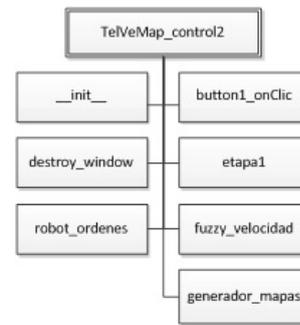


Figura 23. Esquema de Funciones del programa TelVeMap

VI-B. APLICACIÓN FINAL TELVEMAP

TelVeMap es la aplicación donde convergen las aplicaciones de Teleoperación, Adquisición de Datos, Control Difuso de Velocidad y Mapeo de Entornos.

Debido a que la aplicación TelVeMap, usa un mando (palanca) conectado al cliente para el movimiento de la plataforma, se presenta la configuración del mando ver (Fig. 21.)



Figura 21. Configuración de Mando para la aplicación TelVeMap

El botón Hombre Muerto, se utiliza como un sistema de seguridad de la plataforma robótica.

La Fig. 22. muestra la interfaz de la aplicación TelVeMap.

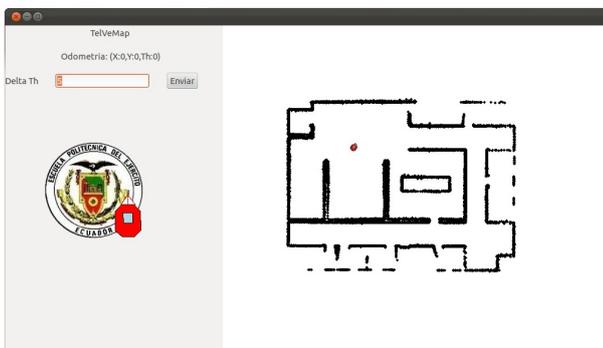


Figura 22. Interfaz de la aplicación TelVeMap

La (Fig. 23.) muestra el esquema de funciones de la aplicación TelVeMap.

La función `__init__` se encarga de inicializar las variables. La función `button1_onClic` sirve para ajustar la coordenada `Th` en el gráfico para que coincida con el entorno real al momento de iniciar la aplicación. La función `Etapa1` es un objeto timer que se ejecuta cada 300 m/s y llama a la función `robot_ordenes` que es la encargada de manejar la palanca para el movimiento de la plataforma, esta función `robot_ordenes` a su vez llama a las funciones `fuzzy_velocidad` encargada del control de velocidad difuso y `generador_mapas` que se usa para realizar el mapeo de entornos.

VI-C. APLICACIÓN DE NAVEGACIÓN AUTÓNOMA

La aplicación de Navegación Autónoma se implementa en el equipo servidor. Esta aplicación realiza la navegación del entorno de forma independiente y además incluye el control difuso de velocidad. La Fig. 24. muestra el esquema de funciones del programa.



Figura 24. Esquema de funciones del programa de Navegación Autónoma

La función `Actions` es la encargada del movimiento de la plataforma robótica, por lo que esta sirve para realizar los movimientos y giros de la misma. La función `Task` es la encargada de llamar al control difuso de velocidad y a los datos de velocidad que recibe del controlador los escribe en un archivo de texto, esto posteriormente sirve para tener una base de datos de la velocidad a la que fue la plataforma y permite generar también una gráfica del comportamiento de velocidad.

VII. PRUEBAS Y RESULTADOS

Esta prueba se la llevó a cabo dentro del laboratorio, para simular las paredes de madera se puso láminas de plywood conjuntamente con las paredes de hormigón del laboratorio.

El área tomada por la plataforma móvil fue de aproximadamente 4.35 metros de largo por 2.45 metros de ancho, lo que da un área de 10.66 m².

En la figura (Fig. 25.) se muestra las medidas reales en metros del escenario planteado.

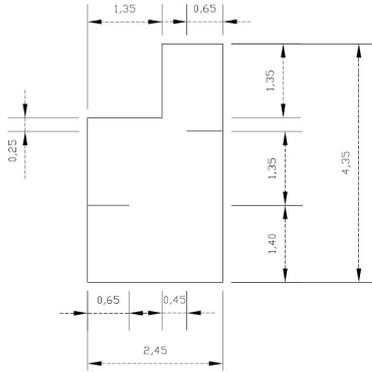


Figura 25. Medidas reales del escenario planteado

En la (Fig. 26.) se aprecia el entorno donde se realizó la prueba, y el mapa de entorno obtenido por la plataforma, los puntos dispersados que no corresponden al entorno se producen por los errores que presentan los sonares en ambientes reales, ya que en pruebas realizadas en simulación no sucede esto.

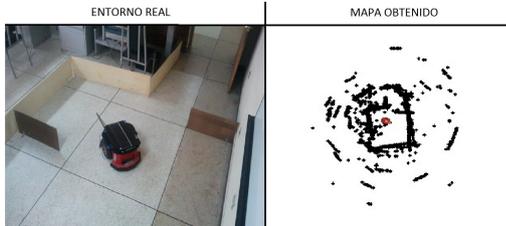


Figura 26. Pruebas en un entorno real

Para establecer un parámetro que permita cuantificar el grado de aproximación de la representación computacional con respecto a las medidas del entorno real, se define el parámetro Nivel de Aproximación del Mapa al Entorno Real (NAMER), que está dado por:

$$NAMER = \frac{D_{map}}{D_{real}} \quad (3)$$

Dónde:

D_{map} : Distancia calculada sobre el mapa.

D_{real} : Distancia real del entorno.

En la figura (Fig. 27.) se muestra las medidas en pixeles tomadas del mapa del entorno creado, esto se hace con la finalidad de poder tener una idea de la medida en metros del mapa obtenido, teniendo en cuenta que 1 pixel = 50 mm.

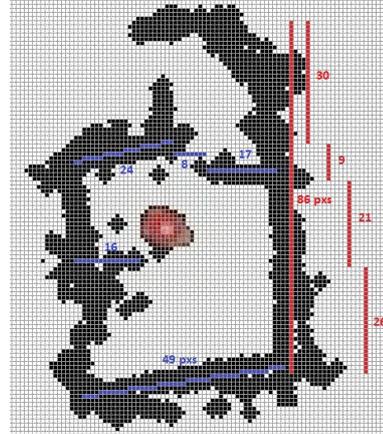


Figura 27. Medidas del mapa obtenido en pixeles

En la tabla siguiente se muestra el nivel de aproximación del mapa al entorno real. Se debe tomar en cuenta que para obtener el parámetro Medida del Mapa Obtenido en metros se utiliza la Ecuación (1) y (2) sin tomar en cuenta el desplazamiento. Se debe multiplicar el número de pixeles por 50 [mm] y el resultado se divide para 1000 para transformar a metros.

	Medida Real [m]	Número de Pixeles	Medida del Mapa Obtenido[m]	Parámetro NAMER
Medida 1	4,35	86	4,3	0,989
Medida 2	1,35	30	1,5	1,111
Medida 3	0,25	9	0,45	1,800
Medida 4	1,35	21	1,05	0,778
Medida 5	1,4	26	1,3	0,929
Medida 6	2,45	49	2,45	1,000
Medida 7	0,65	16	0,8	1,231
Medida 8	0,7	8	0,4	0,571
Medida 9	0,45	8	0,4	0,889
Medida 10	0,65	17	0,85	1,308
			Grado de Confiabilidad	1,060

Como se puede observar el grado de confiabilidad obtenido es muy cercano a 1, por lo que las medidas en metros del mapa obtenido son muy cercanas a las reales.

VIII. CONCLUSIONES

- Mediante este proyecto se ha logrado implementar satisfactoriamente las aplicaciones de Teleoperación, Adquisición de Datos, Control de Velocidad y Mapeo de Entornos para la plataforma robótica Pioneer P3-DX mediante el lenguaje de programación Python y el sistema operativo GNU/Linux.
- La ventaja de utilizar sistemas basados en GNU/Linux es que se puede manipular el sistema de tal manera que

solo se ejecute lo que se necesita, permitiendo utilizar un computador a bordo con menores prestaciones y teniendo un rendimiento igual a un computador de superiores características que utilice Windows.

- La aplicación de Mapeo de Entornos es eficiente aunque presenta limitaciones al mapeo de paredes debido al error que presentan los sonares en las esquinas, a pesar de todo cumple su función de realizar mapas en 2D, aunque cabe recalcar que no se comporta de la misma forma en la simulación como en el entorno real debido al error que presentan los sensores en ambientes reales.
- Un inconveniente que se observó en el manejo de ARIA con Python, es que ARIA no permite manejar el comando ArMutex que sirve para dormir los hilos y trabajar con distintas aplicaciones y procesos a la vez. Llegan a provocar un desbordamiento de memoria en las diferentes librerías de C que utiliza PyAria y PyNetworking.

IX. RECOMENDACIONES

- Los mapas se pueden mejorar realizando procesamiento de imágenes, para reducir el grosor de las líneas y mejorar la forma, es decir líneas más rectas y planas.
- Debido a que ARIA no permite el manejo del comando ARMUTEX con Python, se puede utilizar lenguaje C para el desarrollo de las aplicaciones.
- Como trabajo futuro se puede utilizar teléfonos inteligentes como dispositivos hápticos ya que estos disponen de acelerómetro, giroscopio y pantalla para realizar la inmersión de una cámara o hacer mapas del entorno realizando una conexión a la plataforma robótica con los protocolos HTTP.

REFERENCIAS

- [1] MOBILEROBOTS.INC, *Pioneer3 Operations Manual*, 75 págs, 2007.
- [2] MORALES JIMENA, JARAMILLO DANIEL , *Desarrollo de Aplicaciones y Documentación de las Plataformas Robóticas Pioneer P3-DX y Pioneer3 P3-AT*, 265 págs, 2010.
- [3] PASSINO KEVIN, *Fuzzy Control*, 522 págs, 1998.

Desempeño de Algoritmos Gusano en la Navegación de una Plataforma Robótica Móvil

Luis M. Álvarez, Juan E. Figueroa, Rodolfo X. Gordillo y Luis A. Chacón

Resumen— El presente trabajo presenta la implementación de algoritmos de navegación denominados Gusano 1 y Gusano 2 sobre la plataforma robótica móvil de iRobot Create, los cuales le brindaron la capacidad de autonomía para navegar dentro de entornos desconocidos. Además, se utilizaron a los módulos de radio frecuencia XBee, que funcionan bajo el estándar ZigBee, para dotar de comunicación inalámbrica entre el robot y una PC. Se realizaron pruebas de comunicación para determinar la máxima distancia de trabajo a la que pueden funcionar los módulos de RF en interiores; mientras que, para el análisis de desempeño de los algoritmos básicos de navegación montados sobre la plataforma robótica móvil se definieron tres escenarios diferentes donde se verificó la certeza de haber alcanzado la meta por el robot.

Palabras clave—Navegación; Algoritmos Gusano; IEEE 802.15.4; ZigBee; certeza.

I. INTRODUCCIÓN

En la actualidad, la Robótica está orientándose a los entornos inciertos e impredecibles dejando a un lado los lugares estáticos; lo que ha conducido al reemplazo de los tradicionales manipuladores robóticos por robots móviles. La utilización de robots móviles en aplicaciones industriales y técnicas es amplia y continúa creciendo.

En función de los avances producidos en el campo computacional, tecnologías de comunicación, sensores y dispositivos de navegación, los robots móviles han desarrollado su capacidad de autonomía. Estos robots son capaces de modificar sus comportamientos autónomamente al momento que se origine cualquier cambio en su entorno, es decir, que no requieren de la intervención de una persona para tomar alguna decisión porque la reforma de sus comportamientos está en base a la percepción del entorno real. Por lo tanto, los robots móviles autónomos deberían ser dispositivos capaces actuar de forma independiente y adaptar sus comportamientos en un entorno dinámico e incierto.

L.M. Álvarez, es graduado de Ingeniero en Electrónica, Automatización y Control en la Universidad de las Fuerzas Armadas Espe, Sangolquí-Ecuador, (email:juane_figueroa@hotmail.com)

J.E. Figueroa, es graduado de Ingeniero en Electrónica, Automatización y Control en la Universidad de las Fuerzas Armadas Espe, Sangolquí- Ecuador, (email:lmiguel_ar@hotmail.com)

R.X Gordillo, es profesor del Departamento de Eléctrica y Electrónica en de en la Universidad de las Fuerzas Armadas Espe, Sangolquí- Ecuador, (email:rxgordillo@espe.edu.ec).

L.A. Chacón, es profesor del Departamento de Eléctrica y Electrónica en de en la Universidad de las Fuerzas Armadas Espe, Sangolquí- Ecuador, (email:lachacon@espe.edu.ec).

Para interactuar de una manera óptima y correcta con el entorno, estos dispositivos autónomos pueden utilizar un conocimiento previo de éste o procesar la información entregada por sus sensores para construir una representación adecuada tanto de su entorno como de su ubicación

Es así que, uno de los principales retos de la Robótica actual radica en el hecho de generar un planeamiento de movimiento autónomo, en donde, el objetivo apunta a implementar una tarea en lenguaje de alto nivel que al ser compilada se convierta en un conjunto de movimientos primitivos de bajo nivel que deberán ser ejecutados por el robot para su consecución. Entonces, para aprovechar la movilidad presente en este tipo de robots se requiere desarrollar habilidades de navegación, que le permitan a un robot móvil llegar a su destino de manera segura sin importar condiciones estáticas o dinámicas de su entorno.

Sin embargo, algunos robots móviles, como el caso del iRobot Create, necesitan un controlador externo para dotarlos con autonomía. Generalmente, se recurre a la utilización de una PC para ello, la misma que se comunica con el robot a través de un cable. Al emplear un cable, el área de trabajo del robot móvil está limitada por la longitud de éste.

Como solución a lo anterior, el uso de módulos de comunicación inalámbrica resultan ser un buen recurso. En este caso particular se optó por seleccionar al estándar Zigbee como un camino alternante al que brinda Bluetooth.

Zigbee se establece como un estándar para aplicaciones en redes inalámbricas que requieran comunicaciones seguras, fiables, con baja tasa de envío de datos y maximización de la vida útil de las baterías de los dispositivos que conformen a este tipo de redes. El estándar Zigbee define principalmente a las capas de red y aplicación sobre las especificaciones del estándar IEEE 802.15.4 completando de esta manera al modelo OSI [2].

II. TRABAJOS RELACIONADOS

El desarrollo de algoritmos de navegación y el empleo de dispositivos que trabajan con el estándar ZigBee es poco conocido en nuestro país y también en desarrollo de aplicaciones como proyectos de grado universitario. Los proyectos relaciones nutren de una gran perspectiva del proyecto desarrollado.

En [3] y [4] se desarrollo la técnicas de localización utilizando un robot virtual mente ciego, el cual solo cuenta con sensores de contacto. En [5] implementan prototipos para limpieza y servicio en el hogar.

Se desarrolló el estudio de una red tipo Mesh para el control y cierre de cajas en un casino [6].

Finalmente, [7] se desarrolla el estudio de ZigBee en una aplicación de iluminación.

III. FUNDAMENTO TEÓRICO

El desarrollo de este trabajo tuvo dos directrices que conforman su fundamento teórico. A continuación se describen a cada una de ellas.

A. Navegación

La navegación en un robot es el proceso de planificar una trayectoria libre de obstáculos posible y guiarlo a través de ella para desplazarse desde una posición a otra en un área de trabajo.

Surgen entonces tres problemas por solucionar en relación a la navegación. El primero está relacionado con la auto-localización del robot, en donde, el robot recurre al conocimiento previamente adquirido y a sus observaciones locales, por medio de encoders acoplados a sus llantas, y externas, empleando sistemas de posicionamiento, para determinar su posición con relación a su entorno. La segunda problemática consiste en la identificación de la posición del objetivo con relación al entorno. Finalmente, el tercer inconveniente corresponde a la planificación de la trayectoria que permitirá establecer un camino posible desde la posición inicial a la posición del objetivo o meta esquivando obstáculos.

Un robot puede navegar en dos tipos de ambientes. Los entornos estructurados que permanecen estáticos y no sufren cambios con el transcurso del tiempo, y los entornos no estructurados que son dinámicos y pueden presentar cambios inesperados en los elementos que lo conforman. Con lo anterior, dentro de ambientes desconocidos o no estructurados, el robot deberá desarrollar capacidad para reaccionar ante la presencia de imprevistos en su trayectoria.

1) Algoritmos de navegación

En relación a la tarea de navegación se pueden considerar a dos clases de algoritmos que se enlistan a continuación:

- **Algoritmos fuera línea.** Para emplear un algoritmo fuera de línea se requiere una amplia información acerca del entorno, en la cual, se detalle con exactitud la localización de los obstáculos. El planeamiento de la trayectoria, en este caso, se realiza una sola vez y antes de que el robot móvil comience a desplazarse. De ahí su nombre de algoritmos fuera de línea. El problema en este tipo de algoritmos recae en contar con un proceso computacional eficiente, y no en garantizar que una trayectoria posible exista. Además, a los algoritmos fuera de línea se les pueden añadir criterios de optimización que den como resultado trayectorias en las cuales se tengan: un menor tiempo en atravesar un entorno, trayectorias que representen el camino más corto hacia la meta o simplemente trayectorias que sean las más seguras para el robot móvil.
- **Algoritmos fuera línea.** Al contrario de los anteriores algoritmos, la información del entorno que tienen los

algoritmos en línea es incompleta, por lo cual, la planeación de una trayectoria en estos métodos es un proceso continuo que para generar las acciones de control se recurre a la información del entorno entregada por los sensores del robot móvil. Al utilizar los sensores se retiran los requerimientos de conocer la localización y forma de los obstáculos, e inclusive de que éstos sean estacionarios.

La principal diferencia entre estas dos categorías de algoritmos radica en la cantidad de información que necesitan para su implementación. Tanto los algoritmos fuera de línea como los algoritmos en línea deben alcanzar la meta o determinar que el objetivo es inalcanzable tras el transcurso de un tiempo finito.

2) Propiedades del robot para la navegación

Las propiedades del robot relacionadas con tareas de navegación son las siguientes:

- **Configuración.** La configuración de un robot, denotada por q , es una completa especificación de la posición actual de dicho robot.
- **Espacio de configuración.** Corresponde al espacio C formado por todas las configuraciones que pueden ser alcanzadas por el robot R dentro del entorno de trabajo, y su dimensión es equivalente a los grados de libertad del robot. La dimensión del espacio de configuración es no Euclideano. Además, existe un subconjunto de C que determinan el espacio ocupado por el robot R , que esta denotado por $R(q)$ cuando el robot R se encuentra en q . Se define entonces a $R(q)$ por (1):

$$R(q) = \{q_i \in C \mid \|q, q_i\| \leq \rho\} \quad (1)$$
 Donde, ρ es el radio del robot R
- **Grados de libertad.** Los grados de libertad especifican el número de movimientos independientes que puede hacer el robot y determinan al mínimo número de variables que se necesita para representar a una configuración del robot.

B. Estándar Zigbee

Zigbee utiliza como base el estándar IEEE 802.15.4. Las características más relevantes que se encuentra en IEEE 802.15.4 son:

- Puede operar en tres bandas de frecuencias: 868 Mhz en Europa, 915 Mhz en los Estados Unidos y en la banda libre ISM¹ de 2.4 Ghz a nivel mundial.
- Tasa de transmisión de datos de 250 Kbps a 2.4 GHz, de 20 Kbps a 868 MHz y de 40 Kbps a 915 MHz.
- El acceso al medio lo hace por medio del algoritmo CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*).
- Está orientado hacia un bajo consumo de energía.

¹ ISM (Industrial, Scientific and Medical): son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones WLAN (e.g. Wi-Fi) o WPAN (e.g. Bluetooth).

- Emplea direcciones físicas de 64 bits IEEE y direcciones lógicas o de red de 16 bits.
- Proporciona un indicador de la calidad del enlace, LQI (*Link Quality Indicator*).
- Cuenta con un sistema que detecta la energía, ED (*Energy Detector*).
- La potencia de transmisión es de al menos 1 mW con un alcance mínimo de 10 m.

1) Tipos de dispositivos

En ZigBee se definen básicamente a 3 tipos de elementos:

- **Coordinador.** Es el nodo encargado de medir la energía en cada uno de los canales escogiendo el más silencioso y coloca un identificador de red de área personal (*Personal Area Network ID, PAN ID*) para establecer la red ZigBee. Una vez establecido estos parámetros, el coordinador puede formar una red, permitiendo unirse a él, routers y dispositivos finales. Ya formada la red, el coordinador hace las funciones de Router en el enrutamiento de paquetes y origen.
- **Router.** Es el encargado del enrutamiento de tramas hacia otros dispositivos de la red. Además, puede actuar como coordinador de un pequeño grupo de estos y capaz de almacenar los mensajes para ser enviados posteriormente.
- **Dispositivo final.** Los dispositivos finales tienen una funcionalidad reducida, debido a que no realizan tareas de enrutamiento. Por esta razón, la interacción con otros dispositivos de red se lo realiza a través del coordinador o de un router. Normalmente estos nodos tienen una configuración de bajo consumo de energía (Modo Sleep) para lograr una mayor duración de la batería al momento que interactúan dentro de la red.

2) Comparación con otras tecnologías

Resulta importante realizar una comparación entre Zigbee con los estándares Wi-fi y Bluetooth, que funcionan en la misma banda de frecuencia de 2.4 GHz y han alcanzado una gran acogida a nivel mundial.

La principal diferencia entre estas tecnologías radica en la cantidad de corriente que necesitan para operar, en donde, se evidencia un bajo consumo de energía por parte del estándar Zigbee. Esta característica permite que las baterías de los dispositivos en redes Zigbee duren un mayor periodo de tiempo.

Sin embargo, la tasa de transmisión de datos en Zigbee es la menor comparada con la que maneja Bluetooth y Wi-fi, lo que limita a esta tecnología a ser utilizada únicamente en redes LR-WPAN (*Low Rate – Wireless Personal Area Network*). A pesar del gran consumo de energía por parte del estándar Wi-fi hay que resaltar la capacidad de transmitir gran cantidad de datos.

IV. MATERIALES Y MÉTODOS

A. iRobot Create

El *iRobot Create* es una plataforma de desarrollo enfocada al ámbito educacional o aficionados por la Robótica. Este robot móvil, creado por la empresa iRobot, se programa en base a un conjunto de comandos, mediante los cuales, se pueden desarrollar nuevos comportamientos o inclusive emular a un instrumento musical. Además de ello, se puede añadir electrónica a su estructura, como sensores, indicadores, un brazo robótico, etc., utilizando sus conectores físicos. En la Fig. 1 se puede visualizar a esta plataforma y a algunas de sus partes.

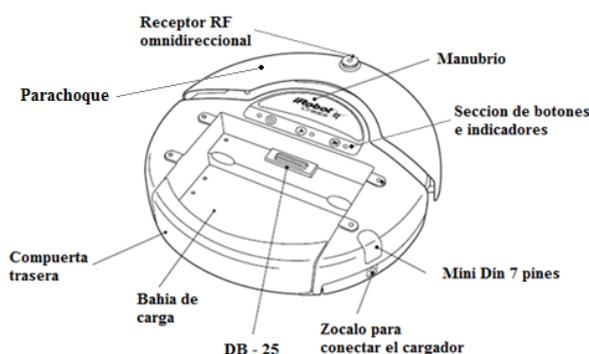


Fig. 1. Vista superior del *iRobot Create*.

El robot cuenta con un cable serial de programación, mediante el cual, se establece su control desde una PC.

B. Módulos RF XBee

Los módulos utilizados en este proyecto son:

- XBee (Series 2).
- XBee – PRO (Series 2).

Estos dispositivos son módulos de radio frecuencia diseñados para operar dentro estándar IEEE 802.15.4 en la banda 2.4GHz ISM. Los módulos proporcionan una comunicación inalámbrica de datos fiable y satisfacen las necesidades de bajo coste y reducido consumo de las aplicaciones inalámbricas. Además, son fáciles de usar y poner en marcha de manera inmediata, ya que poseen el stack correspondiente pre-cargado. En la Fig. 2. se visualizan a los módulos empleados.

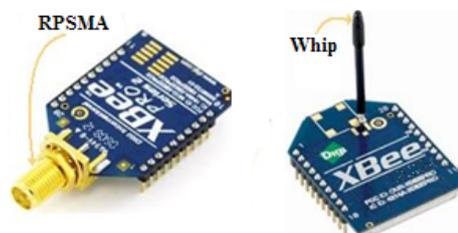


Fig. 2. Módulos XBee

Los módulos pueden ser configurados a través de comandos AT; y, operar de manera transparente ó *API* (*Application Programming Interface*). Cuando se los destina para operar de forma transparente actúan como una extensión

inalámbrica del puerto serial. Por otro lado, cuando su manera de operar es API, toda la información, tanto entrante como saliente está contenida en tramas que definen eventos u operaciones en el módulo.

C. Comunicación inalámbrica entre los módulos Xbee

Para montar la comunicación inalámbrica se configuró a los módulos de tal manera que operen de manera transparente. En la Tabla 1. se sintetiza la configuración de cada uno de los módulos

TABLA 1
Parámetros para la configuración de los módulos Xbee

Parámetro	XBee	Xbee - PRO
Velocidad de transmisión puerto serial	57600 Kpbs	57600 Kpbs
Bits de datos	8	8
Bits de paridad	0	0
Bits de parada	1	1
Direccionamiento	Físico	Físico
Firmware	Zigbee Router AT	Zigbee Coordinator AT

D. Algoritmos de navegación

Para seleccionar los algoritmos de navegación dentro de este proyecto se tomaron las siguientes consideraciones:

- El escenario que se planteó es de carácter desconocido, en donde, la información inicial del entorno con la que se le proveyó al robot fue únicamente su posición inicial y la ubicación de la meta. Los obstáculos en este caso se ubicaron de forma dinámica dentro del entorno.
- Los sensores que fueron considerados para que el *iRobot Create* se desempeñe en tareas de navegación fueron los de rotación y de impacto. El primer grupo de sensores se encarga de efectuar las mediciones en desplazamiento y en rotación del robot, mientras que los sensores del segundo grupo son utilizados para localizar a los obstáculos a lo largo del trayecto desde el origen hacia la meta. Los sensores de impacto del *iRobot Create* detectan a un obstáculo solo después de haber impactado contra éste. De lo anterior se desprende que el robot sustenta la implementación de algoritmos básicos para la navegación porque entre más complejo es el algoritmo mayor es la demanda en los requerimientos de los sensores.

Por lo anterior, se escogió a los algoritmos Gusano 1 y 2 para ser implementados en el *iRobot Create* porque requieren de un robot móvil con sensores de impacto para su funcionamiento y son empleados en ambientes desconocidos y además, le brindarán al robot la capacidad de alcanzar un objetivo mediante el planeamiento de trayectoria basado en la información que entreguen sus sensores.

Los algoritmos Gusano consideran al robot móvil como un punto con una perfecta localización y para detectar obstáculos utilizan sus sensores de impacto. Su funcionamiento está formado por dos comportamientos principalmente:

- Movimiento sobre una línea recta para alcanzar la meta; y,
- Seguimiento del contorno del obstáculo.

El éxito de estos algoritmos radica en usar los dos comportamientos anteriores para generar un camino libre de colisiones que permita alcanzar la meta. Existen tres clases de algoritmos Gusano que se basa en sensores de impacto, sin embargo, solo se toman en cuenta a dos de ellos porque son los que mayor difusión han tenido. A continuación se detallan a los algoritmos Gusano implementados.

1) Gusano 1

El robot parte del conocimiento la partida q_{inicio} , que constituye la primera configuración de salida; y, su llegada q_{meta} , que es la ubicación del objetivo que se desea alcanzar.

Las configuraciones de salida son etiquetadas como q_{i-1}^S , donde $i = 1, 2, 3 \dots$. Para el caso de la primera configuración de salida se tiene que $i = 1$, entonces $q_0^S = q_{inicio}$. Se establece la línea m , que une a q_{i-1}^S con la localización de la llegada q_{meta} , sobre la cual, el robot deberá desplazarse utilizando el comportamiento de moverse hacia un objetivo hasta alcanzar la meta o encontrar a un obstáculo. Si alguno de los sensores de contacto informa sobre un obstáculo, al primer punto de choque con el obstáculo se lo nombra como punto de impacto q_i^I . Entonces, el robot emplea el comportamiento de rodear obstáculos hasta alcanzar la meta o retornar a q_i^I . Después de ello, se determina el punto sobre el perímetro del obstáculo más cercano a la meta, conocido como punto de salida q_i^S . El robot debe dirigirse hasta q_i^S , incrementar en 1 el valor de i e invocar nuevamente al comportamiento de moverse hacia un objetivo. Para los casos en los cuales el robot llega a la meta se declara meta alcanzada. Sin embargo, si la línea que une a un punto de salida q_i^S con el punto de llegada q_{meta} se encuentra el obstáculo con el que se ha chocado anteriormente, el objetivo se declara inalcanzable. En la fig. 3 se muestra un ejemplo del desempeño del Algoritmo Gusano 1.

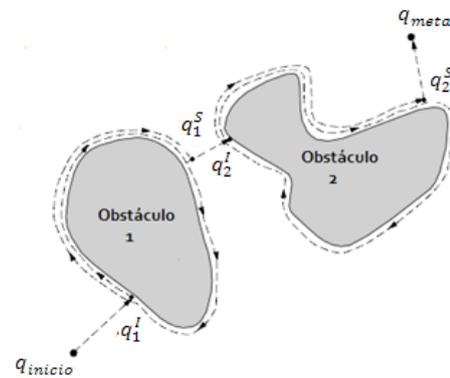


Fig. 3. Ejemplo del Algoritmo Gusano 1

2) Gusano 2

Durante el movimiento a la meta, el robot se mueve hacia el punto de llegada q_{meta} sobre una trayectoria rectilínea desde el punto de partida q_{inicio} , pero esta línea m que une a estos dos puntos permanece fija durante toda la ejecución del algoritmo Gusano 2. El comportamiento de seguimiento de borde es invocado si el robot encuentra un obstáculo, pero dicho comportamiento difiere del Gusano 1, porque el robot circunnavega el obstáculo hasta que encuentre un nuevo punto x que cumpla con la condición de ser el más cercano a la línea m . En este momento el robot nuevamente se llama al comportamiento de moverse hacia el objetivo, repitiendo este proceso si encuentra a otro obstáculo durante su trayecto. La Figura 4.3 indica un ejemplo del algoritmo Gusano 2.

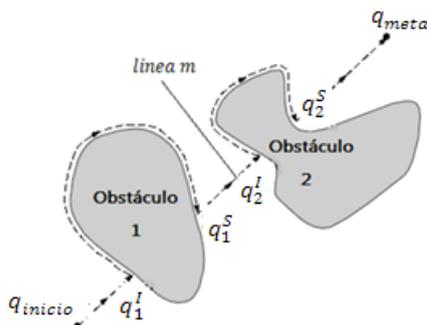


Fig. 4. Ejemplo del Algoritmo Gusano 2

V. IMPLEMENTACIÓN Y PRUEBAS

A. Comunicación Inalámbrica

Para la implementación de la comunicación inalámbrica entre los módulos Xbee se tomaron las siguientes consideraciones:

- **Requerimiento del proyecto.** Se necesita una comunicación inalámbrica fiable que brinde un PER (Packet Error Rate) del 0%. Con esto se garantiza que no existan paquetes perdidos en la recepción.
- **Carga de datos enviada.** Se establece que el periodo de envío de datos sea de 50ms y la máxima carga de datos a ser enviada corresponda a 30 bytes. Estos valores se establecen por la consulta del estado de los sensores al robot.

La Fig. 5. indica la arquitectura que se obtuvo después de integrar estas dos tecnologías.

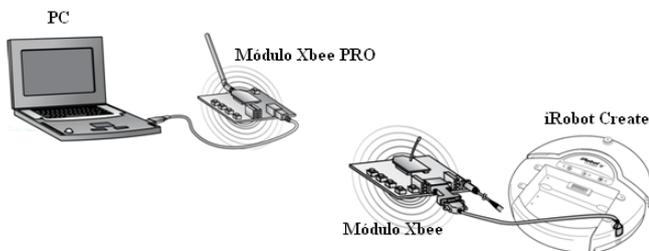


Fig. 1. Comunicación inalámbrica entre la PC y el iRobot Create.

B. Algoritmos Gusano

Para la implementación de los algoritmos Gusano, se requiere determinar las propiedades relacionadas a la tarea de navegación del iRobot Create. Éstas son descritas a continuación:

- **Grados de libertad.** La plataforma robótica móvil tiene tres grados de libertad definidos por sus capacidades: de traslación en línea recta o en arco a lo largo de un plano, y de rotación sobre su propio eje.
- **Configuración.** La información que contiene la configuración q se describe en (2) y se distribuye de la siguiente manera:
 - Dos variables x e y , que informan sobre la posición del robot en un plano.
 - Una variable θ , que detalla la orientación del robot.

$$q = (x, y, \theta) \quad (2)$$

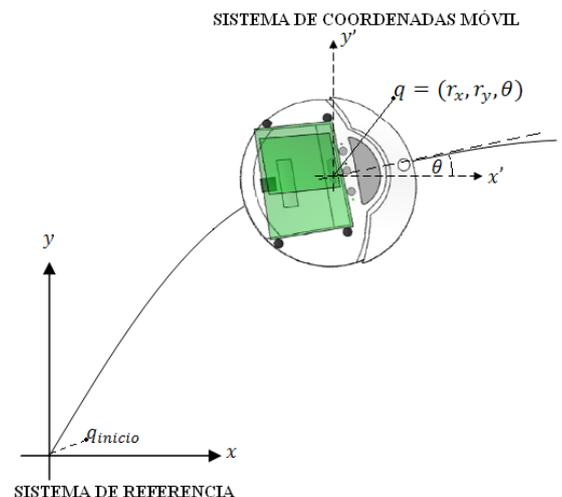


Fig. 6. Sistemas de coordenadas empleados para calcular las configuraciones del robot.

Para calcular las configuraciones alcanzadas por el robot se considera un sistema de referencia cuyo origen coincide con la ubicación de partida conocida como q_{inicio} , y un sistema de coordenadas móvil. Al robot se le considera como un punto, sobre el cual descansa el origen de dicho sistema de coordenadas móvil. La configuración calculada corresponde a la relación existente entre ambos sistemas de coordenadas, empleando vectores de posición y de desplazamiento para encontrar la posición del robot; y, Geometría Euclidiana para hallar la orientación. La Fig. 6. indica los sistemas de coordenadas utilizados.

- **Espacio de configuración.** El espacio de configuración corresponde a todas las configuraciones que puede alcanzar el robot dentro de un plano que tiene 2.0 m de ancho por 1.8 m de largo. En la Fig. 7 se muestra el espacio de trabajo definido.

Como el radio del robot es de 16.75 cm, el subconjunto C se define en (3).

$$R(q) = \{q_i \in C \parallel q, q_i \parallel \leq 16.75\} \quad (3)$$

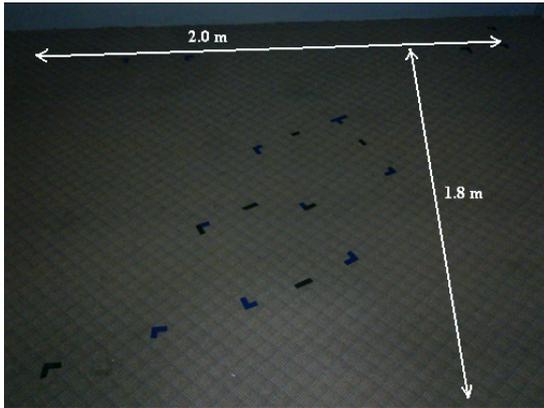


Fig. 7. Espacio de trabajo para la implementación de los algoritmos Gusano.

La implementación de los algoritmos se fundamenta básicamente en el desarrollo de dos rutinas que se describen a continuación:

- **Desplazamiento en dirección a la meta.** Esta rutina establece un vector que está definido desde la configuración de salida q_{i-1}^S hacia la configuración de llegada q_{meta} . Para que al robot se le permita moverse a lo largo de la línea de acción del vector, se calcula su dirección con respecto al sistema de coordenadas asociado a la configuración de salida y su módulo que es la distancia que separa a q_{i-1}^S de q_{meta} . Determinada la dirección, el robot rota sobre su propio eje, para después trasladarse de forma rectilínea el valor de la magnitud del vector. El desplazamiento hacia la meta termina cuando el robot ha alcanzado a q_{meta} , si un obstáculo ha sido encontrado durante el trayecto, ó si la meta es declarada inalcanzable.
- **Seguimiento del contorno del obstáculo.** La terminación de esta rutina depende del tipo de algoritmo Gusano. Es la encargada de indicarle al robot cómo bordear a un obstáculo después haberlo impactado para después planificar la trayectoria que le permita alcanzar la meta nuevamente. El bordeo de un objeto basa su funcionamiento en la información entregada por los sensores de impacto y se lo hace siempre en sentido horario. Para ejecutar esta rutina es necesario determinar la configuración del primer punto de impacto q_i^I , después de lo cual, el robot realiza lo siguiente:
 - a. **Retroceder en la misma dirección.** El robot recorre en sentido contrario una determinada distancia pero manteniendo la dirección con la cual impacto al obstáculo. Esto le permite al robot alejarse del obstáculo.
 - b. **Rotación sobre su propio eje.** Se le ordena al robot girar 45° en sentido anti horario con el propósito de no alcanzar un punto de impacto de manera consecutiva.
 - c. **Traslación en arco de circunferencia.** El robot procede a desplazarse sobre un arco de circunferencia que tiene un radio constante. Esta

acción está encaminada a alcanzar nuevamente al obstáculo definiendo un punto de choque diferente al anterior.

Para la programación de los algoritmos se optó por utilizar como herramienta de programación a MatLab que se caracteriza por manejar un lenguaje de alto nivel garantizando cálculos exactos en un corto periodo de tiempo y permitiendo la creación de una interfaz grafica.

VI. ESCENARIOS Y PRUEBAS

A. Comunicación inalámbrica

El escenario de pruebas que se consideró fue un entorno interno. Para esta prueba se tomó las siguientes consideraciones:

- **Área de trabajo de pruebas.** Para determinar el rendimiento de los módulos se define una separación máxima entre módulos de 30m dada por sus especificaciones en ambientes internos. En la Fig. 8. Se esquematizó anterior.

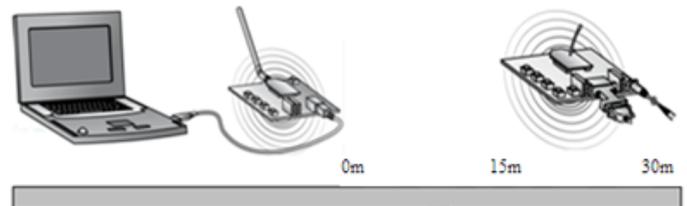


Fig. 8. Prueba de comunicación.

- **Carga de datos enviada.** Se estableció que el periodo de envío de datos sea de 50ms y la máxima carga de datos a ser enviada corresponda a 30 bytes. Estos valores se definieron por la consulta del estado de los sensores al robot.

B. Algoritmos de Gusano

Para efectuar las pruebas se han definido tres escenarios que están en función del tipo de obstáculos mostrados en la Fig.7., Fig. 8., y Fig.9. En cada escenario de pruebas se asume que el robot parte de posición (0 ; 0) m con respecto al sistema de referencia y se especifica las coordenadas de la meta que se desea alcanzar.

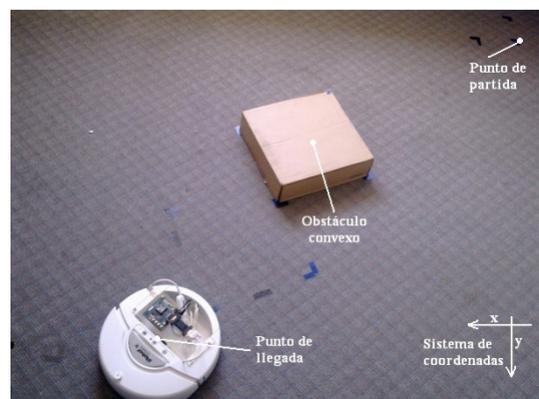


Fig 7. Escenario 1: contiene un a obstáculo convexo.

VII. RESULTADOS

A. Comunicación inalámbrica

A partir de una separación en distancia de 20m existieron pérdidas en la recepción de paquetes independiente de la carga de datos que fue transmitida. Entonces, se considera que el alcance máximo de la red, en la cual se garantiza que 100% de recepción de paquetes recibidos con una intensidad promedio de señal recibida de -64dBm, corresponde a un valor de 15m con una carga máxima de datos de 30 bytes.

El promedio de la intensidad de señal recibida se encuentra dentro del rango de -39dBm a -96dBm, parámetros que especifica el fabricante de las tarjetas.

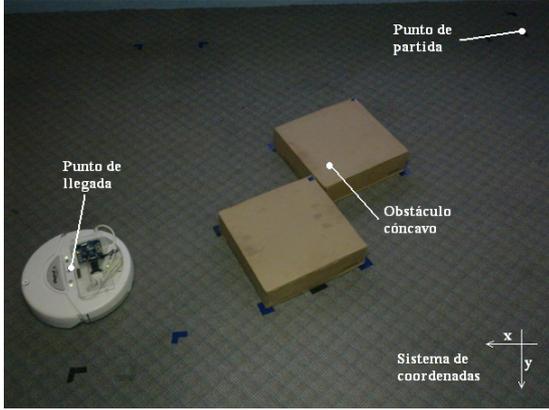


Fig. 8. Escenario 1: contiene un obstáculo cóncavo.

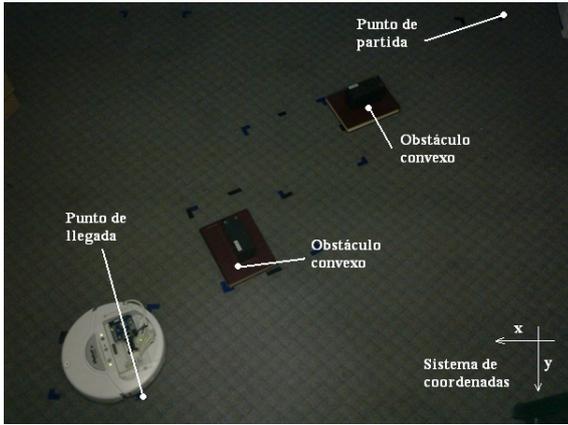


Fig. 9. Escenario 3: contiene a dos obstáculos convexos.

Además de detalló un criterio que sirvió para la recolección de los resultados obtenidos después del desempeño de los algoritmos Gusano. Para ello, se utilizó la definición del espacio ocupado por el robot para determinar un valor porcentual que indique la certeza con la cual el robot alcanza la meta en un determinado número de pruebas. Este proceso consistió en conocer cuan cercano ha quedado el robot de la meta utilizando (3). El momento que se considera para efectuar la medición es cuando la meta se declara alcanzada en el programa implementado sobre el robot. La medición a efectuarse es la distancia d existente entre el centro del robot q y q_{meta} . Existen tres posibilidades cuando esto ocurra:

a. **El robot ha alcanzado la meta.** Este caso corresponde al ideal y no es necesario realizar ninguna medición porque su valor porcentual se considera como 100%. Esta dado por (4):

$$d \leq \rho \quad (4)$$

b. **El robot se encuentra cercano a la meta.** Si pasaría esto el valor porcentual sería menor al 100% pero mayor al 0%. Este caso dependerá de que se cumpla (5):

$$\rho < d \leq 2\rho \quad (5)$$

c. **El robot se encuentra alejado de la meta.** A esta posibilidad se le considera la peor de todas. Se determina que el robot esta distante de la meta y su valor porcentual corresponde al de 0%. En este caso se da por (6):

$$d > 2\rho \quad (6)$$

En la Fig. 10. se ilustran los tres casos que pueden pasar.

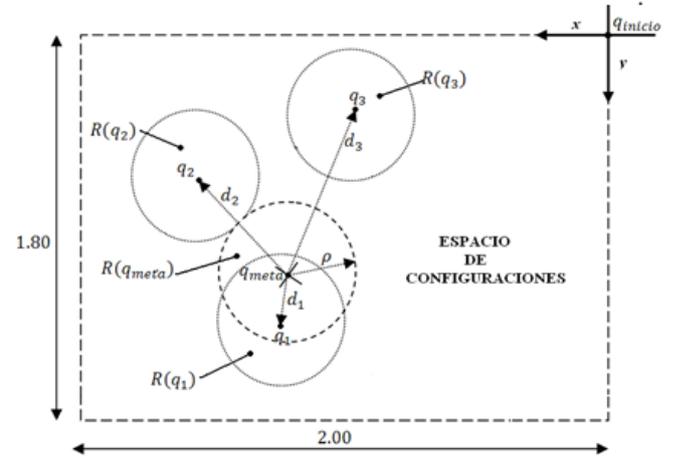


Fig. 10. Consideraciones para determinar el valor porcentual de alcanzar la meta, en donde, para $d_1 \leq \rho$ garantiza que el robot ha alcanzado la meta, para $\rho < d_2 \leq 2\rho$ entrega un valor porcentual indicando la cercanía del robot a la meta; y, para $d_3 > 2\rho$ contradice totalmente a la declaración de meta alcanzada porque el robot se encuentra alejado del punto de llegada.

B. Algoritmos Gusano

En la Figura. 5.8 se visualizan tres diagramas de barra que sintetizan la certeza de alcanzar la meta en cada algoritmo para los diferentes escenarios propuestos. En base a esta figura, se puede evidenciar que al utilizar el algoritmo Gusano 2 se obtuvo una mayor certeza de llegar a la meta para obstáculos convexos comparado con el Gusano 1.

El Gusano 1 sin importar el tipo de obstáculo mantiene una certeza parecida cuando sólo se presenta un obstáculo en su trayecto. Su caso crítico resulta cuando se enfrenta a más de un obstáculo, pues la seguridad de alcanzar la meta en este tipo de escenarios es muy baja como resultado del error de posición.

Otro aspecto a consideración es la distancia recorrida por el robot y el tiempo que demanda cada algoritmo para su ejecución en cada uno de los escenarios propuestos. En todos los escenarios de pruebas, el Gusano 2 le permite al robot recorrer una menor distancia hacia la meta cuando existen obstáculos en su trayecto comparado con lo que hace el robot bajo el Gusano 1. Además de ello, el tiempo que emplea el Gusano 2 para su culminación es inferior al que requiere el Gusano 1.

VIII. CONCLUSIONES

Al finalizar este trabajo, se demostró que los algoritmos implementados sobre la plataforma del *iRobot Create* le permitieron realizar la tarea de navegación en entornos no estructurados y contando únicamente con la información de su posición de origen o salida y la ubicación de la meta o punto de llegada.

Para los escenarios de pruebas propuestos, se comprobó que la certeza de llegar a la meta propuesta en el Gusano 1 depende del número de obstáculos que atraviesan su trayectoria hacia el punto de llegada. Bajo este algoritmo, el robot para más de un obstáculo presenta problemas en alcanzar la meta debido al error de posición que induce el bordeado completo de los obstáculos. Mientras que, la certeza conseguida con el Gusano 2 depende de cuan cercano es el punto determinado para salir del obstáculo con respecto la línea fija que une al punto de partida y llegada. Dentro del algoritmo Gusano 2, el número de obstáculos existentes entre el punto de inicio y de meta no afecta a la certeza de alcanzar la meta de manera considerable.

Bajo el análisis de resultado se demostró que los dispositivos XBee y XBee – PRO Series 2 actúan como una extensión wireless del puerto serie, es decir todos los datos enviados al puerto serie son transmitidos vía RF. Cuando los datos son recibidos por RF, estos son enviados al puerto serie.

IX. FUTUROS TRABAJOS

Para trabajos futuros se recomienda implementar un método que disminuya el error de posición.

También, dotar al robot de un mapa a más de la capacidad de desempeñarse en ambientes desconocidos, para que, sea capaz de ejecutar tareas de vigilancia de una determinada área, ó desempeñarse como guía de museo o bibliotecas.

Por último, añadir otro tipo de sensores a su electrónica que permita implementar algoritmos de navegación más sofisticados.

X. REFERENCIAS

- [1] S. M. LaValle, *Planning Algorithms*. Reino Unido, Universidad de Cambridge, 2006, también disponible en <http://msl.cs.uiuc.edu/planning/>.
- [2] The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard 802.15.4 - IEEE Standard for Information technology – Telecommunications and information exchange between systems - Local and Metropolitan area networks – Specific requirements*, October, 2003.
- [3] Lawrence H. Erickson, Joseph Knuth, Jason M. O’Kane, Steven M. LaValle, *Probabilistic localization with a blind robo*. In *Proceedings IEEE International Conference on Robotics and Automation*, 2008.
- [4] Ilari Vallivaara, Janne Haverinen, Anssi Kemppainen, Juha Rönning, *Simultaneous Localization and Mapping Using Ambient Magnetic Field*. 2010.
- [5] Max Kuipers *Localization with the iRobot Create, 2009*.
- [6] Roberto P. Baca, Sanny R. Checha, *Diseño y construcción de una red inalámbrica para cierre de caja de mesas de juego del Best Western Hotel Casino Plaza, utilizando tecnología ZigBee*, Febrero 2010.
- [7] O. Hyncica, P. Kacz, P. Fiedler, Z. Bradac, P. Kucera and R. Vrba, *The ZigBee Experience*.

Teleoperación Plataforma Móvil Irobot Create

W. Quishpe, A. Salas, R. Gordillo y C. D. Altamirano

Resumen—El tema de investigación está enfocado en la teleoperación de la plataforma móvil Irobot Create, esta plataforma móvil es distribuida por iRobot Corporation. La teleoperación consiste en enviar comandos a dicha plataforma para la ejecución de movimientos, esto mediante la utilización de un HMI basada en PC, así también la interface de teleoperación es capaz de mostrar imágenes del entorno de la plataforma móvil con ayuda de una cámara analógica montada sobre esta.

Palabra Clave: Teleoperación, Irobot, Wi-Fi

I. INTRODUCCIÓN

En la actualidad existen conjuntos robóticos destinados a realizar acciones y procesos específicos que ayudan en las actividades humanas; entre los existentes, se menciona con más detalle a los conjuntos robóticos móviles que, en la mayoría de casos, estos son tele-operados para realizar acciones desde locaciones remotas, y ayudan a:

- Evitar el contacto humano directo con el proceso.
- Realizar trabajos con difícil acceso para personas.

Entre las principales características que tienen estos conjuntos robóticos, son las distancias máximas de teleoperación, funciones que realizan y la autonomía del sistema.

Todas las especificaciones mencionadas, brindan la potencialidad necesaria a un conjunto robótico, dependiendo de la aplicación, las características se pueden mejorar individualmente o todas a la vez.

Para una teleoperación, además es necesaria una interface HMI, encargada de tener un control total sobre los accionamientos del conjunto robótico e información sobre el entorno del mismo con ayuda de sensores.

W. Quishpe es Ingeniero Electrónico en Automatización y Control por la Universidad de las Fuerzas Armadas ESPE (e-mail: william0016@hotmail.com).

A. Salas es Ingeniero Electrónico en Automatización y Control por la Universidad de las Fuerzas Armadas ESPE (e-mail: sulphur_17@hotmail.com).

R. Gordillo es Docente Tiempo Completo de la Carrera de Ingeniería Electrónica en Automatización y Control en la Universidad de las Fuerzas Armadas ESPE (e-mail: rxgordillo@espe.edu.ec).

C. D. Altamirano es Docente Tiempo Completo de la Carrera de Ingeniería Electrónica en Telecomunicaciones en la Universidad de las Fuerzas Armadas ESPE (e-mail: cdaltamirano@espe.edu.ec).

II. FUNDAMENTOS TEÓRICOS

A. Plataforma móvil Irobot Create

La base móvil Irobot Create mostrada en la Figura 1., es una plataforma de desarrollo, diseñada para ser acoplada con accesorios extras con la finalidad de proveer una mayor potencialidad a la misma. Irobot Create es capaz de maniobrar sobre superficies planas debido a su diseño, su capacidad de carga de 2.26 kg, permite montar sistemas adicionales sobre la plataforma móvil, como también movilizar objetos extras que no superen este peso máximo. Continuando con las características, Irobot Create dispone de un Puerto de propósito general, dicho puerto es utilizado para establecer comunicación con los actuadores y sensores de la base móvil, y tener un control total sobre esta; en adición se menciona que el puerto tiene una interface de entradas y salidas digitales disponibles para adicionar elementos electrónicos y brindar una mayor potencialidad.



Figura 1. Plataforma móvil Irobot Create

Al ser una plataforma de desarrollo, su programación se torna relativamente fácil para la implementación de diversos algoritmos de movimientos y su posterior aplicación.

Su diseño amigable y estratégico, permite montar con gran facilidad dispositivos y elementos extras que mejoren el desempeño y capacidad.

B. MiniCore RCM5600W

El módulo MiniCore RCM5600W Figura 2., es un microprocesador fabricado por Digi International; brinda recursos necesarios para la utilización en todo tipo de procesos convencionales e industriales. La velocidad, factor fundamental en un microprocesador, se hace presente en el RCM5600W, que opera a 74 MHz; soporta multitarea, característica que

ayuda en la velocidad, optimización y ejecución de los programas guardados en la memoria flash de 1 Mb.

Su interface de 52 pines, permite comunicarse hacia el exterior disponiendo de entradas y salidas digitales, puertos de comunicación serial, salidas pwm, y su característica principal, comunicación inalámbrica, que integra el estándar 802.11b/g.

Esta amplia gama de características hacen del MiniCore RCM5600W, un microcontrolador ideal para implementar en procesos que implementen cálculos matemáticos complejos, velocidades de respuesta rápida y procesos que necesiten una comunicación inalámbrica.



Figura 2. Minicore RCM5600W

C. Manipulador Robótico

Existen diferentes tipos de robots dependiendo de las acciones y movimientos que realicen, entre ellos están los robots antropomórficos ó manipuladores robóticos que simulan la extremidad superior del ser humano.

Una de las potencialidades extras que tiene provista la plataforma móvil Irobot Create, es la capacidad de manipular objetos que se encuentren en el entorno del conjunto robótico. El brazo implementado sobre la base móvil dispone de un grado de libertad que realiza un movimiento de elevación y descenso, además cuenta con un gripper o pinza en su extremo final, con la capacidad de sujetar objetos. Dichos elementos son accionados con la ayuda de servomotores para lograr movimientos suaves y minuciosos. El brazo manipulador se muestra en la Figura 3.



Figura 3. Manipulador robótico

D. Comunicaciones Inalámbricas

La flexibilidad dentro un proceso convencional o industrial, tiene como ventaja la posibilidad de ser monitoreado y operado, y el sistema se torna más eficiente cuando se dispone de un control inalámbrico. En la actualidad se puede encontrar en el mercado electrónico, numerosas opciones para implementar sistemas teleoperados.

Entre los protocolos inalámbricos más conocidos están: estándar Wi-Fi 802.11, bluetooth, Zigbee, WIMAX; siendo el primero el utilizado en la teleoperación de la plataforma móvil Irobot Create, es utilizado el estándar Wi-Fi 802.11 por sus características y desempeño en cuanto a distancias máximas de comunicación, velocidad de transmisión relativamente alta y la gran variedad de dispositivos que disponen de este estándar que pueden ser acoplados al conjunto robótico.

El conjunto robótico al disponer del estándar Wi-Fi 802.11, las Interfaces Hombre Máquina basadas en PC se enlazan directamente sin disponer de ningún dispositivo extra para su comunicación inalámbrica, excepto la tarjeta de red inalámbrica que por lo general vienen integradas en los ordenadores; con ello se elimina dispositivos extras para la teleoperación de sistemas que muchas veces entorpece el trabajo del operario o vuelve lento al sistema.

III. INTEGRACIÓN DE TECNOLOGÍAS

El conjunto robótico está implementado por la plataforma móvil Irobot Create encargado del desplazamiento en superficies planas, el MiniCore RCM5600W microcontrolador que sirve como interface entre la plataforma móvil y el HMI y el monitoreo y control remoto basado en PC ó en un teléfono inteligente que disponga de la aplicación TouchOSC.

A. Comunicación Irobot Create – RCM5600W

La comunicación establecida entre la base móvil y el microcontrolador, permite accionar movimientos al conjunto robótico y además tener información del entorno del conjunto para mostrar en el HMI. La comunicación entre estos dos dispositivos está basada en transmisión serial RS-232, Figura 4., el RCM5600W con ayuda de la comunicación serial acciona los movimientos manuales o automáticos de la base móvil y también es encargado de recibir información del entorno del conjunto robótico por medio de los sensores. Cabe mencionar que el accionamiento del brazo manipulador lo realiza directamente el módulo RCM5600W.



Figura 4. Comunicación Irobot Create – RCM5600W

B. Comunicación RCM5600W – HMI

La Interface Hombre Máquina que monitorea y acciona los movimientos del conjunto robótico está basada en PC.

El enlace entre el conjunto robótico y la interface HMI, es vía inalámbrica utilizando el protocolo UDP, permitiendo maniobrar el conjunto hacia lugares en que el hombre no puede tener un contacto directo con el proceso o simplemente locaciones inaccesibles para personas.

El conjunto robótico permite establecer la comunicación inalámbrica hacia el HMI de dos diferentes maneras:

1. Comunicación inalámbrica modo Adhoc.

En este modo el enlace entre el conjunto robótico y el HMI, se establece directamente Figura 5., entre ambos dispositivos, sin ayuda de elementos extras para el propósito, brindando mayor movilidad al sistema completo y reduciendo distancias de teleoperación.

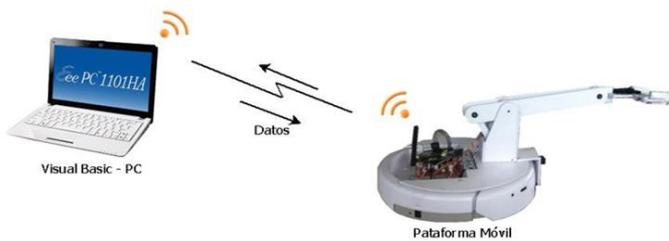


Figura 5. Comunicación inalámbrica modo Adhoc

2. Comunicación inalámbrica modo Infraestructura.

Este modo establece comunicación entre los dispositivos con ayuda de un dispositivo extra llamado router Wireless, Figura 6., el router crea un punto de acceso estableciendo una red entre todos los dispositivos que se unan a este. A diferencia del modo Adhoc, reduce la movilidad del sistema completo al depender de la conectividad con el router Wireless, y aumenta las distancias de teleoperación.



Figura 6. Comunicación inalámbrica modo Infraestructura

C. Interface Hombre Máquina

El HMI diseñado para monitoreo o control del conjunto robótico, está implementado en PC en el entorno de Visual Basic o en cualquier teléfono inteligente que disponga de la aplicación TouchOSC, ambas HMI disponen de funciones similares pero no en su totalidad.

El HMI basado en PC Figura 7., tiene como función la teleoperación del sistema robótico, permitiendo accionamientos manuales y recolección automática de objetos a corta distancia, además muestra información del entorno de la plataforma emitidos por sensores integrados, así también es capaz de adquirir imágenes desde una cámara montada sobre la base móvil utilizando la librería openCV, para tener información visual extra del entorno en el que se encuentra la plataforma Irobot Create.



Figura 7. HMI PC – Visual basic

El HMI integrado en un teléfono inteligente, Figura 8., estaba basada en la aplicación de pago TouchOSC disponible para IOS y android, esta permite únicamente la ejecución de los movimientos de la plataforma móvil y del brazo manipulador, aquí no se dispone de lectura de información del entorno, ni información visual emitida por la cámara.



Figura 8. HMI teléfono inteligente – TouchOSC

IV. RESULTADOS

Los resultados que se mencionan a continuación, muestran datos numéricos del desempeño y las capacidades experimentales del conjunto robótico.

- A. Alcance máximo de teleoperación modo Adhoc
 - a) Línea de vista: 60 m.
 - b) Cobertura lugares cerrados: 100 m² aprox.
- B. Alcance máximo de teleoperación modo Infraestructura
 - a) Línea de vista plataforma Irobot Create – router wireless: 210 m.
 - b) Línea de vista router wireless – HMI: 220 m.
 - c) Cobertura lugares cerrados: 700 m² aprox.
- C. Capacidad máxima soportado por el manipulador: 300 g.
- D. Repetibilidad recolección objetos a corta distancia
 - a) A 3 cm del gripper (mínima): 100%
 - b) A 6 cm: 100%
 - c) A 10 cm: 100%
 - d) A 15 cm: 80%
 - e) A 25 cm (máxima): 60%
- E. Autonomía: 120 min aprox.
- F. Velocidad plataforma Irobot Create
 - a) Modo avanzar: 0.83 – 24.11 cm/seg
 - b) Modo retroceder: 0.84 – 24.6 cm/seg
 - c) Giro derecha, izquierda: 4 – 21.25 rpm

V. CONCLUSIONES

Para el envío y recepción de datos entre el conjunto robótico y la interface HMI, es necesario de una comunicación inalámbrica en donde la transferencia de información sea en tiempo real, para ello se utiliza del protocolo UDP.

El MiniCore RCM5600W dispone de gran capacidad de procesamiento y memoria, permitiendo realizar todo tipo de proceso que demande altas complejidades de cálculo, como también procesos extensos donde es necesario memoria suficiente para el programa.

El uso del puerto de propósito general, presente en la plataforma Irobot Create, permite utilizar entradas digitales y

analógicas, y su puerto de comunicación, para la integración del manipulador robótico y el MiniCore RCM5600W.

En el diseño de la interface HMI, el uso de la librería de visión artificial, openCV, ayuda a tener un rendimiento óptimo en la adquisición de imágenes en tiempo real.

El MiniCore RCM5600W, refiriéndose en su comunicación inalámbrica, puede trabajar en modo Adhoc e Infraestructura, permitiendo en esta última mayores alcances de teleoperación.

El diseño eficiente del brazo manipulador, permite un menor esfuerzo de sus actuadores y por consiguiente un menor consumo de energía.

El algoritmo de recolección automática de objetos que realiza el conjunto robótico, es efectivo cuando el objeto se encuentra en línea directa, caso contrario, su eficacia de recolección es baja debido a que se tiene un único sensor de distancia encargado de realizar este proceso.

REFERENCIAS

- [1] Irobot Corporation, Irobot Create. *Irobot Create Open Interface*. [En línea]. <http://www.irobot.com>
- [2] MiniCore RCM5600W, User's Manual. *MiniCore RCM5600W C-Programmable Wi-Fi Core Module*. [En línea]. <http://www.rabbit.com>
- [3] Dynamic C, Function Reference Manual. *Integrated C Development System for Rabbit 4000, 5000, and 6000 Microprocessors*. [En línea]. <http://www.rabbit.com>
- [4] ARTAS, Engineering Software. *SAM – The Ultimate Mechanism Desigers*. [En línea] 2012. <http://www.artas.nl/>
- [5] Naranjo, Lilian. Diseño e implementación de un sistema de monitorización y control remoto mediante la tarjeta minicore RCM5600W. [En línea] 2011. <http://repositorio.espe.edu.ec/bitstream/21000/4535/1/T-ESPE-032706.pdf>
- [6] Álvarez, Luis y Figueroa, Juan. Implementación de algoritmos de navegación utilizando la plataforma del Irobot Create y módulos de comunicación inalámbrica XBEE. [En línea] 2011. <http://repositorio.espe.edu.ec/handle/21000/2643>
- [7] Bradsky, Gary. *Open Source Computer Vision*. [En línea] 2012. <http://opencv.willowgarage.com/wiki/>
- [8] National Instruments, forum. *Control LabVIEW from the iPhone, iPad or iPod Touch*. [En línea] 2010. <https://decibel.ni.com/content/docs/DOC-12289>.

Diseño y análisis de desempeño de un inversor de voltaje utilizando controladores inteligentes

Juan José Salazar Salvador

Resumen —El presente documento describe el desarrollo de controladores inteligentes, para la supresión de armónicos en inversores monofásicos con modulación senoidal por ancho de pulso (SPWM). En la primera fase se diseñó un controlador PI que sirve como modelo de referencia para la obtención de los patrones de entrenamiento de la red neuronal. Posteriormente se diseñó un controlador PD lógico difuso utilizando una base de 25 reglas. Las simulaciones realizadas con el modelo matemático del inversor y utilizando “SimPowerSystems” de MATLAB indican que es posible obtener una onda sinusoidal pura con un mínimo de distorsión armónica total (THD) mediante el uso de los controladores inteligentes propuestos.

Palabras Clave – Electrónica de Potencia, inversor de tensión, distorsión armónica, THD, red neuronal, control lógico difuso.

I. INTRODUCCIÓN

Este documento se enfoca en los dispositivos de potencia denominados inversores (convertidores DC/AC), los inversores son ampliamente usados para inyectar corriente a la red, proveniente de generación de energía eléctrica, además de sistemas de alimentación ininterrumpida UPS. Con el empleo de inversores, por ejemplo en granjas eólicas y fotovoltaicas, es posible inyectar la energía eléctrica permitiendo que operen en isla o conectados a la red.

Los inversores no son capaces de proporcionar un voltaje sinusoidal cuando operan bajo condiciones de carga no lineal y generan corrientes armónicas, por lo tanto se requiere controlar la forma de la señal a fin de que tenga la menor distorsión. Las soluciones para este problema se encuentran en plena fase de desarrollo, se sabe que un sistema de control podría corregir dicha distorsión de manera favorable, sin embargo las alternativas tradicionales no entregan resultados eficaces al trabajar con cargas no lineales. Existen diversas alternativas para controlar sistemas no lineales, entre ellas se encuentran: lógica difusa, algoritmos genéticos y redes neuronales.

En control los sistemas inteligentes ofrecen ventajas como: Aprendizaje adaptativo, tolerancia a fallos, operación en tiempo real y fácil inserción en computadores dedicados y controladores. Por estos motivos los controladores inteligentes son una estrategia de control ideal para modelos matemáticos no lineales como es el caso de determinados sistemas de potencia.

II. PRINCIPIO DE FUNCIONAMIENTO INVERSOR MONOFÁSICO PUENTE COMPLETO SPWM

El inversor monofásico propuesto consta de tres etapas que se describen mediante la Fig. 1:

- Generador PWM
- Etapa de potencia

- Filtro

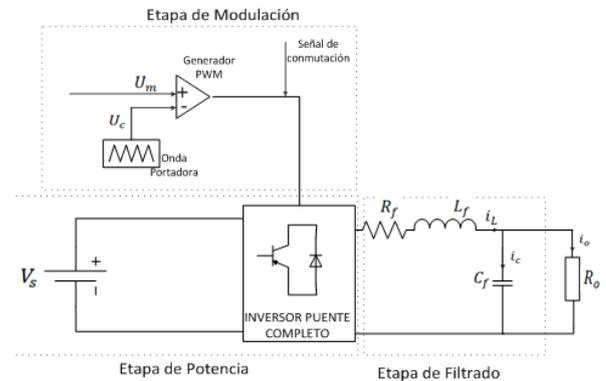


Fig. 1. Convertidor DC/AC Monofásico SPWM de puente completo

A. Generador PWM

El generador PWM es el encargado de entregar la señal de conmutación a los dispositivos semiconductores de potencia. Para este sistema se utilizó una modulación senoidal por ancho de pulso unipolar como se muestra en la Fig. 2. Este tipo de modulación consta de un generador de señal triangular (onda portadora) y un generador de señal de referencia sinusoidal (onda moduladora u onda de referencia). La tensión instantánea de la onda moduladora u_m es comparada con la tensión de la señal portadora u_c y de esta manera se obtiene la señal de conmutación.

El índice de modulación debe ser de un valor lo suficientemente alto para no tener distorsión armónica en frecuencias cercanas a la fundamental, mientras que no podrá tener un valor demasiado elevado para evitar pérdidas en la conmutación de los dispositivos de potencia y reducir el costo computacional, en este caso se utilizó un índice de modulación de 100.

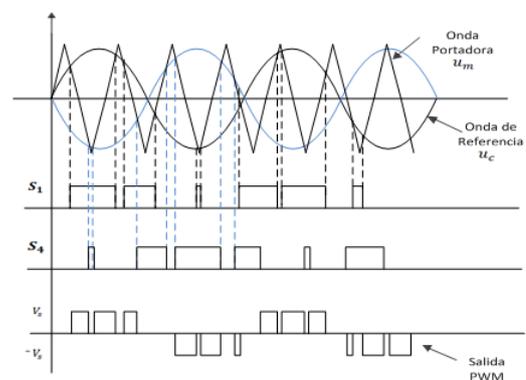


Fig. 2. Formas de onda modulación senoidal por ancho de pulso

La frecuencia fundamental de la onda moduladora debe tener un valor igual al de la salida deseada. En este caso la frecuencia deseada es igual a 60 Hz.

$$m_f = \frac{f_c}{f_m}$$

$$f_c = 60 * 100 = 6000\text{Hz}$$

Por lo tanto la frecuencia de la onda portadora triangular queda establecida en un valor igual a 6000Hz. Otro aspecto a considerar es el índice de modulación en amplitud el cuál debe tener un valor igual o inferior a 1 para asegurar la conmutación de los componentes electrónicos de potencia, la señal moduladora tendrá una amplitud estándar en sistemas de almacenamiento de energía en corriente continua, el cual corresponde a 12 Voltios.

$$m_a = \frac{U_m}{U_c} = 1$$

$$U_c = U_m = 12\text{V}$$

Para generar la onda triangular la pendiente se representa de la forma indicada en la siguiente ecuación:

$$m = \frac{4V}{T}$$

Donde m es la pendiente de la onda triangular y T representa el periodo de la onda el cual es igual a:

$$T = \frac{1}{60} = 0.0001667$$

Por lo tanto las ecuaciones que representan a la onda portadora se indican a continuación:

$$V = \begin{cases} m * t & ; \quad \left(0 < t \leq \frac{T}{4}\right) \\ U_c - m * t & ; \quad \left(\frac{T}{4} < t \leq \frac{3T}{4}\right) \\ -U_c + m * t & ; \quad \left(\frac{3T}{4} < t \leq T\right) \end{cases}$$

Donde U_c es la amplitud de la onda portadora y t representa el tiempo en segundos.

La onda moduladora tiene una forma sinusoidal y puede ser representada mediante la siguiente ecuación:

$$V(t) = 12 \cdot \text{sen}(2\pi \cdot 60 \cdot t)$$

B. Etapa de Potencia

La etapa de potencia consta de los dispositivos semiconductores de conmutación, en este caso se modeló un sistema de conversión DC/AC tipo puente completo como se indica en la Fig. 3, las señales de control de conmutación de los dispositivos son entregadas por el generador PWM. La tensión en cada uno de los brazos del inversor de tensión se representa de las siguientes ecuaciones:

$$V_a = \begin{cases} V_s & (u_m \geq u_c) \\ 0 & (u_m < u_c) \end{cases}$$

$$V_b = \begin{cases} V_s & (-u_m \leq u_c) \\ 0 & (-u_m > u_c) \end{cases}$$

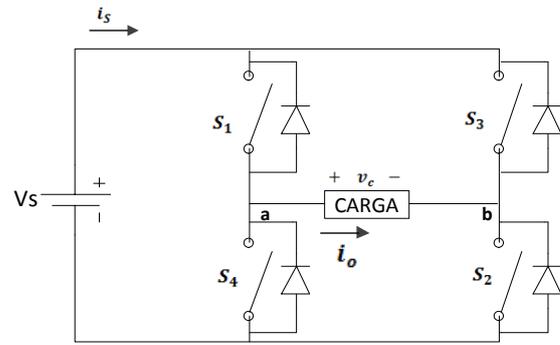


Fig. 3. Etapa de potencia inversor monofásico puente completo

La ecuación que resume el comportamiento del puente completo se presenta a continuación:

$$V_c = (V_a - V_b)$$

C. Filtro

Para reducir la distorsión armónica total es necesario utilizar filtros y de este modo obtener una forma de onda sinusoidal pura. Para este efecto se diseña un filtro LC que elimina las componentes armónicas en frecuencias superiores a la fundamental (60Hz). Es decir que se diseñó un filtro pasa bajos.

Los valores de los componentes resultantes del diseño del filtro son:

$$L_f = 200\mu\text{H}$$

$$C_f = 50\mu\text{F}$$

$$R_f = 0.02\Omega$$

El diagrama del inversor monofásico SPWM modelado en SIMULINK se muestra en la Fig. 4. En la Tabla I se muestran los parámetros del inversor.

TABLA I. PARÁMETROS DEL INVERSOR

Parámetro	Valor	Unidad
Frecuencia de Conmutación, fs	6	KHz
Tensión nominal de salida	48	V_{pico}
Frecuencia nominal de salida	60	Hz
Inductancia del filtro, Lf	200	μH
Resistencia del inductor, Rf	0.02	Ω
Capacitor del filtro, Cf	50	μF

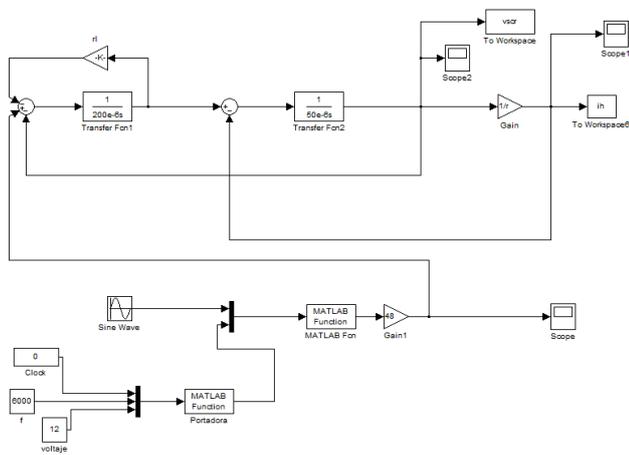


Fig. 4. Modelo inversor SPWM con Filtrado

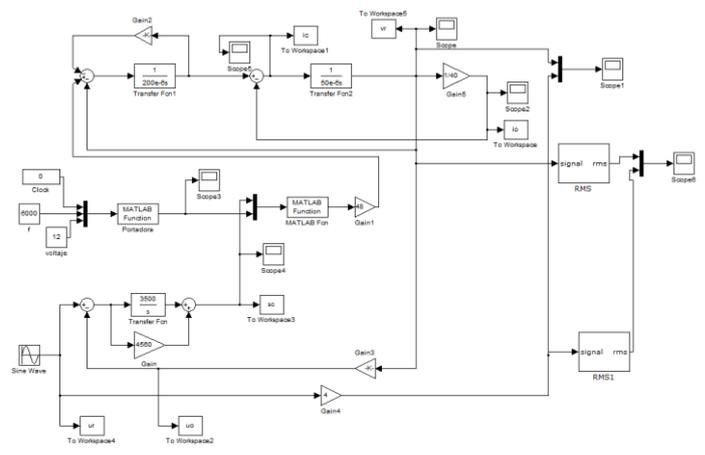


Fig. 6. Diagrama de bloques SIMULINK del controlador PI del inversor PWM

III. DISEÑO DE CONTROLADORES

A. Controlador PI

Para obtener valores de referencia para la construcción de los controladores inteligentes se diseñó un modelo de referencia utilizando controladores clásicos. Por lo tanto se diseñó el controlador PI representado por el diagrama de bloques de la Fig. 55.

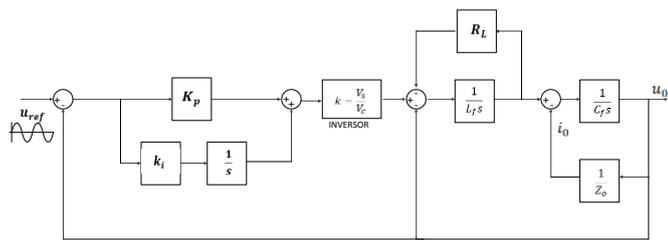


Fig. 5. Diagrama De Bloques Inversor con controlador PI

El controlador PI actúa sobre la onda de referencia u onda moduladora modificando el índice de modulación para compensar las corrientes armónicas que trabajan en los múltiplos de la frecuencia portadora de 6KHz. Los valores de las ganancias del controlador se muestran en la Tabla II.

TABLA II VALORES CONTROLADOR CLÁSICO

Ganancia	Valor
k_p	4560
k_i	3500

En la Fig. 6 se muestra el diagrama de bloques del inversor como modelo matemático utilizando un control de tipo PI.

B. Controlador Redes Neuronales

El controlador PI se utilizó como modelo de referencia para el entrenamiento de la red neuronal, es necesario definir las entradas primordiales que permitan un correcto desempeño del controlador basado en RNA, para poder controlar el inversor SPWM. Las entradas utilizadas son las siguientes:

- Corriente de capacitor i_c
- Corriente de carga i_o
- Tensión de salida u_o
- Error entre tensión de salida y tensión de referencia e_o

Mientras que la salida de la red neuronal es la señal de compensación que modificará el índice de modulación en frecuencia m_f y en amplitud m_a , y por lo tanto elimina las componentes armónicas.

La estructura de la red neuronal tendrá la forma que se muestra en el diagrama de la Fig. 7. Para la elección de la arquitectura de la red neuronal se debe considerar que el número de neuronas en la capa oculta debe ser bajo para reducir el costo computacional, pero no se puede sacrificar la eficiencia de la red, por lo cual se propuso una red con una capa de cuatro entradas, una capa oculta con 5 neuronas y una sola neurona de salida.

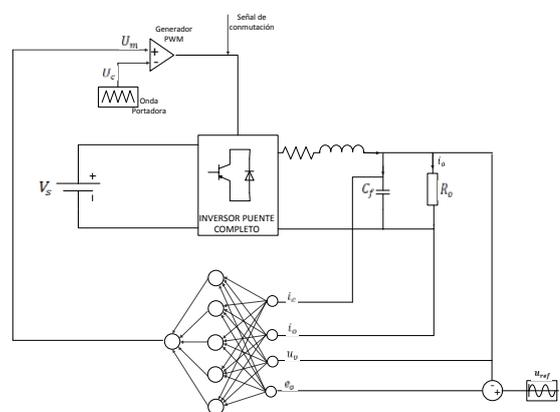


Fig. 7. Diagrama Inversor SPWM controlado mediante RNA

Para el entrenamiento fuera de línea de las RNA se necesitó de una amplia base de datos que contenga las relaciones de entrada y salida. En el caso del inversor propuesto, la base de datos debe incluir los patrones de entrada y salida bajo todas las condiciones de carga posibles. Para la obtención de los patrones se simula el modelo del controlador PI.

La red neuronal que actúa de forma similar al controlador PI se entrena utilizando la herramienta de MATLAB (Neural Network Training)

A continuación se presenta un resumen de los pasos a seguir para el diseño del controlador con redes neuronales artificiales para aplicaciones de inversores.

- 1) Construir el controlador PI simulado como se muestra en la Figura 5.
- 2) Para cada condición de carga, recolectar la salida de tensión, corriente de carga, y corriente de capacitor como entradas de la RNA, y la señal de compensación como la salida deseada de la misma. Estos patrones forman una base de datos para el entrenamiento de la red neuronal artificial
- 3) Seleccionar una estructura de red neuronal que sea simple y además suficiente para el control adecuado del inversor
- 4) Entrenar la red neuronal utilizando herramientas de software (MATLAB con “Neural Network Toolbox”)
- 5) Simular el modelo utilizando el bloque de la red neuronal generado. Como se muestra en la Fig. 8.

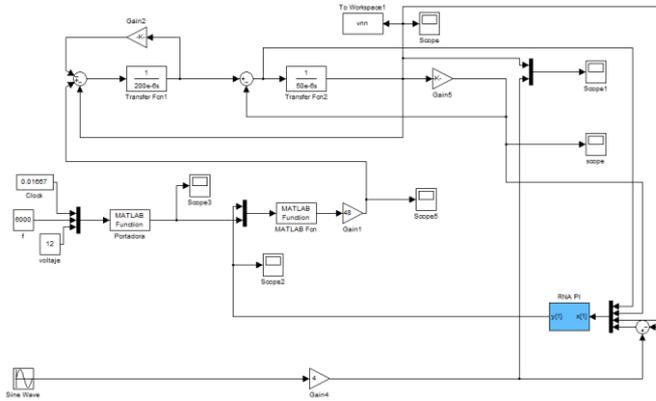


Fig. 8. Modelo matemático SIMULINK del inversor controlado con RNA

C. Controlador Difuso

Para el diseño del controlador difuso se utilizó la herramienta “FIS Editor” de MATLAB donde se diseña el sistema difuso de inferencia considerando las entradas:

- Error (Control Proporcional)
- Cambio de error (Control Derivativo)

Mientras que la salida es la señal compensada que hace las funciones de onda moduladora o señal de referencia. Para conseguir los rangos en que el error y el cambio de error se presentan, se obtuvieron los valores con el sistema sin controlador. El valor de salida del sistema difuso se obtuvo de acuerdo a la variación en la tensión de salida del controlador PI diseñado anteriormente. Al realizar estas acciones se consiguieron los siguientes resultados:

TABLA III. RANGOS DE ERROR Y CAMBIO DE ERROR

	Error	Δ error	Salida
Mínimo	-0.5097	-5825	-200
Máximo	0.5710	5963	200

La base de reglas se resume en la tabla mostrada a continuación:

TABLA IV. REGLAS DEL SISTEMA DIFUSO

		Error				
		GN	PN	Z	PP	GP
Δ error	GN	GN	GN	GN	PN	Z
	PN	GN	GN	PN	Z	PP
	Z	GN	GN	Z	PP	GP
	PP	PN	Z	PP	GP	GP
	GP	Z	PP	GP	GP	GP

En la Fig. 9 se observa el modelo matemático del inversor con el controlador lógico difuso, como se puede ver no es necesario añadir ganancias proporcionales ni derivativas, ni tampoco ganancia a la salida del sistema difuso, ya que el controlador funciona correctamente con los rangos establecidos.

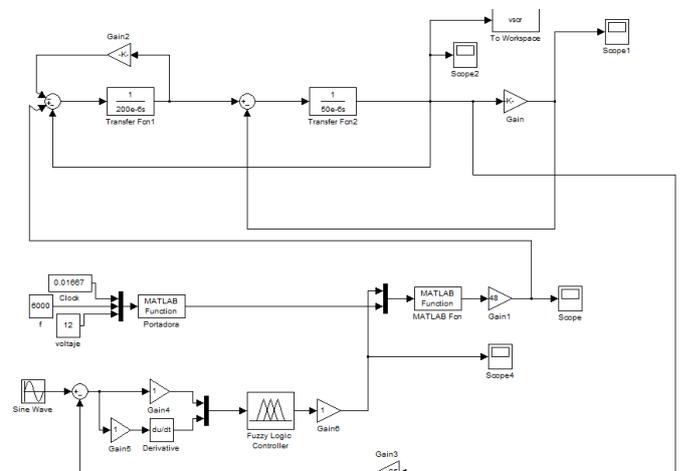


Fig. 9. Inversor modelo Matemático con controlador Difuso

IV. MODELAMIENTO MEDIANTE “SIMPOWERSYSTEMS”

Para el modelamiento del sistema en “SimPowerSystems” se utilizó los datos de un transistor MOSFET de potencia con las características principales enunciadas en la siguiente tabla:

TABLA V. CARACTERÍSTICAS DEL ELEMENTO CONMUTADOR DE POTENCIA

$I_D(A)$	50
$V_{DS}(V)$	50
$P_D(W)$	200
$R_{DS}(\Omega)$	0.028

Para la simulación del puente que actuó como etapa de potencia se utilizó el bloque “Universal Bridge”, en el cual se configura un puente de dos brazos tipo MOSFET con las características anteriormente señaladas en la Tabla V.

El filtro fue diseñado utilizando los bloques RLC de Simulink con los mismos valores indicados en el modelamiento matemático. Mientras que la etapa de comparación de la onda moduladora y la onda portadora se la realizó mediante el bloque “PWM Generator” el mismo que trabaja como amplificadores operacionales que permiten la comparación de ondas, y a la salida se obtienen cuatro señales de conmutación para los cuatro elementos de potencia del puente inversor. En la Fig. 10 se muestra el sistema de potencia completo del inversor SPWM modelado en “SimPowerSystems”.

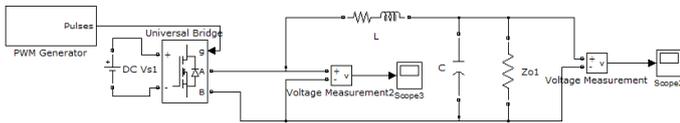


Fig. 10. Inversor de potencia SPWM modelado en “SimPowerSystems”

En las Fig. 11, 12 y 13 se muestran los modelos del inversor con controlador PI, RNA y difuso respectivamente.

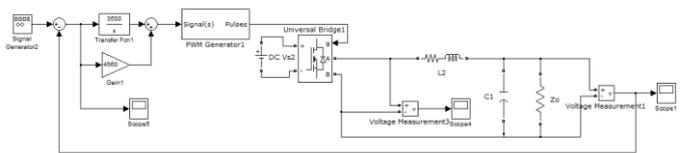


Fig. 11. Modelo de transistor del inversor SPWM con controlador PI

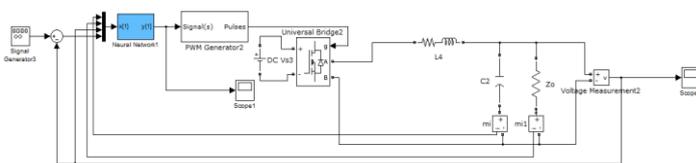


Fig. 12. Modelo de transistor del inversor SPWM con RNA.

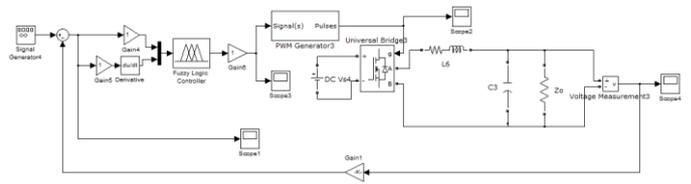


Fig. 13. Modelo de transistor del inversor SPWM con controlador difuso

V. RESULTADOS

Para medir la eficiencia del controlador se obtuvo la THD con cada uno de los controladores diseñados y con varios controladores de carga.

A. Resultados Carga lineal modelo matemático

Utilizando una carga resistiva de 40Ω combinada con varias cargas inductivas y capacitivas se obtuvieron los resultados de THD mostrados en la Tabla VI. En las Fig. 14, 15, 16 y 17 se muestran las formas de onda para carga de 40Ω , sin controlador, con control PI, RNA y difuso.

TABLA VI DISTORSIÓN ARMÓNICA CARGA LINEAL MODELO MATEMÁTICO

Tipo de Carga	Impedancia	THD% Sin Control	THD %PI	THD % RNA	THD % FLC
R	40Ω	1.59	0.11	0.08	0.08
RL	$40\Omega-0.1mH$	1.58	0.28	0.07	0.07
	$40\Omega-1mH$	1.42	0.25	0.10	0.09
	$40\Omega-10mH$	1.30	1.13	1.08	1.08
RC	$40\Omega-10\mu F$	1.43	0.21	0.10	0.09
	$40\Omega-100\mu F$	0.96	0.33	0.11	0.10
	$40\Omega-1000\mu F$	1.01	0.26	0.10	0.09

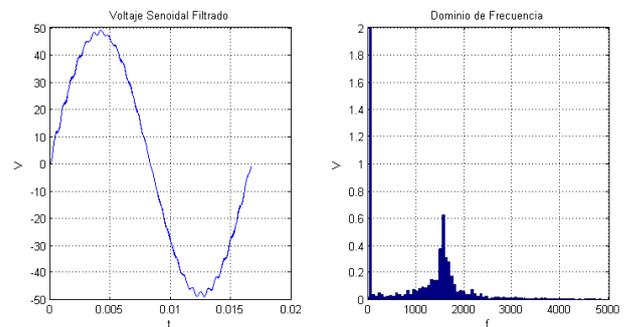


Fig. 14 Forma de onda Tensión de salida sin controlador modelo matemático

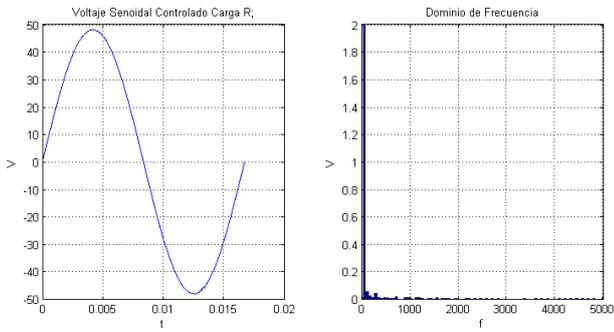


Fig. 15. Forma de onda Tensión de salida controlador PI modelo matemático

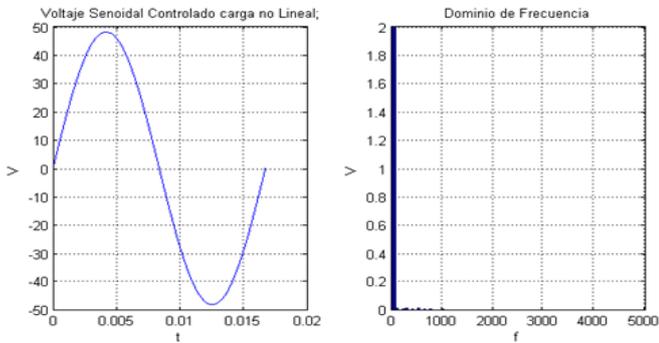


Fig. 16. Forma de onda Tensión de salida controlador RNA modelo matemático

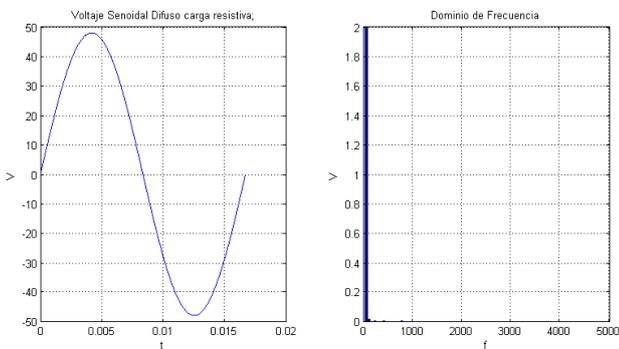


Fig. 17. Forma de onda Tensión de salida controlador difuso modelo matemático

B. Resultados carga no lineal modelo matemático

Para medir la distorsión armónica bajo condiciones de carga no lineal se utilizó un puente rectificador conectado a diferentes cargas R, RL y RC. En la Tabla VII se muestra el resumen de THD analizado con varios valores de carga. En las Fig. 18, 19, 20 y 21 se muestran las formas de onda para carga de no lineales con impedancia de 40Ω , sin controlador, con control PI, RNA y difuso.

TABLA VII. DISTORSIÓN ARMÓNICA CARGA NO LINEAL MODELO MATEMÁTICO

Tipo de Carga	Impedancia	THD% Sin Control	THD %PI	THD % RNA	THD % FLC
R	40Ω	4.38	0.24	0.06	0.09
RL	40Ω -0.1mH	4.37	0.17	0.08	0.07
	40Ω -1mH	4.37	0.19	0.06	0.08
	40Ω -10mH	4.51	0.17	0.06	0.08
RC	40Ω -10uF	3.87	0.07	0.05	0.07
	40Ω -100uF	4.77	0.09	0.07	0.09
	40Ω -1000uF	4.85	0.10	0.07	0.08

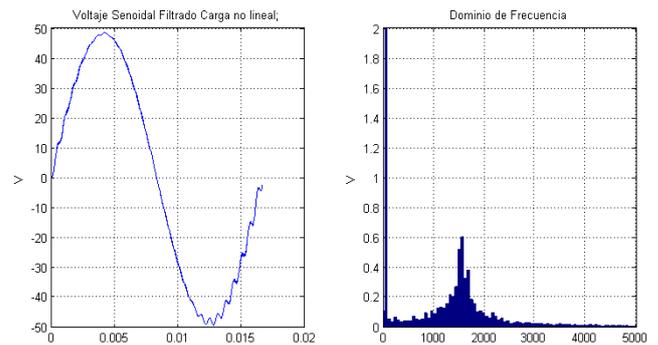


Fig. 18. Forma de onda Tensión de salida sin controlador carga no lineal modelo matemático

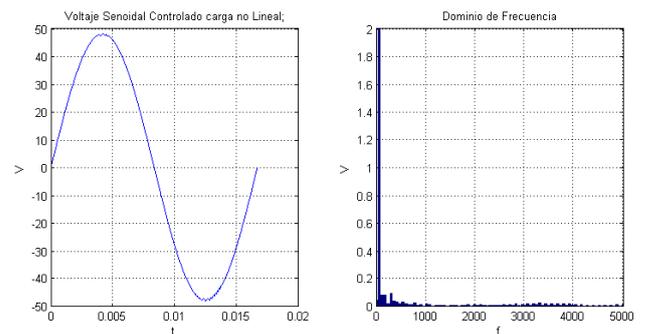


Fig. 19. Forma de onda Tensión de salida controlador PI carga no lineal modelo matemático

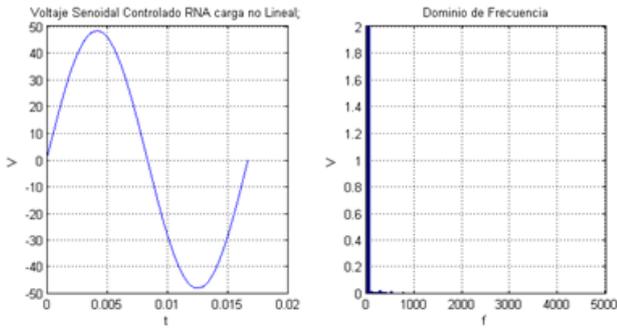


Fig. 20. Forma de onda Tensión de salida controlador RNA carga no lineal modelo matemático

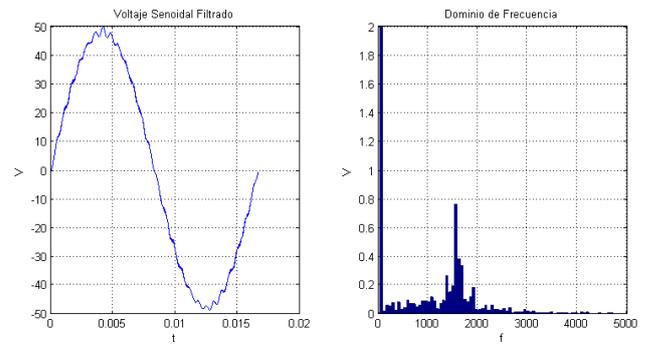


Fig. 22. Forma de onda Tensión de salida sin controlador carga lineal modelo "SimPowerSystems"

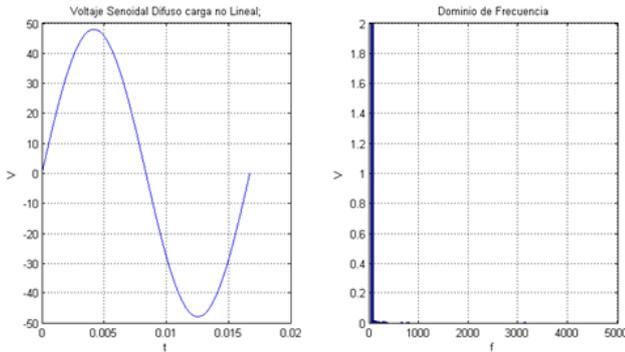


Fig. 21. Forma de onda Tensión de salida controlador RNA carga no lineal modelo matemático

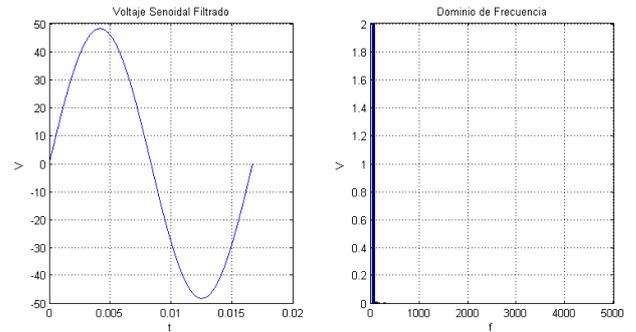


Fig. 23. Forma de onda Tensión de salida controlador PI carga lineal modelo "SimPowerSystems"

C. Resultados carga lineal modelo "SimPowerSystems"

Con el modelo de "SimPowerSystems" con carga lineal se realizaron las pruebas utilizando una carga resistiva de 40Ω combinada con varias cargas inductivas y capacitivas se obtuvieron los resultados de THD mostrados en la Tabla VIII. En las Fig. 22, 23, 24 y 25 se muestran las formas de onda para carga de 40Ω , sin controlador, con control PI, RNA y difuso.

TABLA VIII. DISTORSIÓN ARMÓNICA CARGA LINEAL MODELO "SIMPOWERSYSTEMS"

Tipo de Carga	Impedancia	THD% Sin Control	THD % PI	THD % RNA	THD % FLC
R	40Ω	1.27	0.06	0.13	0.05
RL	$40\Omega-0.1mH$	1.27	0.06	0.07	0.05
	$40\Omega-1mH$	1.28	0.07	0.29	0.05
	$40\Omega-10mH$	1.61	0.07	0.08	0.05
RC	$40\Omega-10\mu F$	1.30	0.06	0.06	0.07
	$40\Omega-100\mu F$	1.27	0.06	0.07	0.07
	$40\Omega-1000\mu F$	1.27	0.06	0.07	0.07

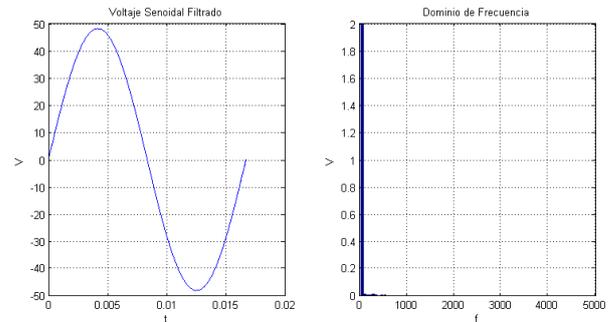


Fig. 24. Forma de onda Tensión de salida controlador RNA carga lineal modelo "SimPowerSystems"

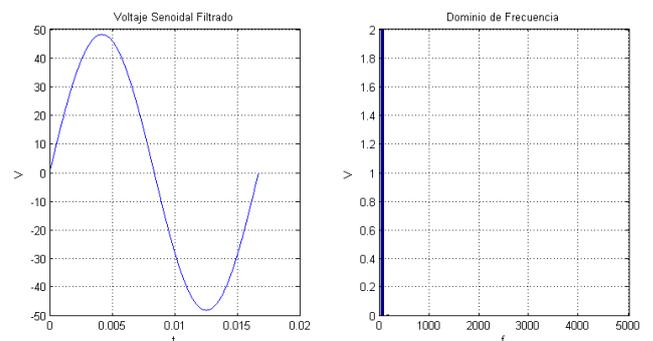


Fig. 25. Forma de onda Tensión de salida controlador difuso carga lineal modelo "SimPowerSystems"

D. Resultados carga no lineal modelo “SimPowerSystems”

Con el modelo de “SimPowerSystems” con no carga lineal, es decir un puente rectificador, se realizaron las pruebas utilizando una carga resistiva de 40Ω combinada con varias cargas inductivas y capacitivas se obtuvieron los resultados de THD mostrados en la Tabla VIII. En las Fig. 26, 27, 28 y 29 se muestran las formas de onda para carga de 40 Ω, sin controlador, con control PI, RNA y difuso.

Tabla IX. DISTORSIÓN ARMÓNICA CARGA NO LINEAL MODELO “SIMPOWERSYSTEMS”

Tipo de Carga	Impedancia	THD% Sin Control	THD % PI	THD % RNA	THD % FLC
R	40 Ω	24.22	2.04	1.11	0.30
RL	40 Ω-0.1mH	24.22	1.95	1.14	0.30
	40 Ω-1mH	24.22	1.94	1.14	0.30
	40 Ω-10mH	24.23	1.65	1.15	0.30
RC	40 Ω-10uF	24.50	1.78	1.04	0.06
	40 Ω-100uF	24.35	1.74	1.02	0.27
	40 Ω-1000uF	24.24	1.69	1.05	0.30

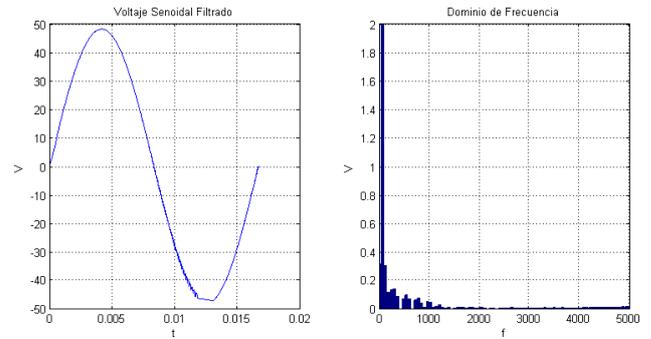


Fig. 28. Forma de onda Tensión de salida controlador RNA carga no lineal modelo “SimPowerSystems”

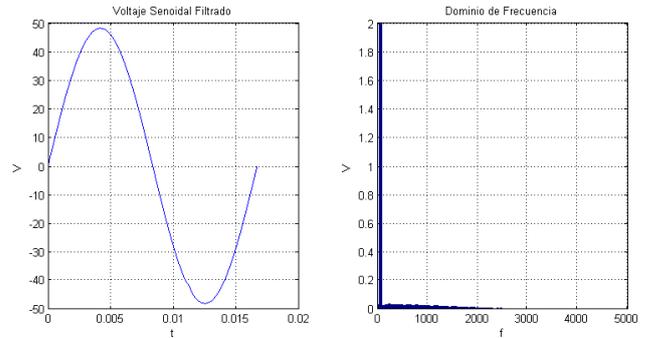


Fig. 29. Forma de onda Tensión de salida controlador Difuso carga no lineal modelo “SimPowerSystems”

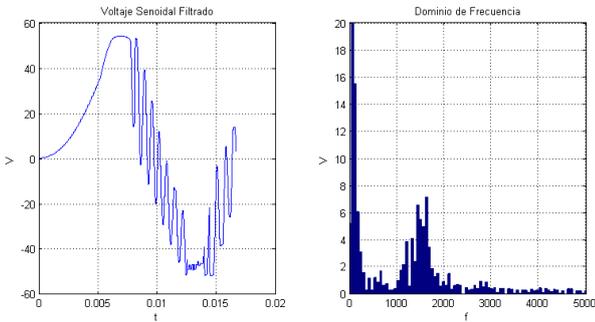


Fig. 26. Forma de onda Tensión de salida sin controlador carga no lineal modelo “SimPowerSystems”

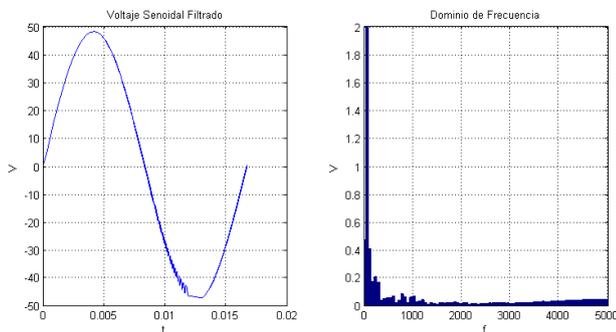


Fig. 27. Forma de onda Tensión de salida controlador PI carga no lineal modelo “SimPowerSystems”

VI. CONCLUSIÓN

En el presente documento se presentó alternativas de control inteligente para obtener una tensión de salida con baja distorsión armónica total para un inversor monofásico con modulación senoidal por ancho de pulso (SPWM). Los controladores propuestos se basan en redes neuronales artificiales y control difuso. Los controladores fueron diseñados utilizando el modelo matemático del inversor, además se comprobó su funcionamiento utilizando el modelo de transistor mosfet utilizando “SimPowerSystems” de MATLAB. El comportamiento de cada uno de los controladores fue medido utilizando cargas lineales y un puente rectificador que trabajó como carga no lineal.

Los controladores inteligentes mostraron un mejor desempeño que el controlador clásico en líneas generales, especialmente al actuar bajo condiciones de carga no lineal, el controlador neuronal es capaz de trabajar con cargas para las cuáles no fue entrenado además de reducir la THD, mientras que el control difuso disminuye a casi cero el valor de distorsión armónica total.

El uso de controladores inteligentes para la regulación de armónicos nos permite tener un equipo altamente robusto y que prevenga los efectos instantáneos y a largo plazo, que generan la distorsión armónica.

Los controladores inteligentes al suprimir la mayoría de corrientes armónicas son capaces de prevenir falsas conmutaciones de dispositivos de potencia, además son

capaces de eliminar fenómenos como vibración y calentamiento de equipos principalmente electromagnéticos, además de evitar inconvenientes con los sistemas de protección.

La eficiencia mostrada por los controladores inteligentes podría permitir reducir el tamaño de los sistemas de potencia al eliminar pérdidas por calor, además de ser sistemas ideales ante el uso de cargas no lineales tales como fuentes conmutadas. Con el avance de los procesadores modernos los controladores inteligentes pueden ser implementados de manera económica por lo cual presentan una alternativa muy eficiente para el control de sistemas de potencia.

REFERENCIAS

- [1] RASHID, Muhammad, *Electrónica de Potencia Circuitos Dispositivos y aplicaciones*, Tercera edición, Prentice Hall, México, 2004.
- [2] HART, Daniel, *Electrónica de Potencia*, Primera Edición, Prentice Hall, Madrid, 2001.
- [3] [http://tec.upc.es/el/TEMA-5%20EP%20\(v1\).pdf](http://tec.upc.es/el/TEMA-5%20EP%20(v1).pdf), Convertidores u onduladores.
- [4] http://catarina.udlap.mx/u_dl_a/tales/documentos/meie/fernandez_n_jm/ca_pitulo1.pdf, Introducción a los controladores CD-AC.
- [5] MOHAN, UNDELAND, ROBBINS, Ned, Tore, William, *Power Electronics Converters applications and design*, Segunda Edición, John Wiley, Nueva York, 2002.
- [6] DÍAZ, Juan, *Inversores PWM*, Oviedo-España, 1998.
- [7] BHOOPAL, MADHAV, *Neural and Neuro-Fuzzy Controllers for UPS Inverter Applications*, *International Journal of Recent Trends in Engineering*, Volumen 2, Nro: 8, Noviembre 2009.
- [8] SHAHGHOULIAN, FAIZ, JABBARI, Ghazanfar, Jawad, Masoud, *Voltage Control Techniques in Uninterruptible Power Supply Inverters* *International Review of Electrical Engineering (I.R.E.E.)*, Vol. 6, N. 4, Julio-Agosto 2011.
- [9] XIAO SUN, Martin, *Analogue Implementation of a Neural Network, Controller for UPS Inverter Applications*, *IEEE transactions on power electronics*, vol. 17, no. 3, mayo 2002.
- [10] DJAFFAR, WIRA, MERCKLE, Abdeslam, Patrice, Jean, *Artificial Neural Networks to Control an Inverter in a Harmonic Distortion Compensation Scheme*, *International Symposium on Industrial Electronics (ISIE 2008)*, Cambridge, Inglaterra, 2008.
- [11] HUDSON, HAGAN ,DEMUTH, Mark, Martin, Howard *Neural Network Toolbox User's Guide*, The MathWorks Inc, 2010.

Diseño e implementación del sistema de monitoreo, supervisión y control automático del proceso de destilación de agua en el área de inyectables de LIFE C.A.

Alan D. Cuenca S. y Milton S. León E.

Resumen — El presente artículo describe el diseño e implementación de un sistema desarrollado para el monitoreo, supervisión y control del proceso de destilación de agua, empleando un autómata programable y un panel táctil de marca SIEMENS, además del protocolo de comunicación Industrial Ethernet TCP/IP. Este sistema almacena información recopilada del proceso en una base de datos creada en SQL Server 2005. La interfaz gráfica de la pantalla táctil permite la visualización de los datos y variables más relevantes y generar reportes en hojas de cálculo de Microsoft Excel. Este sistema ha sido orientado para ser utilizado como herramienta que facilite a administradores a conocer la producción y funcionamiento del área de Inyectables y mediante el análisis de los reportes la toma de decisiones.

I. INTRODUCCIÓN

La imposibilidad de las empresas para tomar decisiones correctas puede ser resultado de la falta de datos confiables relacionados a la productividad industrial. En la actualidad grandes empresas han optado por la utilización de sistemas automatizados así como la implementación de herramientas que permitan un correcto análisis de sus procesos para reducir las pérdidas y por consecuente incrementar los niveles de producción.

La automatización de los procesos de producción consiste en la búsqueda del hardware adecuado para la aplicación, el diseño del software para el cumplimiento de diversas sentencias de control, seleccionar variables de interés real para su almacenamiento como datos históricos y la aplicación de sofisticadas herramientas de análisis,

todo esto se complementa con el desarrollo de bases de datos.

Los procesos automatizados aportan numerosas ventajas a la producción industrial. Un proceso de fabricación automatizada en la industria hoy en día, significa un producto final de mayor calidad y más competitivo debido a factores tales como la normalización de procesos y productos, la velocidad de producción, programación de la producción, la reducción continua de los residuos y menos probabilidades de equivocarse. Evita el contacto del producto con los recursos humanos, cuyos resultados van desde más higiene en el proceso, alta confiabilidad hasta un elevado nivel de calidad del producto final.

El gran desarrollo de la automatización de procesos en la actualidad permite disponer de controladores lógicos programables (PLC's), sistemas SCADA, sistemas de control distribuido, y servidores OPC que junto a paquetes de software capaces de realizar tareas de adquisición, almacenamiento y presentación de datos históricos, ofrecen características de mejoramiento productivo en un alto y sofisticado nivel.

Las principales desventajas de estos sistemas son el elevado costo de implementación debido a la adquisición de hardware y software con licencias de última tecnología que permitan un correcto y eficaz funcionamiento de los mismos.

El desarrollo del sistema de monitoreo, supervisión y control busca obtener un alto nivel de producción industrial en la sección de Inyectables de mayor volumen, así como un sistema confiable y adaptable al medio industrial farmacéutico ecuatoriano.

II. PROCESO DE DESTILACIÓN DE AGUA [1]

Alan D. Cuenca S., Milton S. León E., Ingeniería en Electrónica, Automatización y Control, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador, E-mails: alandcuenca@hotmail.com, milen37@hotmail.com.

El agua usada en la industria farmacéutica para los inyectables se denomina AGUA INYECTABLE (Water Injection) (Fig. 1)



Fig. 1 Agua Inyectable

La destilación de los sistemas de agua farmacéutica es uno de los puntos críticos en la industria, por no decirlo de vital importancia, no solo por las grandes cantidades de agua que se consumen sino también por la naturaleza propia del agua, ya que puede caracterizarse e identificarse por el tipo y cantidad de sustancias que contiene. Estos atributos de calidad no siempre pueden medirse antes de su uso ya que la contaminación microbiana necesita varios días para poder determinarse y durante ese tiempo, el agua se utiliza, los productos se fabrican y los costes suben. La destilación es esencial para prevenir la necesidad de retirar o destruir productos por problemas de contaminación causados por el agua.

En la empresa LIFE el equipo más importante de este proceso es el destilador de agua OLSA QV 2000 (Fig. 2), el cual trabaja automática y continuamente y permite lograr una producción de 2000 lt/h de agua destilada.



Fig. 2 Destilador de agua OLSA QV 2000

III. DISEÑO DEL SOFTWARE

Para el desarrollo del sistema de este proyecto se han utilizado software tanto para el desarrollo del programa de control del PLC, para la pantalla táctil (HMI) y para la aplicación del almacenamiento continuo de los datos obtenidos del proceso de destilación de agua.

Así pues para desarrollar el programa del PLC SIEMENS S7-300 se ha empleado el software SIMATIC STEP 7 [2], este software de programación, cuyas principales ventajas son la instalación y configuración sencilla, posee herramientas de programación muy comprensibles, fácil corrección de errores, herramientas para manipular cambios en el PLC mientras entra a RUN y configuración de bloques PID que se usan en este proyecto. En la Fig. 3 se presenta una visión global del software SIMATIC.

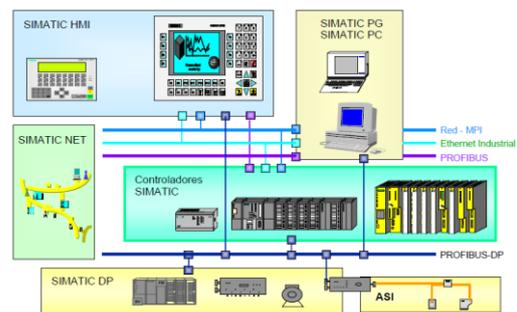


Fig. 3 Visión global del software SIMATIC

Para controlar la variable de presión requerida en el proyecto, el control se realiza con las funciones PID que están integradas en el PLC, en este caso la función que se utilizará es la FB 41 CONT_C (Fig. 4), que sirve para la regulación de procesos industriales con magnitudes de entrada y salida continuas. La función FB 41 CONT_C puede regular fácilmente procesos como el nivel, la presión o el caudal.

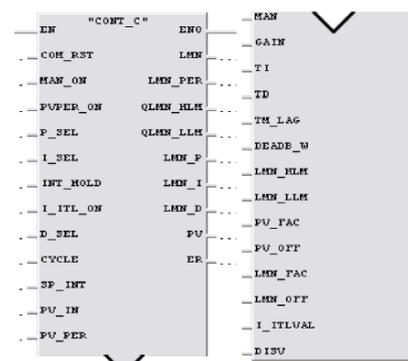


Fig. 4 Aspecto de la función FB 41 "CONT_C"

Para la realización de la estación de monitoreo y supervisión se diseñó una Interfaz Hombre-Máquina o HMI ("Human Machine Interface") que es la que permite presentar los datos más relevantes a un operador (humano) (Fig. 5) y a través de la

cual este controla el proceso, se utilizó el software Wincc Flexible Advanced 2008.

WinCC Flexible [3] también permite la posibilidad de integrar el proyecto del panel de operador al proyecto de Step7 de Siemens, y en nuestro caso, se ha integrado al proyecto de la línea del PLC S7-300 donde hace de interfaz.

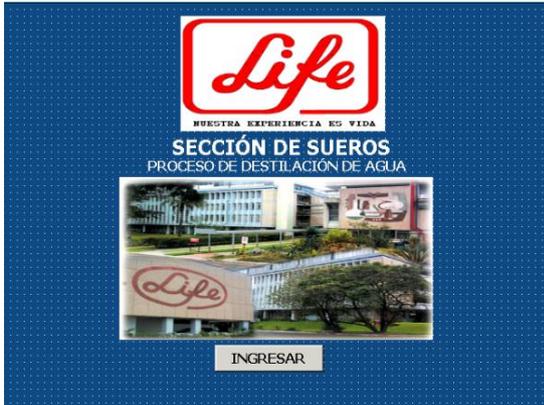


Fig. 5 Interfaz Humano-Máquina (HMI)

El sistema de Monitoreo y Visualización está dividido en once pantallas o ventanas (Fig. 6), en las que están involucradas todas las variables del proceso. En todas las pantallas existen botones que permiten navegar entre ellas.

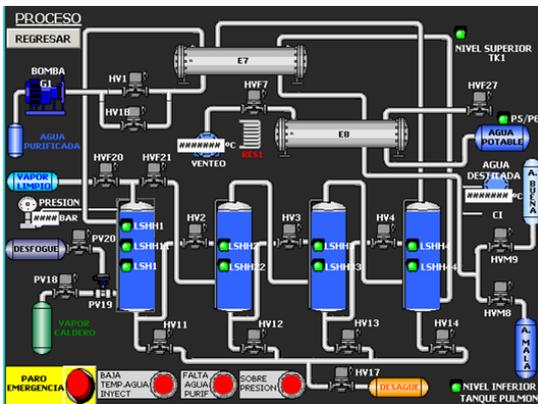


Fig. 6 Ventana del Proceso de Destilación

Para la visualización y control de las variables del proceso (presión y temperatura) en tiempo real se dispone de ventanas con curvas gráficas tal como lo muestran las Figuras 7 y 8.

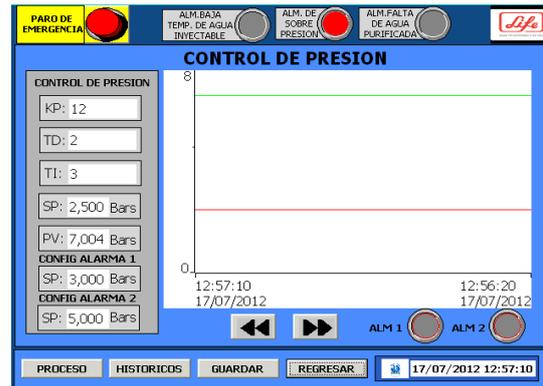


Fig. 7 Ventana de Control de Presión

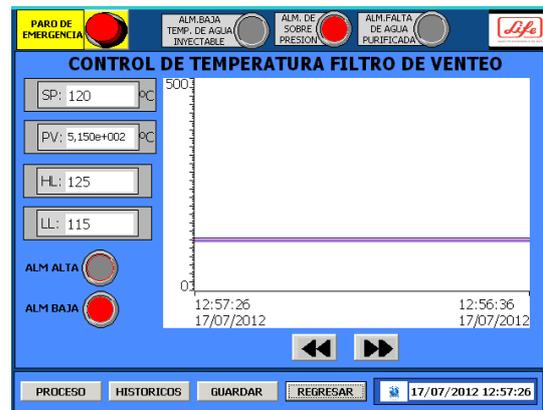


Fig. 8 Ventana de Control de Temperatura

En la ventana configuración de parámetros el operador con la debida autorización puede modificar los valores más relevantes de control para cada una de las variables de proceso (Fig. 9).



Fig. 9 Ventana de configuración de parámetros

Para la creación de los reportes en hojas de MS Excel (Fig. 10) se utilizó Scripts de WinCC flexible ("Storage_Tag_V1" y "Script_Storage_Path"), los cuales permiten disponer de las variables temperatura y presión. Los datos recopilados del proceso estarán

disponibles en el computador destinado para el proceso de destilación de agua.



Fig. 10 Reporte .CSV generado

Para la base de datos se ha utilizado el software SQL Server 2005 [4] el cual permitirá almacenar en una tabla todas las variables de proceso más relevantes tal como lo muestra la Fig. 11.

Fecha	PRESION_BARES	Mp	Td	Ti	T_Filtro_Vertero_°C	T_Agua_destilad
07/07/2010 11:07	2.5	6	10	1	100	70.125
05/08/2012 12:32	3.4	5	4	2	250	703.125
05/08/2012 12:32	3.4	5	4	2	250	703.125
05/09/2012 12:32	3.4	5	4	2	250	703.125
05/10/2012 12:32	2.5	6	10	1	100	70.125
05/12/2012 12:32	2.5	6	10	1	100	70.125
05/08/2012 12:35	2.5	6	10	1	100	70.125
05/08/2012 12:32	3.4	5	4	2	250	703.125
05/08/2012 12:32	3.4	5	4	2	250	703.125
05/09/2012 12:32	3.4	5	4	2	250	703.125
05/10/2012 12:32	2.5	6	10	1	100	70.125
05/10/2012 12:32	2.5	6	10	1	100	70.125
06/11/2013 13:33	3.4	6	10	1	100	70.125

Fig. 11 Tabla de la Base de datos

Para la captura de datos operacionales (Fig. 12) se utiliza el software PC Access el cual permite realizar una comunicación entre el PLC con programas de cómputo tales como: Excel y Visual Basic de esta forma es posible archivar las variables de proceso presión y temperatura en una hora y evaluarlas posteriormente.

Fecha	PRESION_BARES	Mp	Td	Ti	T_Filtro_Vertero_°C	T_Agua_destilad
12:32:00	3.4	5	4	2	250	703.125
12:32:00	3.4	5	3	2	250	703.125
10:00:00	3.4	5	4	2	250	703.125
10:45:00	2.5	6	10	1	100	70.125
12:00:00	2.5	6	9	1	100	70.125
11:35:00	2.5	6	10	3	100	70.125
10:32:00	3.4	6	5	1	98	70.125

Fig. 12 Captura de datos operacionales

IV. DISEÑO DEL HARDWARE

El sistema debe ser capaz de controlar todo el proceso de destilación de agua por simple efecto por medio de un PLC el cual gobierna los estados de DESTILACIÓN, GENERACIÓN DE VAPOR, LLENADO, DESCARGA, ABRIR VÁLVULA DE VAPOR Y SANITIZACIÓN, además este controlador debe tomar las señales de los respectivos sensores de nivel, presión y temperatura (Figura 3.1), una vez procesada la información de acuerdo a un modo de control configurado se envían instrucciones a las respectivas electroválvulas que permiten el ingreso de agua y vapor necesarias para los efectos del destilador de agua. Será posible visualizar las variables del proceso y su respectivo control gracias a una pantalla táctil que contendrá una HMI de configuración, sintonización y monitoreo del sistema.

El diseño de cada una de las etapas de hardware con las funciones se describe a continuación:

A. Señales de los sensores

El destilador OLSA QV 2000 dispone de 9 sensores de nivel, 3 en el primer tanque (Efecto 1), 2 en el segundo tanque (Efecto 2), 2 en el tercer tanque (Efecto 3) y 2 en el cuarto tanque (Efecto 4) de estos sensores se registra la señal de 0VDC y 5 VDC (desactivación y activación correspondientemente) para el control de los estados del proceso.

Además dispone de 2 sensores de temperatura, cuyas señales de corriente de 4 mA – 20 mA también son registradas, la primera es escalada para la realización del control ON-OFF cuando una válvula de filtro es activada en los estados de Destilación, Llenado, Descarga y Generación de Vapor, la señal del segundo sensor de temperatura en cambio es registrada para indicación en la HMI de la temperatura de agua destilada a la salida del proceso. Por último el destilador cuenta con un sensor de presión cuya señal de 4mA-20 mA es registrada, calibrada y escalada para la realización del control de la válvula de corte de vapor en el estado de Destilación y Generación de vapor.

B. Módulos de entradas y salidas

De acuerdo a las señales de los respectivos sensores anteriormente mencionados se puede determinar el tipo de módulos a utilizar para registrar dichas señales, así tenemos: Módulos SIEMENS de entradas digitales SM 321, salidas digitales SM 322, entradas analógicas SM 331 y salidas analógicas SM 332.

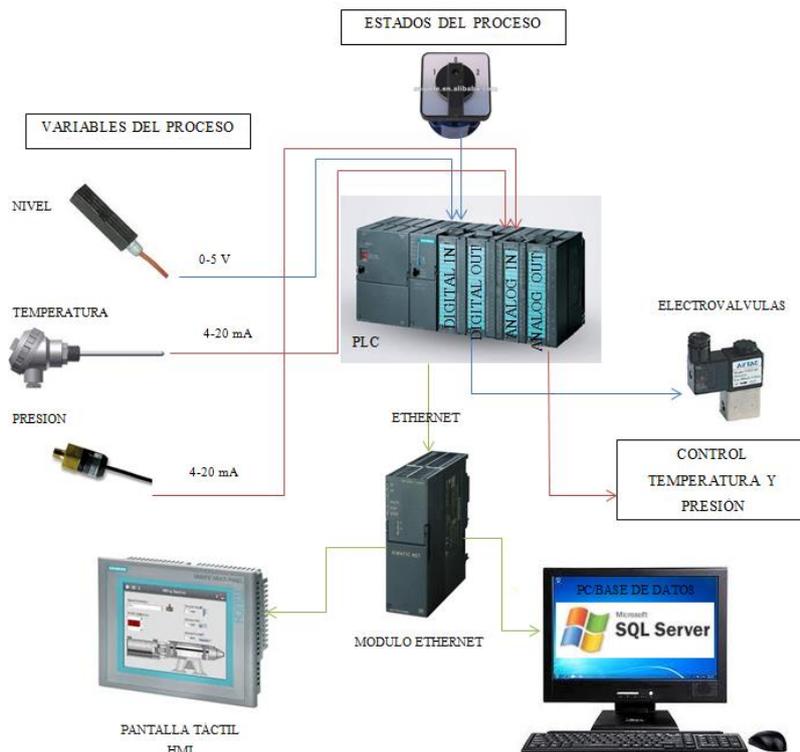


Fig. 13 Etapas de Hardware

C. PLC SIEMENS S7-300 [5]

De acuerdo a requerimientos de la empresa y a las condiciones de trabajo donde se va a desarrollar el proyecto el controlador lógico programable empleado será el SIEMENS S7-300, que cumple con todas las necesidades del sistema a desarrollar. Posee un CPU 313C-2DP con un canal de 16 entradas digitales de 24 VDC, alimentación de 24VDC, 16 salidas a relé, y un puerto periférico para conexión de módulos de entradas/salidas y de comunicación.

D. Módulo de comunicación SIEMENS CP 343-1 Lean

Dispositivo periférico de la línea SIEMENS, este módulo permite conectar el SIMATIC S7-300 a la red Industrial Ethernet. Como dispone de procesador propio, descarga a la CPU de tareas de comunicación y permite establecer conexiones adicionales la cual permite establecer la transferencia en entornos industriales a largas distancias.

E. Panel Táctil MP 277 10"[6]

Los Multi Panels (MP) se utilizan como los paneles de operador, para el manejo y la monitorización de máquinas y procesos, además se utilizan en las más diversas ramas y aplicaciones y se pueden ampliar en su aplicación gracias a las

opciones Multipanel. La pantalla elegida para este proyecto es una Multipanel de 10 pulgadas con pantalla a color TFT y táctil.

F. Computador

En el ordenador del proceso se encontrarán funcionando de manera ininterrumpida las aplicaciones para la adquisición y almacenamiento de datos más relevantes (variables del proceso).

Los requerimientos mínimos del ordenador son:

- Procesador: Intel Pentium IV CPU 3.40GHz
- Memoria Ram: 512 MB
- Espacio libre en disco: 20 GB
- Sistema operativo: Windows XP Versión 2002 Service Pack 3
- Software: - Microsoft Office 2007, SQL Server 2005 y Visual Studio 2008

V. RESULTADOS

Para comprobar el correcto funcionamiento del proyecto desarrollado se realizaron dos tipos de pruebas de cada una de las etapas del sistema.

A. Pruebas de hardware

Entre las pruebas realizadas, se han considerado el correcto funcionamiento del

controlador lógico programable y sus respectivos módulos de entradas/salidas, así como la comunicación entre el autómatas, el panel táctil y el ordenador.

En la prueba de funcionamiento del PLC y sus módulos se activaron y calibraron los respectivos sensores y electroválvulas de acuerdo a lo estados del proceso. Estas pruebas se pueden observar en las Figuras 14, 15 y 16.



Fig. 14 Pruebas de las señales digitales

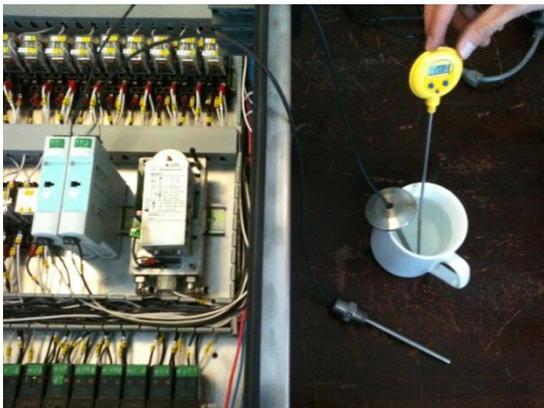


Fig. 15 Calibración de los sensores de temperatura



Fig. 16 Calibración del transductor de presión

Para la prueba de comunicación se montó la red Ethernet a través de la conexión del PLC, el panel táctil y el computador a un switch (Fig. 17) asignando a cada uno su respectiva dirección IP. Para verificar que la comunicación se realizó correctamente se ejecuto el comando ping (dirección IP) en el cmd de Windows del PC del proceso para la detección automática de dispositivos.

Este enlace se realizó correctamente y los dispositivos pudieron establecer una comunicación sin interrupciones.

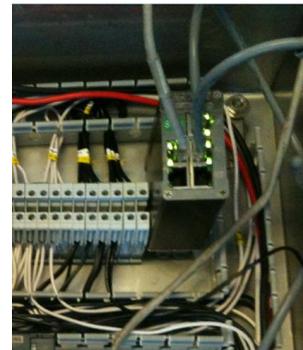


Fig. 17 Comunicación Ethernet entre PLC, PC y Pantalla Táctil

B. Pruebas de software

En esta prueba se verificó la Interfaz de Monitoreo, Supervisión y Control diseñada para el proceso, así pues se cargo la HMI (Fig. 18) en el panel táctil para comprobación del funcionamiento de las diferentes funciones que van desde los controladores de las variables del proceso (Fig. 19), generación de históricos, hasta la generación de reportes y su almacenamiento en una base de datos (Fig. 20).



Fig. 18 Programa HMI cargado en la pantalla táctil

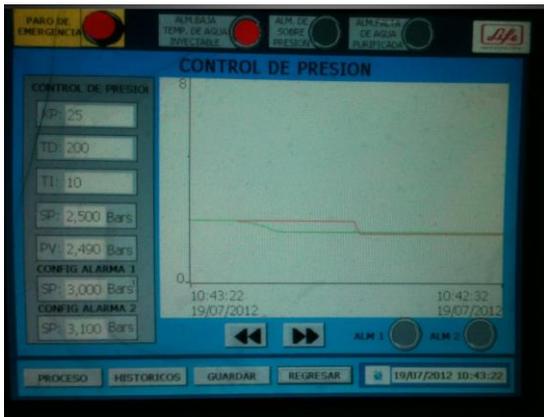


Fig. 19 Verificación del funcionamiento del Controlador PID de Presión

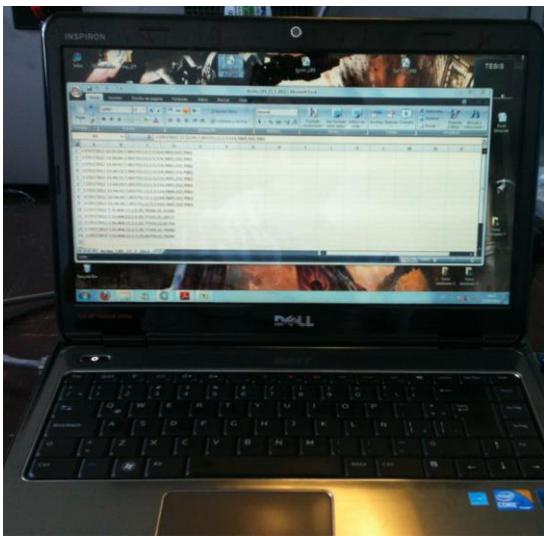


Fig. 20 Pruebas de generación de reportes en MS Excel y base de datos

Todas las pruebas se llevaron exitosa y correctamente permitiendo establecer la validez y aprobación del sistema para su respectiva implementación (Fig. 21).



Fig. 21 Nuevo sistema de monitoreo, supervisión y control del proceso de destilación de agua

VI. CONCLUSIONES

Se desarrolló un sistema de monitoreo, supervisión y control del proceso de destilación de agua empleando un PLC SIEMENS, un panel táctil y software de programación de manera correcta y precisa.

Además, se diseñó una interfaz gráfica sencilla y fácil de entender en una pantalla táctil empleando herramientas de software compatibles que facilita la supervisión, visualización, monitoreo en tiempo real y control de operación de los equipos del proceso de destilación de agua.

Se configuraron y sintonizaron de manera precisa el controlador PID de presión y el controlador de temperatura del filtro de venteo según las necesidades del proceso.

A través de la creación de reportes en hojas de cálculo en Excel y la base de datos se facilitó el análisis de los valores registrados de las variables del proceso, además resultaron herramientas muy importantes para los operadores ya que permiten incrementar y mejorar los niveles de producción.

VII. REFERENCIAS

- [1] Pagina Oficial Laboratorios Farmacéuticos del Ecuador LIFE C.A. [Online]. Habilitado: http://laboratorioslife.com/templates/life/_html/productos.html
- [2] Software para controladores SIMATIC [Online]. Habilitado: http://www.automation.siemens.com/salesmaterial-as/brochure/es/brochure_simatic-industrial-software_es.pdf
- [3] SIMATIC HMI WinCC flexible 2008 Compact / Standard / Advanced.pdf
- [4] Cómo hacer una base de datos sencilla en SQL SERVER 2005... [Online]. Habilitado: <http://jotask8punk.wordpress.com/otras-tecnologias/como-hacer-una-base-de-datos-sencilla-en-sql-server-2005%E2%80%A6/>
- [5] Sistemas de automatizacion industrial SIMATIC [Online]. Habilitado: <http://www.distribuidor-oficial-siemens-productos-electricos.control-technics.com.ar/distribuidor-siemens/es/Notas%20y%20articulos/Automatizacion/SIMATIC/1682/Sistemas%20de%20automatizacion%20industrial%20SIMATIC.htm>
- [6] SIMATIC HMI Panel de operador MP 277 (WinCC flexible) Instrucciones de servicio.pdf

Diseño e Implementación de un Multiprocessor Systems-on-Chip (MPSoC) Interconectado por una Networks-on-Chip (NoC)

Wilson Mauricio Chicaiza y Daniel Gonzalo Verdezoto

Resumen— En el presente documento se presenta una breve caracterización de los medios de comunicación empleados en arquitecturas multiprocesadas. Esta caracterización tiene como objetivo principal el mostrar un nuevo modelo de comunicación basado en conmutación de paquetes a los cuales se les denomina como Networks-On-Chip (NoC). Esta publicación muestra una arquitectura de red llamada NoC Hermes, la cual fue interconectada a un Multiprocessor-Systems-on-Chip (MPSoC) compuesto de cuatro procesadores MicroBlaze. Esta conexión se la realizó gracias al diseño y desarrollo de una Interfaz de Red generada en código VHDL. Por medio de la Interfaz de Red se consiguió que los procesadores MicroBlaze interactúen con los Switches de Hermes a fin de crear una arquitectura multiprocesada interconectada por una NoC. Con el motivo de realizar comparaciones también se creó otra arquitectura de multiprocesadores interconectados por buses. Para ambas arquitecturas se desarrolló una aplicación de Esteganografía en la que existe multiprocesamiento de dos procesadores trabajando simultáneamente. Lamentablemente sobre dicha aplicación no fue posible medir directamente la latencia y el consumo de energía, razón por la cual se utilizó simuladores que permitieron estimar dichas mediciones.

Palabras Claves; *Buses, System-on-Chip (SoC), Multiprocessor System-on-Chip (MPSoC), IP-Core (Bloque de Propiedad Intelectual), Networks-on-Chip (NoC).*

I. INTRODUCCIÓN

En la actualidad, el nivel de integración en los circuitos integrados está creciendo rápidamente gracias a la utilización de nuevas técnicas y herramientas de diseño e implementación, lo que ha permitido disminuir la brecha entre la capacidad de integración de transistores dentro de un chip y el diseño de los mismos. Sin embargo, aparecen nuevos desafíos para los diseñadores de dispositivos electrónicos modernos que requieren de la implementación de sistemas completos dentro de un circuito integrado SoC [1]. Entre los principales retos a los que se enfrentan, se pueden nombrar la disipación de potencia, la disminución de la latencia y el aumento del ancho de banda en la comunicación intrachip.

Para satisfacer los requerimientos del mercado actual, se utilizan sistemas embebidos basados en SoC que incorporan un solo procesador o chip maestro. A fin de solventar una aplicación de alto nivel computacional, se utilizaban varios SoCs que debían compartir el control y procesamiento general de la aplicación creándose un problema de paralelismo, desperdicio de transistores, consumo de potencia y limitación física. De allí que aparece la necesidad de desarrollar los MPSoC.

El MPSoC es un SoC compuesto por múltiples procesadores, memorias y circuitería especializada,

W. M. Chicaiza estudió Ingeniería Electrónica, Automatización y Control en la Escuela Politécnica del Ejército, Sangolquí, Ecuador (cmoswilson@gmail.com)

D. G. Verdezoto estudió Ingeniería Electrónica, Automatización y Control en la Escuela Politécnica del Ejército, Sangolquí, Ecuador (dgverdezoto@gmail.com)

interconectados a través de una infraestructura de comunicaciones.

Tradicionalmente, los MPSoC se han comunicado a través de arquitecturas basadas en buses. Lamentablemente, los buses no satisfacen los requerimientos actuales debido a que la convergencia de aplicaciones demanda la mezcla de varios tipos de tráfico en el mismo SoC. Es así que al compartir el medio de transporte, se requiere manejar el concepto de QoS. Por tanto, las arquitecturas modernas exigen una estructura flexible, reconfigurable y reprogramable. Esta exigencia ha dado paso al nacimiento de un nuevo paradigma de interconexión, el empleo de Networks-On-Chip. [2]

En este ámbito, la arquitectura NoC se proyecta como una solución prometedora en la comunicación On Chip. Características como la eficiencia energética, fiabilidad, escalabilidad, ancho de banda y reutilización, son factores que sustentan su desarrollo. Cabe señalar que, las NoC emplean teorías y métodos de comunicación por medio de redes de conmutación de paquetes dentro del chip, introduciendo notables mejoras. Los módulos intercambian información entre sí gracias a la intervención de las NoC que establecen una red como un subsistema de transporte público para el tráfico de información. De esta forma la información pasa por varios enlaces, de acuerdo a las decisiones que se tomen dentro del algoritmo de la NoC. [3]

II. MULTIPROCESSOR SYSTEMS-ON-CHIP

Un MPSoC representa una rama distinta de multiprocesadores tradicionales que cumplen con requerimientos específicos de aplicaciones embebidas que utilizan varios procesadores interconectados junto con otros subsistemas de hardware implementados en un solo sistema. [4]

Un MPSoC incorpora varios elementos de procesamiento necesarios para una aplicación que utiliza multiprocesadores programables como parte de los elementos de un sistema, unidos por una estructura de interconexión, ajustándose a las necesidades y evolución de la arquitectura de computadores, tales como: operación en tiempo real, consumo de baja potencia y aplicaciones multitarea. [5]

A. Tipos de MPSOC

En el estudio de MPSoC se argumenta la formación de dos importantes familias de multiprocesadores: La familia de Multiprocesadores Homogéneos cuyo primer prototipo fue Lucent Daytona [6] [4] y la familia de Multiprocesadores Heterogéneos, en la que destaca el C-5 Network Processor [7], Nexperia y OMAP [8].

TABLA 1. COMPARACIÓN DE MPSOC HOMOGÉNEOS Y HETEROGÉNEOS.

Homogéneos	Heterogéneos
Son conjuntos de procesadores basados en el modelo de arquitectura computacional paralela.	Están conformados por procesadores interconectados con diferentes funcionalidades.[6]
Con la arquitectura paralela se divide el tiempo de ejecución de cada recurso, para hacer más eficiente el tiempo de procesamiento.[7]	Cada uno de los procesadores está dedicado a un propósito específico, como aceleradores de hardware, procesadores digitales de señales, periféricos, etc.
Los MPSoC Homogéneos son utilizados comúnmente en consolas de videojuegos, servidores y computadores de escritorio.	Los MPSoC Heterogéneos tienen aplicaciones en sistemas portátiles y en sistemas integrados.

B. Arquitecturas MPSOC

La gran capacidad de implementar varios procesadores dentro de un mismo diseño, permite desarrollar diversas arquitecturas dentro de los MPSoC, con la finalidad de aumentar el rendimiento de una aplicación específica.

TABLA 2. DESCRIPCIÓN DE LA ARQUITECTURAS EXISTENTES EN MPSOC.

Arquitecturas de MPSoCs	Descripción
1. Procesadores no interconectados.	Incluyen varios procesadores pero cada uno se dedica a una tarea independiente y no existe comunicación entre ellos. Pudiendo ser el caso que los procesadores sean diferentes.
2. De propósito general y específico.	Son MPSoC interconectados en los que se hace necesario la ejecución de tareas de forma conjunta y el intercambio de información entre los procesadores, mediante una memoria local (2) o a una compartida (3).
3. Memoria Compartida.	
4. Configuración Maestro – Esclavo.	Todos los procesadores esclavo ejecutan el mismo código y es el procesador maestro el encargado de recibir la información y repartirla entre los esclavos para su procesamiento.
5. Segmentada.	Cada procesador realiza una parte de la tarea y pasa sus resultados al siguiente procesador. Esta arquitectura se puede utilizar en aplicaciones en donde no existe paralelismo, sin embargo pueden dividir las tareas entre los procesadores de forma independiente.[13]

III. INTERCONEXIÓN EN UN MPSOC

La arquitectura de comunicación en un MPSoC es uno de los principales retos tecnológicos en la actualidad debido al aumento de frecuencias de operación y el incremento del tamaño del chip. Por lo cual la dependencia de rendimiento en el diseño de un MPSoC recae en la eficiencia de su estructura de comunicación [9]. Por ende el presente documento presenta a continuación dos modelos de MPSoC, una interconectada por buses FSL (Fast Simple Link) y la otra interconectada por una NoC, que fueron diseñados para compararse entre sí y demostrar cual se desempeña mejor, a través de resultados obtenidos en latencia y disipación de potencia por medio de simulaciones.

Cabe mencionar que todo el trabajo de creación del MPSoC se lo realizo en un FPGA (Field Programmable Gate Array) Virtex 6 ML605, que posee un encapsulado XC6VLX240T-1FFG1156, y que además incluye el software EDK 13.2 para el co-diseño de hardware y software, el cual provee un entorno flexible para el diseño de aplicaciones de alto rendimiento. [10][11]

A. Interconexión mediante buses FSL

La implementación de buses para la interconexión de IP-Cores dentro un MPSoC es la más simple y ampliamente utilizada. De la gran variedad existente en buses se utiliza el FSL, que es un bus que se emplea para la comunicación dentro de un MPSoC, debido a que puede soportar efectivamente topologías como mesh, anillo y estrella, a media que se incrementa el número de procesadores en el FPGA [12]. Se menciona además que este bus es utilizado también para la comunicación de IP-Cores de diseño específico (dentro del XPS, Xilinx Platform Studio) como se verá más adelante en la interconexión de la NoC.

La arquitectura del hardware que ha sido constituido se detalla en la Figura 1 en donde se puede apreciar la interconexión de los elementos necesarios para cumplir con la aplicación de Esteganografía, de la cual se hablará posteriormente.

Esta arquitectura consta de 4 procesadores MicroBlaze comunicados entre sí por buses FSL. Destaca en esta arquitectura la Interfaz MPMC (Multi-Port Memory Controller), que es un IP-Core que permite la comunicación entre los procesadores con una memoria externa, en este caso una DDR3 de 512MB, soporta hasta 8 conexiones (un par para cada MicroBlaze) de manera personalizada, haciendo uso de la interfaz XCL o bien por conexión directa al bus PLB [12] [14]. Por XCL se asigna un acceso de memoria directa para el procesador y por bus PLB la memoria se comparte para todos los procesadores.

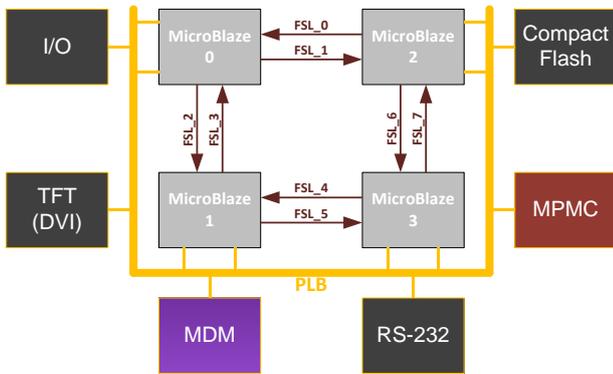


Figura 1. Arquitectura de 4 procesadores embebidos interconectados por Buses FSL para aplicación de Esteganografía

Los componentes más relevantes en esta arquitectura además del MPMC, son el TFT (Thin Film Transistor) y el lector de CF, los cuales se detallan a continuación:

- El IP-Core TFT, es el medio por el cual se controla una pantalla LCD o CTR, consta de una conexión maestra conectada de forma independiente hacia la memoria DDR3 y una conexión esclava conectada al bus PLB para adquirir la información de los pixeles que serán proyectados. El TFT además trabaja a una frecuencia de 25MHz con una resolución de 640x480 pixeles y tolera conexiones para DVI (Digital Video Interface) y VGA (Video Graphics Array) [15].
- El lector de CF (Compact Flash) o SYSACE (System ACE Interface Controller), es un IP-Core flexible que permite obtener información de una memoria CF [16]. El SYSACE usualmente es empleado para verificar que el FPGA está funcionando de manera correcta al tener acceso a archivos ejecutables ubicados dentro de la CF que permiten realizar un test de periféricos. Para el caso de la arquitectura mostrada en la Figura 1 el módulo SYSACE fue adaptado para poder adquirir información necesaria para la aplicación de Esteganografía.
- Los demás IP-Cores que también forman parte de la arquitectura son: dos General Purpose IO (GPIO's) constituidos por los Led 8 Bits y Dip-Switchs, un RS232_Uart16550 y un Microblaze Debug Module (MDM). Los dos últimos IP-Cores son elementos indispensables en cualquier diseño, pues el RS232 permite una comunicación entre el computador y el FPGA [17], y el MDM provee soporte para depuraciones en el software que se está desarrollando desde uno hasta ocho procesadores MicroBlaze [18].

B. Interconexión mediante NoC Hermes

En la Figura 2 se detalla la interconexión de los cuatro MicroBlaze interconectados por NoC Hermes. El proceso para el desarrollo de esta arquitectura es similar a la realizada en la conexión de buses, salvo por el IP-Core que fue desarrollado por la Facultad de Informática de la Pontificia Universidad

Católica de Rio Grande (Brasil) y adaptado a través de una NI desarrollada en la Escuela Politécnica del Ejército.

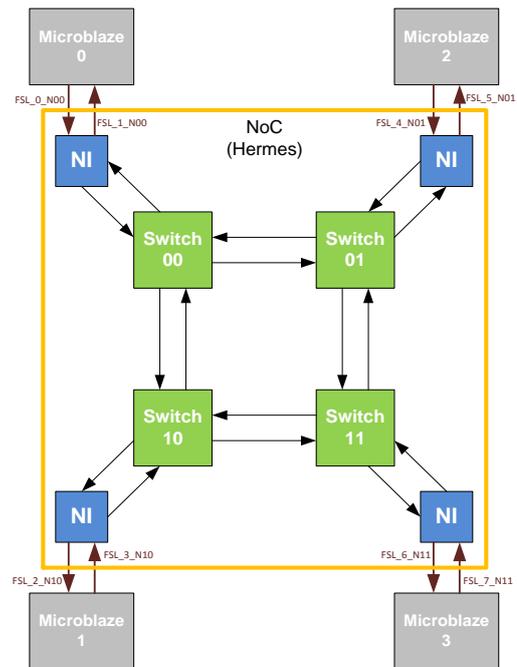


Figura 2. Arquitectura de 4 de procesadores interconectados por una NoC para aplicación de Esteganografía

1) Diseño y desarrollo de una NI para NoC Hermes

Para el planteamiento del diseño se parte del modo de operación de los elementos involucrados. Uno es el medio por el cual se puede transmitir información desde el MicroBlaze hacia un IP-Core cualquiera, este medio es el bus FSL, y el otro es el Switch Hermes [19] al que le hace falta una interfaz de red que le permita interactuar con el procesador a través del mencionado bus.

a) Operación FSL

FSL es un bus unidireccional de punto a punto utilizado para aplicaciones desarrolladas en FPGA's. La interfaz FSL puede ser habilitada desde el propio MicroBlaze y se lo utiliza para transferir datos hacia y desde los archivos de registro del procesador al hardware implementado. [20]

Los puertos de entrada y salida del bus FSL que son utilizados en el diseño se especifican en la siguiente tabla.

TABLA 3. PUERTOS DEL FSL NECESARIOS PARA EL DISEÑO.

Fast Simple Link (FSL)			
Maestro	I/O	Esclavo	I/O
FSL_M_DATA	In	FSL_S_DATA	Out
FSL_M_WRITE	In	FSL_S_READ	In
FLS_M_FULL	Out	FSL_S_EXISTS	Out

De la tabla anterior se analiza el modo de operación, empezando por el Maestro, en donde la forma de transmisión de datos (FSL_M_DATA) se la realiza cuando se activa en

alto el FSL_M_WRITE, y mientras se mantenga así los datos se mantendrán circulando a través del FSL Maestro hasta que el FSL_M_WRITE vuelva a un nivel bajo o se emita una señal en alto por el FSL_M_FULL indicando que el bus está lleno [20].

En el Esclavo los datos (FSL_S_DATA) se trasladan cuando el FSL_S_EXISTS se activa en alto y únicamente bajo esta condición se puede activar el FSL_S_READ, caso contrario si no existen datos en el bus, el FSL_S_READ se mantendrá en un nivel bajo [20].

b) Operación Switch Hermes

La topología de Hermes es una 2D-Mesh que emplea control de flujo por Handshake y técnica de ruteo por algoritmo XY para la realización de la transmisión de información desde un nodo de origen hacia un nodo de destino al conmutar los paquetes por medio de la técnica Wormhole. [19]

La distribución de entradas y salidas presentes en el puerto local del Switch Hermes se muestran en la siguiente tabla.

TABLA 4. PUERTOS DEL SWITCH HERMES NECESARIOS PARA EL DISEÑO.

Switch Hemes (Local Port)			
Receptor	I/O	Transmisor	I/O
rxlocal	In	txlocal	Out
data_inlocal	In	data_outlocal	Out
ack_rxlocal	Out	ack_txlocal	In

En el nodo de origen se emplea Handshake a través de rx y ack_rx, en donde cada flanco positivo va acompañado de un flit del paquete de datos por data_in. Este paquete va encabezado por la dirección de destino y por el tamaño del paquete. [19]

Entonces cada vez que se hace una transmisión se pone en alto la señal rx y a través de ack_rx se confirma que el dato transmitido por data_in ha sido enrutado, así sucesivamente hasta que todo el paquete de datos haya sido enviado. [19]

Los datos al ser receptados en el nodo de destino presentan el mismo comportamiento, es decir que tx se activa en alto cada vez que llega un nuevo dato, del mismo modo la recepción se confirma por medio de ack_tx. La recepción de los flits finaliza una vez que se haya receptado todo el paquete [19].

c) Desarrollo de la Network Interface (NI)

Entre todas las posibles arquitecturas que pueden existir en diferentes literaturas que tratan sobre NoC, se puede distinguir la presencia de tres principales componentes que lo conforman, los cuales son los elementos de red (Networks Elements, NE) (también llamados routers o switches), las interfaces de red (Networks Interface, NI) (llamados también adaptadores de red) y los nodos o elementos de proceso (Process Elements, PE) [21]. De estos componentes, el interés se centra en el NI puesto que es un elemento faltante que tuvo que ser creado para la arquitectura de red que aquí se expone.

Dentro de las herramientas de co-diseño de hardware y software (EDK, Embeddd Development Kit) se encuentra incluido también el ISE, el cual fue utilizado para programar el VHDL del NI que fue diseñado a partir de diagramas de estados, para poder crear las máquinas secuenciales que conforman el NI que se muestra en la siguiente figura.

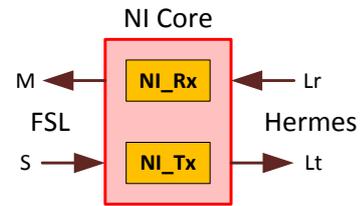


Figura 3. Módulos internos del NI.

En la Figura 3 se puede observar que el NI consta de dos módulos internos NI_Rx y NI_Tx. Estos módulos son las máquinas secuenciales encargadas de administrar la información que proviene tanto de los Switches de Hermes como de los MicroBlaze's. Cada uno de estos subsistemas tiene terminaciones que se conectan al maestro (M) y al esclavo (F) del FSL, y al puerto local de recepción (Lr) y transmisión (Lt) de la NoC.

En la siguiente tabla se puede apreciar todas las entradas y salidas listas para ser conectadas al MicroBlaze (por medio del FSL) y a Hermes (Por medio del Puerto Local), completando así la arquitectura red.

TABLA 5. PUERTOS DE LA NETWORK INTERFACE

Network Interface			
FSL	I/O	Local Port	I/O
FSL_S_DATA	In	Data_out_Local	Out
FSL_S_EXISTS	In	txLocal	Out
FSL_S_READ	Out	Ack_txLocal	In
FSL_M_DATA	Out	Data_in_Local	In
FSL_M_FULL	In	rxLocal	In
FSL_M_WRITE	Out	Ack_rxLocal	Out

2) Interconexión Procesador – Noc a Través se la Network Interface

El NI aquí descrito es integrado a NoC Hermes dentro de la plataforma ISE como un todo, antes de ser importado al XPS, que es el software en donde se encuentra el MPSoC. Ya en el XPS la integración del nuevo IP-Core con el MPSoC es más sencilla.

El proceso de importación se lo realiza desde el XPS al utilizar la herramienta Create or Import Peripheral Wizard, el cual permite a diseños realizados en el ISE tomar la forma de un IP-Core mucho más flexible para su interacción con otros IP-Cores [22].

IV. APLICACIÓN DE ESTEGANOGRAFÍA

La Esteganografía trabaja con meta información, es decir información dentro de otra información y ha sido empleada desde tiempos remotos dentro de la vida del ser humano. En donde a partir de una imagen portadora se ocultan caracteres ASCII codificados en binario dentro de los pixeles que constituyen una imagen. [23]

Para el desarrollo de la aplicación se ha seleccionado el formato PPM por ser ideal para trabajos relacionados con procesamiento de imágenes [24] [25] [26]. Su utilización es sencilla así como la programación realizada para adquirir su información. Además se mantiene bajo los recursos de memoria lo cual lo hace apto para ser empleado en el FPGA Virtex-6.

La aplicación de esteganografía se ejecuta entre cada par de procesadores de forma simultánea. A continuación se explica este proceso.

En la Figura 4 se aprecia una arquitectura de cuatro procesadores MicroBlaze comunicados por buses FSL. El MicroBlaze 0 y el MicroBlaze 2 realizan el proceso de esteganografía de forma paralela sobre dos imágenes distintas. Mientras que el MicroBlaze 1 y el MicroBlaze 3 realizan el proceso de decodificación de forma paralela para extraer la información contenida en las imágenes.

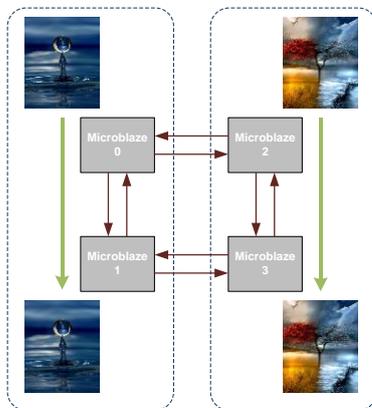


Figura 4. Proceso de Esteganografía y de Decodificación de mensaje sobre arquitectura MPSoC interconectada por buses FSL.

En la Figura 5 se puede apreciar una arquitectura de cuatro procesadores MicroBlaze comunicados por NoC Hermes. El MicroBlaze 0 y el MicroBlaze 2 realizan el proceso de esteganografía de forma paralela sobre dos imágenes distintas. Mientras que el MicroBlaze 1 y el MicroBlaze 3 realizan el proceso de decodificación de forma paralela para extraer la información contenida en las imágenes.

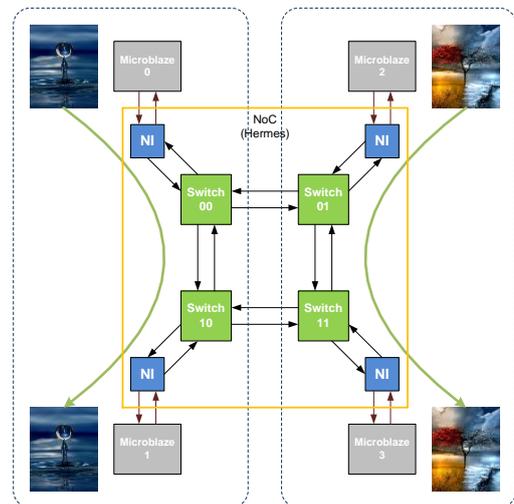


Figura 5. Proceso de Esteganografía y de Decodificación de mensaje sobre segunda arquitectura

En ambas arquitecturas se interrumpe el paralelismo cuando los MicroBlaze se comunican entre sí para acceder a recursos compartidos del FPGA. Los procesadores comparten el acceso a memoria CF para la adquisición de imágenes y mensajes, y la proyección de resultados en un monitor a través del puerto DVI.

V. ANÁLISIS DE RESULTADOS

Las dimensiones de las imágenes empleadas en la aplicación de esteganografía son de 240x320 pixeles, con capacidad para ocultar 25600 caracteres. Por los recursos de memoria del FPGA únicamente se pudo ocultar un 2% de los 25600 caracteres que se pueden contener dentro de la imagen. Esta limitante únicamente permite trabajar con imágenes de un tamaño menor o igual a 240x320 pixeles. Debido a que no se pueden realizar mediciones directas de latencia y consumo de energía sobre el FPGA, se realizaron simulaciones por medio de las herramientas ISim (Simulador del ISE) y XPower Analyzer que están incluidos en el kit de desarrollo EDK. Por medio de estas herramientas se realizaron una serie de pruebas con las cuales se obtienen resultados que se exponen a continuación.

A. Análisis de Resultados en Latencia

a) Análisis de Resultados para NoC Hermes en Dos y Tres Nodos

En la Figura 6 se puede apreciar la curva de tendencia de latencia simulada en NoC Hermes para dos y tres nodos.

Latencia Simulada en NoC Hermes para 2 y 3 Nodos

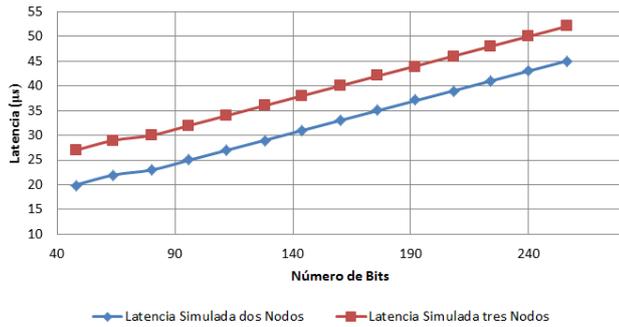


Figura 6. Gráfica de comparación entre NoC Hermes para 2 y 3 nodos.

Como se puede observar en la figura existe un incremento en latencia por aumento del número de bits transmitidos. También existe un diferencial constante debido al número de nodos. Por lo tanto se deben realizar pruebas en topologías de más de cuatro nodos con el objetivo de verificar si la diferencia se mantiene o cambia.

b) Análisis de Resultados entre FSL y NoC Hermes a Dos Nodos

La Figura 7 muestra los valores de los datos simulados para latencia entre NoC Hermes a dos nodos y bus FSL. Esto se debe principalmente a que el tiempo de dispersión es mayor en NoC Hermes con respecto a FSL. Sin embargo estas simulaciones fueron realizadas de punto a punto. Por este motivo se evidencia que es imperceptible evaluar el rendimiento en latencia para una arquitectura de multiprocesamiento MPSoC interconectado por NoC Hermes de 2x2 en topología 2D-Mesh. De manera que se recomienda el desarrollo de trabajos futuros sobre arquitecturas MPSoC más grandes en donde se pueda percibir el potencial real en latencia de NoC Hermes con respecto a buses FSL.

Latencia Simulada en NoC Hermes y en Bus FSL

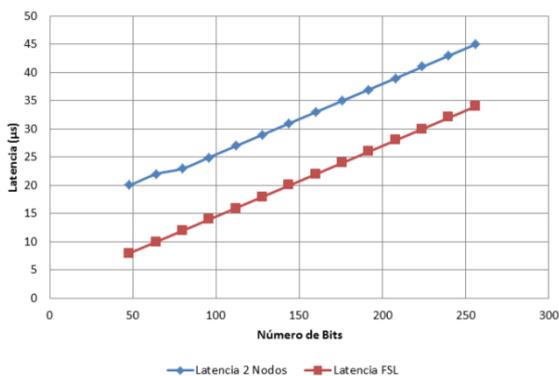


Figura 7. Gráfica de comparación entre NoC Hermes y FSL.

c) Análisis de Resultados entre FSL y NoC Hermes con Interfaz de Red

La Tabla 6 muestra los resultados obtenidos de la simulación con NoC Hermes a dos nodos y NoC Hermes con interfaz de red para un paquete de 80 bits. Dado que solo se hicieron simulaciones, en donde aparentemente se incrementa la latencia, los valores obtenidos no interpretan la realidad.

TABLA 6. DATOS DE SIMULACIÓN EN NOC SIN NI Y NOC CON NI.

Latencia Simulada		
Núm. Bits	NoC Hermes sin NI (µs)	NoC Hermes con NI (µs)
80	23	25

De la tabla anterior se puede concluir que la latencia en NoC Hermes con NI se incrementa en 2µs adicionales. Por lo tanto se demuestra que aparentemente la interfaz de red diseñada incrementa la latencia en el valor mencionado.

Por otra parte la simulación en NoC Hermes con Interfaz de Red también permite tomar en cuenta los valores de latencia sobre el NI diseñado. La Tabla 7 muestra las latencias de los NI00 y NI10.

TABLA 7. LATENCIA EN NI.

Latencia Simulada			
Núm. Bits	FSL (µs)	NI00 (µs)	NI10 (µs)
80	12	11	10

En la Tabla 7 se muestra que la latencia en FSL es de 12µs. Este valor al ser comparado con las latencias de la Tabla 6 muestra claramente que el bus FSL conmuta paquetes de 80 bits de forma más efectiva que NoC Hermes con o sin NI. Además las simulaciones prueban la latencia de 2µs en el FSL. AL bus FSL le toma 1µs para escribir en el lado del maestro y 1µs en leer en el lado del esclavo. Por lo tanto la latencia en el NI00 corresponde a la latencia en lectura del bus FSL. Es decir, que le toma tan solo 1µs menos que al FSL para transmitir información una vez que adquiere los datos. En otras palabras la latencia del NI00 depende mucho del tiempo de respuesta del esclavo del FSL ante datos presentes en su memoria FIFO interna. Lo mismo se puede concluir con el NI10 pues en este caso a la memoria FIFO le toma 1µs en recibir nuevos datos escritos sobre el maestro del FSL.

B. Análisis de Resultados en Estimación de Potencia

a) Análisis de Resultados en Estimación de Potencia Dinámica para las Arquitecturas Interconectadas por NoC Hermes y por Buses FSL

En la Figura 8 se compara el consumo total de potencia para cada una de las arquitecturas implementadas. Al realizar un análisis comparativo de estimación de potencia consumida por los elementos del FPGA, se determina que en la infraestructura conectada por NoC Hermes, el consumo de potencia es mayor a la infraestructura conectada por buses FSL en todos los escenarios de estimación. También se puede observar un incremento de potencia directamente proporcional al incremento de la tasa conmutación en cada uno de los escenarios de prueba tanto para la arquitectura interconectada

por NoC Hermes como para la arquitectura interconectada por Buses FSL.

Para determinar un modelo de estimación se utiliza la variación de la tasa de conmutación tanto para Flip Flops, como para elementos I/O. En la Tabla 8 se enlistan los escenarios de pruebas bajo los cuales se realizó la estimación de potencia tanto para una arquitectura interconectada por NoC Hermes, como para una arquitectura interconectada por buses FSL.

TABLA 8. ESCENARIOS DE ESTIMACIÓN DE POTENCIA

Tasa I/Os	Tasa Flip Flops	Escenarios de Prueba
12.5 %	12.5 %	Escenario 1
12.5 %	25 %	Escenario 2
25 %	12.5 %	Escenario 3
25 %	25 %	Escenario 4
50 %	50 %	Escenario 5
75 %	75 %	Escenario 6

Estimación de Potencia

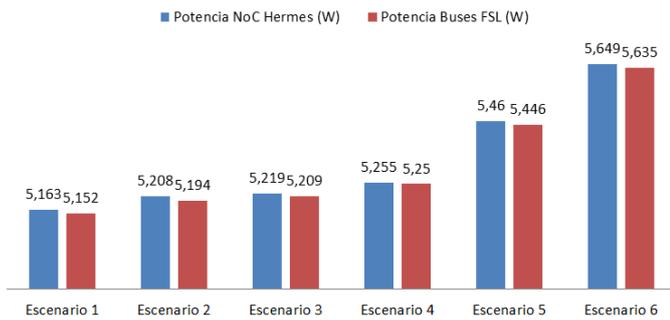


Figura 8. Análisis comparativo de estimación de potencia en función de los escenarios de estimación.

En conclusión el incremento en el consumo de potencia en la arquitectura interconectada por NoC Hermes se debe a la cantidad de elementos lógicos presentes en los Switches de Hermes y en las Interfaces de Red diseñadas para las interconexiones con los distintos procesadores. Estos elementos se obviaron en la arquitectura de buses y por esta razón, tanto instanciar, como configurar dichos elementos, ya conlleva un incremento de consumo de potencia.

b) Análisis de Resultados en Estimación de Potencia Estática para las Arquitecturas Interconectadas por NoC Hermes y por Buses FSL

En la Figura 9 se puede observar que a medida que se incrementa el consumo de potencia estática también se incrementa la temperatura de juntura. Es así que la arquitectura interconectada por NoC Hermes consume 33mW más que la arquitectura interconectada por buses FSL. En consecuencia la temperatura de juntura de la arquitectura interconectada por NoC Hermes es 0.1 °C más que la arquitectura interconectada por buses FSL. Este aumento en potencia y temperatura sobre la arquitectura interconectada por NoC Hermes se debe a la cantidad de recursos utilizados en el FPGA tal como se puede observar en la Figura 9.

Comparación de Temperatura de Juntura y Potencia Estática Consumida por el FPGA

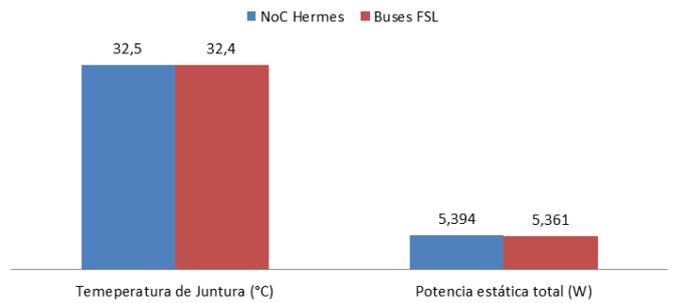


Figura 9. Comparación de temperatura de juntura y de potencia estática total.

En la Figura 10 se observa la comparación de recursos utilizados en el FPGA para la arquitectura interconectada por NOC Hermes y para la arquitectura interconectada por buses FSL.

Comparación de Recursos Utilizados en el FPGA

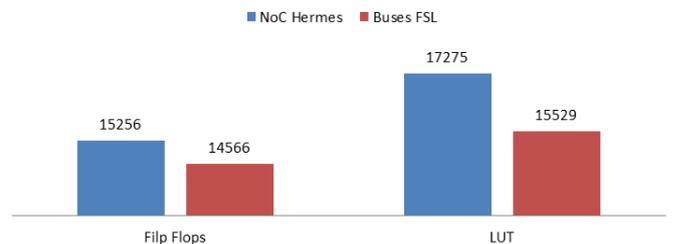


Figura 10. Comparación de los recursos utilizados en el FPGA Virtex-6.

En base a la herramienta Xilinx Power Estimator, la cual es una hoja de cálculo para evaluar potencia, se puede realizar un análisis comparativo de consumo de potencia estática para cada una de las arquitecturas, en donde se evalúan los valores de estabilización con respecto a la fuente de alimentación Vccint.

Potencia en NoC Hermes y en FSL Vs Vccint

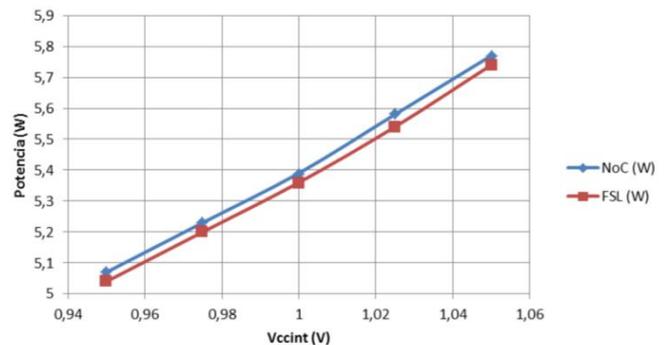


Figura 11. Potencia en NoC Hermes y en FSL Vs. Vccint.

En la Figura 11 se puede apreciar que la potencia es mayor en una arquitectura interconectada por NoC Hermes con respecto a una arquitectura interconectada por buses FSL en 32mW aproximadamente. Esto se debe a que ambas arquitecturas están conectadas en la misma topología de malla, siendo NoC Hermes la arquitectura de comunicación que más elementos emplea para su interconexión a nivel intrachip.

Finalmente, la Figura 12 muestra un análisis comparativo de Temperaturas de Juntura para la arquitectura conectada por NoC y para la arquitectura conectada por buses FSL, tomando en cuenta la estimación de potencia para cada uno de los diseños.

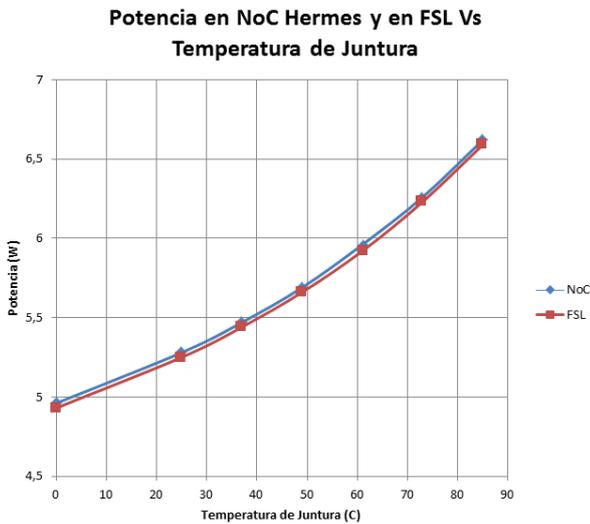


Figura 12. Potencia en NoC Hermes y en FSL Vs. Temperatura de Juntura.

Como se mencionó anteriormente ambas arquitecturas utilizan una topología de malla en su infraestructura de comunicación. Por lo tanto en la Figura 12 se puede apreciar que la disipación de potencia por temperatura de juntura para una arquitectura interconectada por NoC Hermes es mayor que para una arquitectura interconectada por buses FSL en 31mW aproximadamente. Sin embargo es posible que conforme se realicen trabajos futuros sobre topologías en donde se incrementen más dispositivos con buses FSL se vea un aumento en la disipación de potencia por temperatura de juntura.

VI. CONCLUSIONES

Se implementó dos arquitecturas MPSoC de 4 procesadores MicroBlaze sobre la tarjeta de desarrollo ML605 Virtex-6. La primera tiene una arquitectura de comunicación por medio de Buses FSL y la otra está comunicada por medio de NoC. Estas arquitecturas fueron comparadas con el fin de demostrar que NoC es más eficiente en latencia que buses FSL en la transmisión de paquetes para una aplicación que demande alto tráfico de datos y mínima pérdida de información.

Después de realizar una caracterización del estado del arte de NoC se encontró que existe una amplia variedad de modelos de switches interconectados en diferentes topologías a nivel intrachip. Gracias al aporte del Área de Sistemas Digitales de la Escuela Politécnica del Ejército (ESPE) se planteó la utilización de NoC Hermes desarrollada por la Facultad de Informática de la Pontificia Universidade do Rio Grande do Sul (PUCRS, Porto Alegre, Brasil) y la implementación sobre la FPGA mediante código VHDL a través de la herramienta ATLAS desarrollada por GAPH (Hardware Design Support Group-PUCRS). Esta herramienta configura únicamente los switches de NoC Hermes en una topología 2D-Mesh en código abierto VHDL.

Con la finalidad de crear una arquitectura multiprocesada basada en MicroBlaze que emplea NoC Hermes para su comunicación fue necesario diseñar y desarrollar una interfaz de red (NI). El IP-Core resultante es de código abierto VHDL, puede ser modificado y es exclusivamente para comunicación por FSL. Este IP-Core fue desarrollado en la ESPE como contribución al presente proyecto de grado.

A pesar de que se logró implementar las dos arquitecturas MPSoC interconectadas por buses e interconectadas por NoC, no se pudo medir físicamente la latencia y el consumo de energía debido a que la plataforma de desarrollo no permite realizarlas. Por este motivo se recurrió a la estimación de estos parámetros mediante el uso de simuladores. Las simulaciones se realizaron utilizando generadores de tráfico randómico que carecen de un patrón de comportamiento que emule las características de una arquitectura multiprocesada. Por lo tanto, lo ideal para trabajos futuros es que se desarrollen generadores de tráfico que permitan simular de forma paralela diversos números de nodos dentro de una topología NoC Hermes superior a una 2D-Mesh de 2x2. Por otra parte se deberán crear generadores de tráfico que permita evaluar a NoC Hermes con patrones de tráfico de aplicaciones reales.

Mediante la utilización de generadores de tráfico se realizaron simulaciones a dos y tres nodos encontrando como resultado que la arquitectura NoC Hermes aumenta su latencia a medida que se incrementa el número de nodos involucrados en la simulación. Esto se debe principalmente al tiempo de dispersión que le toma a cada switch para establecer una ruta de comunicación mediante el algoritmo XY.

En las pruebas simuladas entre NoC Hermes a dos nodos y buses FSL, en la transmisión de paquetes de punto a punto se demostró que el bus FSL es más rápido que la arquitectura NoC Hermes, debido a que los tiempos de transmisión son más prolongados en NoC Hermes que en FSL. En conclusión una arquitectura NoC Hermes de 2x2 en topología 2D-Mesh no es suficiente para ser evaluada. Por lo cual se requiere topologías más grandes en donde las latencias se incrementen a tal punto que se pueda demostrar la potencial mejora del desempeño al utilizar NoC Hermes frente a buses FSL.

En la arquitectura de comunicación NoC Hermes sin NI, conectada con generadores de tráfico, la latencia es de 23µs para un paquete de 5 flits. Mientras que para una arquitectura

de comunicación NoC Hermes con NI, la latencia es de 25µs para el mismo tamaño de paquete. En consecuencia los NI diseñados para comunicar a los procesadores con NoC Hermes presentan una latencia de 2µs adicionales.

Con el fin de evaluar las arquitecturas multiprocesadas interconectadas por buses FSL y por NoC Hermes implementadas sobre el FPGA se desarrolló una aplicación de esteganografía en la que existe multiprocesamiento de dos procesadores trabajando simultáneamente. Esta aplicación fue útil para evaluar ambas arquitecturas dando como resultado que la implementación mediante buses FSL ocupó el 10% de los recursos del FPGA y NoC Hermes el 11%. Mientras que para las dos arquitecturas se utilizó el 100% de los recursos de memoria disponibles en el FPGA.

Para medir latencia se necesita gran cantidad de tráfico de información pero por las limitaciones de memoria en la plataforma de desarrollo se utilizó imágenes de 320x240 píxeles y con mensajes de 300 caracteres. Las limitantes en dimensión de imágenes y en tamaño de mensajes están condicionadas por las capacidades de memoria del FPGA ML605 Virtex-6 sobre el cual fueron implementados.

Para comparar el consumo de energía de cada una de las arquitecturas multiprocesadas se realizó una estimación de potencia variando las tasas de conmutación tanto para Flip Flops como para elementos I/O. El proceso de estimación da como resultado que el consumo de energía en la arquitectura interconectada por NoC Hermes en topología 2D-Mesh es 33mW mayor que el consumo de energía para la arquitectura interconectada por buses FSL. Este incremento se produce debido a que la arquitectura interconectada por NoC Hermes posee más elementos que son necesarios para su funcionamiento, los mismos que no están presentes en la arquitectura interconectada por buses FSL. Estos elementos corresponden a los Switches de Hermes y las Interfaces de Red que permiten la comunicación con los procesadores MicroBlaze.

AGRADECIMIENTO

Agradecemos a los Autores de la Facultad de Informática PUCR por facilitar el uso de la NoC que desarrollaron y en especial agradecemos a la Escuela Politécnica del Ejército por permitirnos ser parte de este importante proyecto de investigación.

REFERENCIAS

[1] K.Rajesh , “On-chip memory architecture exploration of embedded Systems on chip”, Tesis Doctoral, Facultad de Ingeniería , Instituto de Ciencias de la India, Supercomputer Education and Research Centre, Sep. 2008.

[2] L. Benini, G. de Micheli, “Network-on-Chip: A new paradigm for component-based MPSoC desing”, DEIS Universita di Bologna Italia, CSL Stanford University Stanford CA.

[3] T. Tao Ye, “On-chip Multiprocessor Communication Network design and analysis ”, Stanford University , December 2003.

[4] L. Torres, P. Benoit, G. Sassatelli, M. Robert, “An introduction to Multi-Core Systems on Chip – Trends and challenges”, University of Montpellier 2, LIRMM UMR CNRS, France.

[5] A. Schranzhofer, J. Chen, L. Thiele, “Power-Aware Mapping of Probabilistic Applications onto Heterogeneous MPSoC Platforms ”, Computer Engineering and Networks Laboratory (TIK), Zurich Switzerland.

[6] W. Wolf, A. Amine, G. Martin, “Multiprocessor System-On-Chip (MPSoC) Technology”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, No 10, October. 2008.

[7] “Interfaz WISHBONE”, Revisión 1.6, OpenCores, 2010.

[8] R. Ernst, “MPSOC Architecture Modeling”, TU Braunschweig, 2003.

[9] M. Milica, M. Stojcev, “An Overview of On-Chip Buses”, Facta Universitatis (NIS), Vol 19, December 2006.

[10] M. Aparicio, “Diseño de un sistema de lectura y procesado para múltiples sensores embebidos en un FPGA”, Proyecto de Fin de Carrera, Universidad Autónoma de Madrid, Diciembre. 2007.

[11] J. Peña, “Diseño de módulos para el manejo de Puertos, Temporización e Interrupciones para el Núcleo KCPSM3 e Implementación en el FPGA XC3S500 de Xilinx”, Proyecto de Fin de Carrera, Universidad Tecnológica de la Mixteca, Junio. 2008.

[12] S. Pantelopoulos, A. Brokalakis, “High Level Architecture of Parallel System”, Information and Communication Technologies (ICT) Programme, May 2012.

[13] S. Hessabi, “SoC Design”, Department of Computer Engineering, Sharif University of Technology.

[14] Xilinx, “LogiCore IP Multi-Port Memory Controller (MPMC) (v6.04.a)”, EDK, DS643 July 6, 2011.

[15] Xilinx, “XPS Thin Film Transistor (TFT) Controller (v2.01a)”, EDK, DS695 May 3, 2010.

[16] Xilinx, “XPS SYSACE (System ACE) Interface Controller (v1.01a)”, EDK, DS583 March 4, 2010.

[17] Xilinx, “EDK Concepts, Tools and Techinques”, UG683 EDK 12.2.

[18] Xilinx, “MicroBlaze Debug Module (MDM) (v2.00b)”, EDK, DS641 March 1, 2011.

[19] F. Moraes, N. Calazans, A. de Mello, L. Möller, L. Copello Ost, “HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip”, TECHNICAL REPORT SERIES, Facultad de Informática PUCRS-Brazil, Number 034, October, 2003.

[20] Xilinx, “LogiCORE IP Fast Simplex Link (FSL) V20 Bus (v2.11e)”, EDK, DS449, June 2011.

[21] D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, G. de Micheli, “Network-on-Chip design and synthesis outlook”, INTEGRATION, the VLSI journal 41 (2008) 340–359, Received 26 November 2007

[22] Xilinx, “Embedded System Tools Referce Manual”, EDK 13.2, UG111 July 6. 2011.

[23] M. Death, “Introducción a la Esteganografía”, Death Master, 2004 (GFDL), <http://www.death-master.tk/>.

[24] “Procesamiento de Imágenes”, Universidad Simón Bolívar, Departamento de Comunicación y TI, Abril – Julio 2008.

[25] “Procesamiento de Imágenes Digitales”, Universidad de Málaga, Dpto. Lenguajes y CC. Computación E.T.S.I. Telecomunicación.

[26] P. Juan, “Proyecto Encriptación de texto sobre imágenes BMP mediante software”.

Dispositivo para medición de la presión arterial sobre la plataforma Android para dispositivos móviles

M. Vinueza, D. Alulema, F. Pineda

Resumen.- En este proyecto se presenta un dispositivo para la medición de la presión arterial implementado sobre la plataforma Android para su funcionamiento en smartphones y tablets. La implementación consta de software y hardware, la forma de medición del dispositivo se basa en la Técnica Oscilométrica. El software toma en cuenta la interfaz gráfica de usuario, los algoritmos de programación y la lógica de funcionamiento del módulo, todo esto ejecutado en JAVA como lenguaje de Programación. El hardware se compone de una microbomba, una electroválvula y como tarjeta de control y adquisición de datos la IOIO de Android; estos elementos en conjunto permiten la automatización de la medición de la presión arterial. El dispositivo proporciona al paciente una medición de su presión arterial (sistólica, diastólica y media); además, permite tener un registro que se almacena en su dispositivo móvil.

I.-INTRODUCCIÓN

Según estudios realizados por la sociedad Internacional de Hipertensión (SIH), cerca del 80% de muertes debido a esta enfermedad se producen en países subdesarrollados o en vías de desarrollo, ocupando Ecuador el primer lugar en Latinoamérica, llegando a cifras desconcertantes, pues el 46% de la población Ecuatoriana de edad adulta sufre de hipertensión y tan solo el 15% de dicha población sabe y tiene un control adecuado, el resto por no tratarla sufre de enfermedades cardiovasculares en edad más avanzada.

Este signo vital es muy importante pues determina la fuerza con la que el corazón está impulsando la sangre hacia todo el cuerpo, tanto subida como bajada de los valores normales, 120/80, pueden causar daños cardiovasculares.

En la actualidad existen varios métodos para la medición como lo son: directos e indirectos.

Los directos son los más precisos pues se hace una incisión en el brazo con un catéter y mediante un transductor se obtiene la presión del paciente.

El método indirecto no es tan exacto como el primero pero no es invasivo con el cuerpo humano, es por esto que dentro de este método existen técnicas como:

Auscultatoria.- Se basa en escuchar los sonidos de korokoff o sonidos que permiten identificar la presión sistólica y diastólica a través de un estetoscopio.

Oscilométrica.- Se registra las variaciones de presión mediante un transductor cuando la vena deja de estar ocluida.

Ultrasónica.- Se mide la diferencia de frecuencia existente entre un emisor y receptor ubicado a—un extremo y a otro respectivamente de una arteria.

Color del Capilar.- Estima el valor de la presión basándose en la coloración de la piel.

Todas estas técnicas a excepción de la ultrasónica hacen uso de un brazalete (bolsa de aire con dos orificios que abraza el brazo o muñeca) para lograr la oclusión de la vena y permitir la medición de la presión arterial.

En esta clasificación la técnica prevaleciente en los dispositivos automáticos es la oscilométrica, que basa su funcionamiento en la detección de las variaciones de presión que se originan en el brazalete.

II.- DISEÑO

Para llevar a cabo la medición se infla un brazalete en la muñeca del paciente llegando a una presión aproximadamente de 200mmHg en donde se asegura la oclusión de la arteria, se ubica el sensor de presión, al llegar a este límite se deja escapar el aire lentamente para poder captar las diferencias de presión existentes en el interior del brazalete o mango hasta un valor aproximado de 20mmHg, el cual no se encuentra dentro del rango de presiones en las personas.

La figura 1 muestra el descenso de la presión arterial desde los 180mmHg hasta 10mmHg con un número de muestras de 2500, es decir cuando se deja escapar el aire del brazalete lentamente para el registro de los valores.

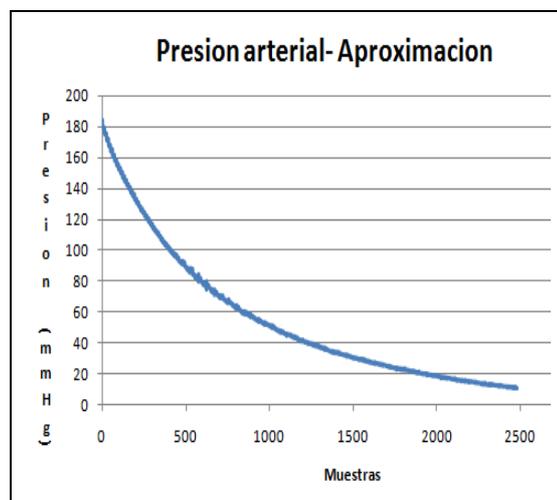


Figura 1. Presión arterial-desinflado

Un acercamiento de la figura 1 muestra claramente las variaciones que se producen en el interior. Es vital determinar la frecuencia de la señal, los pulsos normales para una persona adulta varían en el rango de 60-180 pulsos por minuto. Transformado al dominio de la frecuencia de 1hz a 3hz, por teorema del muestreo la frecuencia de lectura debe ser al menos el doble de la frecuencia de la señal, sin embargo para asegurar la correcta lectura de los datos se toma una frecuencia de muestreo de 22 veces.

$$f_m = 22 * 3 = 66\text{hz} \quad (1)$$

Teniendo esta frecuencia se puede determinar la velocidad que debe darse entre cada muestra o el periodo de muestreo.

$$T = \frac{1}{f_s} = \frac{1}{66\text{hz}} = 0.015\text{s} \quad (2)$$

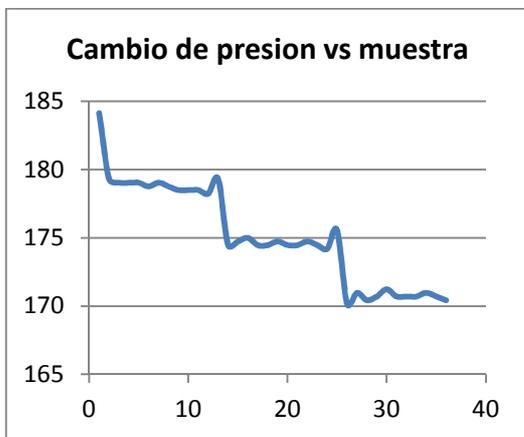


Figura 2. Presión arterial-desinflado

Implementada la consideración para el muestreo se debe linealizar la señal para evitar errores de sobrepicos que puedan alterar la identificación de la presión media sistólica y diastólica, para esto se hace un algoritmo de ordenamiento. El algoritmo toma 12 muestras, reordenarlas de mayor a menor y las almacena en un nuevo arreglo de muestras, porque cada 12 muestras la electroválvula se activa por un periodo de 15ms, tiempo también que se da entre muestras; entonces realizando el algoritmo se logra eliminar los sobre picos debido al tiempo que se activa la electroválvula y se concentra la atención en los datos que se produzcan por el desinflado del brazalete.

Las variaciones de presión se hallan como:

$$VP = \text{Muestra Real} - \text{Muestra Aproximada} \quad (3)$$

Donde V_p = Variación de Presión.

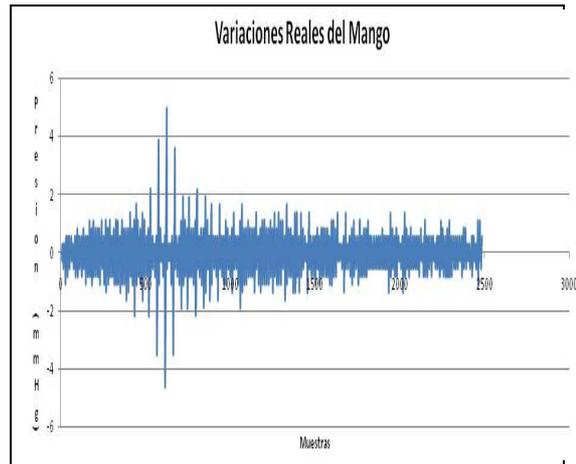


Figura 3. Respuesta aplicando oscilometría

La figura 3 muestra las variaciones producidas donde se identifica de manera precisa la presión media y las presiones a su alrededor. Mediante la utilización del método de las alturas se aproxima la presión sistólica y diastólica.

El método consta en identificar la presión media o de pico más alto de variación y asignar constantes para la identificación de la presión sistólica y diastólica.

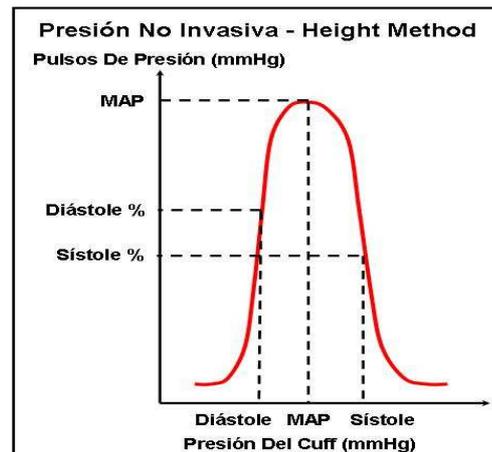


Figura 4. Método de las alturas [2]

Con las pruebas realizadas en pacientes se estableció un valor de 0,6 para la presión sistólica haciendo un barrido de izquierda a derecha de la gráfica de la figura 3 y un valor de 0.8 para el valor de la presión diastólica de derecha a izquierda.

Teniendo las siguientes fórmulas en el algoritmo de identificación.

$$PM = \text{Valor máximo}$$

$$PS = 0,6 * PM$$

$$PD = 0,8 * PM$$

Cabe mencionar que no son valores exactos de PS y PD, es decir no se calcula multiplicando por las constantes de la formula solo que son el patrón para la

aproximación haciendo el barrido de la señal, la muestra donde se encuentre un valor superior o igual a este, será la presión respectiva del paciente.

III.- IMPLEMENTACION

Para la implementación del Dispositivo se tienen los siguientes elementos, que permiten leer las variaciones, directamente de la arteria radial:

- Sensor de Presión (MPX5050DP) fabricado por Motorola con un rango de operación entre 0a 300mmHg.
- Bomba de Aire (Micro bomba para el inflado del brazalete y capaz de llegar a 250mmHg de presión)
- Mangueras (de ¼ de pulgada para su acoplamiento a sensor y bomba)
- Tarjeta IOIO
- Dispositivo móvil(con sistema operativo Android versión mínima 2.0)
- Electroválvula(solenoides)
- Brazalete (de 13,5 a 19,5 cm de diámetro para la muñeca estándar para una persona adulta)

El proceso para medir se lo realiza secuencialmente:

- Recepción de datos
- Linealización de la curva característica
- Cálculo de los valores puntuales de presión diastólica, sistólica y media.



Figura 5. Dispositivo de Medición ensamblado

Es importante la ubicación de cada elemento, el sensor de presión debe estar justo debajo de la arteria radial, la microbomba y electroválvula deben ubicarse lo más cerca posible de el brazalete para no tener pérdidas por la diferencia de nivel, los demás elementos deben acoplarse de manera precisa para que el volumen del módulo sea compacto.

La comunicación entre los elementos es diferente de unos a otros, la tarjeta IOIO con la microbomba, el Smartphone y la electroválvula son del tipo eléctrico, por dar las señales de control y alimentación, adicionalmente con el sensor la comunicación también es eléctrica, para la lectura de voltaje proporcional a la presión en el mango. La parte neumática entra en la

comunicación de la microbomba-brazalete-electroválvula y sensor pues mediante un ducto el aire va a pasar por toda esta ruta, la microbomba infla el brazalete, el sensor mide la presión, en ese instante, la electroválvula deja escapar el aire paulatinamente.

Cada uno de los elementos posee características técnicas acordes para la comunicación entre ellos, la tarjeta IOIO es capaz de bridar 2A con un voltaje de 3.3V de alimentación con lo que es capaz de activar la electroválvula de consumo de 1.5A y también de activar la microbomba de 1A en diferentes tiempos por la secuencia del proceso. Debido a esto la batería de polímero de litio es capaz de proporcionar la energía suficiente para realizar al menos 32 mediciones seguidas.

$$tiempodedescarga = \frac{cargaelectricadelabateria}{consumoelectricodeldispositivo} \quad (4)$$

$$tiempodedescarga = \frac{800mAh}{1500mA} = 0.53h = 32min$$

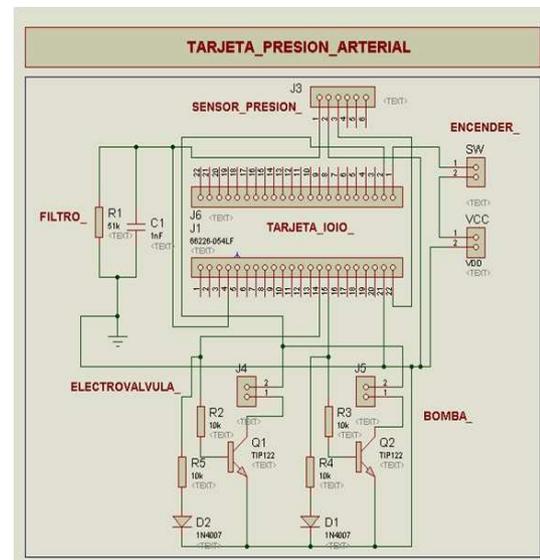


Figura 6. Tarjeta Electrónica

Es necesario una etapa de control neumático y eléctrico para automatizar el proceso, por lo que un diagrama P&ID del sistema muestra las señales, el controlador y las conexiones entre cada uno de los componentes que forman parte activa del sistema.

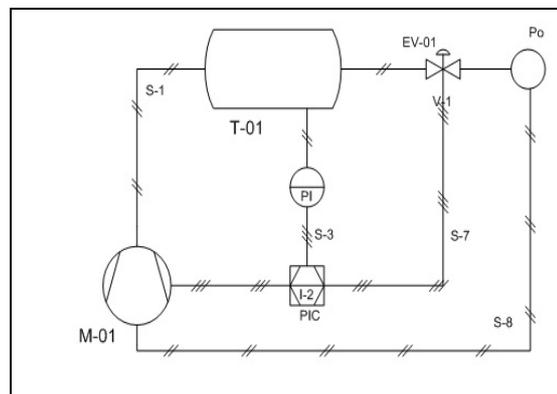


Figura 7. Diagrama PID

- T-01 → Brazaletes
- M-01 → Micro-bomba
- EV-01 → Electroválvula
- PIC → Controlador indicador de Presión (tarjeta IOIO)
- PI → Indicador de Presión
- Po → Presión Atmosférica.

Todos los elementos se unen en un lazo cerrado, la micro-bomba se une con el brazaletes, este a su vez con el sensor y finalmente con la electroválvula; el primero ingresando aire y el último expulsando aire del sistema. En la parte eléctrica la tarjeta proporciona alimentación a los actuadores y señales de control para los mismos.

Para la aplicación en Android primero se ha establecido el rol del usuario en el sistema.

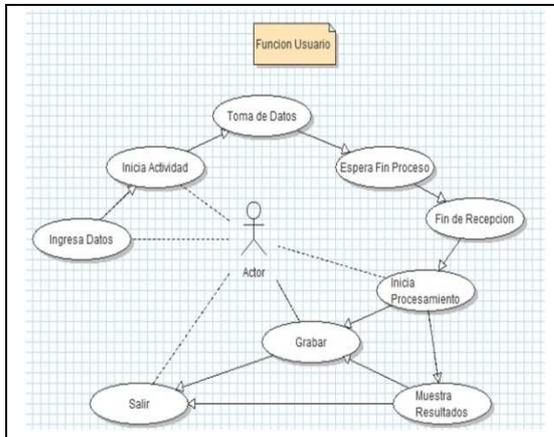


Figura 8. Función del usuario

La aplicación tiene varios pasos secuenciales para el uso del paciente, primero se ingresa los datos del usuario: nombre y edad. Después de esto se inicia un proceso dentro de la actividad donde se toman los datos de la presión, se ejecuta automáticamente un análisis de los datos, luego se muestran al paciente y tiene la opción de volver a realizar la medición o de guardar y salir de la aplicación.

Los algoritmos son implementados en software, tanto para el desarrollo de linealización y cálculo de aproximación de las presiones.

Para la ejecución se manejan dos archivos principales:

- ❖ Main.java
- ❖ Main.xml

La primera (Main.java) crea una subclase de IOIO Abstract, que implementa el IOIO framework, detrás de las escenas, donde yace la lógica de programación. En la segunda (Main.xml) Java es notificado del diseño de la interfaz del usuario, concurrentemente están todas las etiquetas del diseño de la interfaz. Ambos tienen una interrelación, mostrando el componente gráficamente y como se verá en la

realidad y el otro en forma de código, dando el diseño de la aplicación respectivamente.

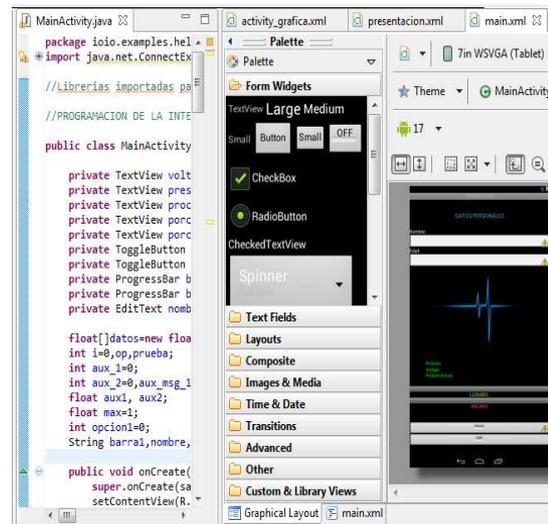


Figura 9. Interrelación de elementos

Por otra parte, el archivo.java maneja dos clases principales para el proceso de la aplicación, estos son:

MainActivity → Aquí se definen las variables de la interfaz gráfica del usuario, proporcionando un nombre a cada objeto del que se forma la pantalla. También se crean las variables generales del sistema, y se hereda de la clase principal IOIOActivity el método onCreate.

Looper → Es la clase donde se implanta todas las variables propias de la tarjeta como son: entradas análogas, digitales, y por la herencia de la clase BaseIOIOLooper tiene métodos para lectura de voltaje asignación a pines, señales pwm y acceso a diferentes mecanismos para control de la tarjeta

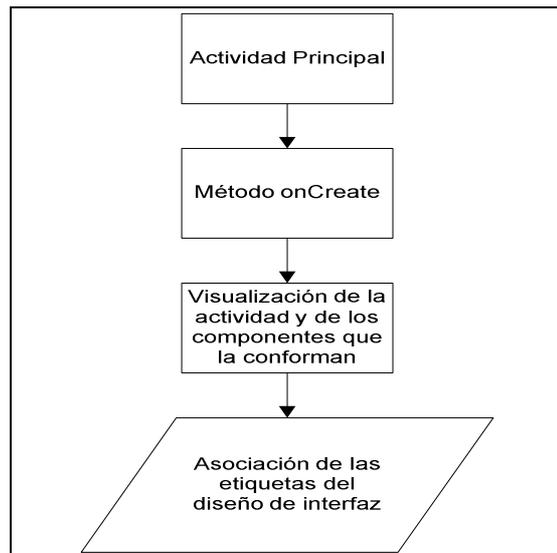


Figura 60. Ciclo de inicialización

Cuando se crea una nueva actividad en Android se inicializa el esquema de la figura 11 y cuando ya se está ejecutando la figura 12.

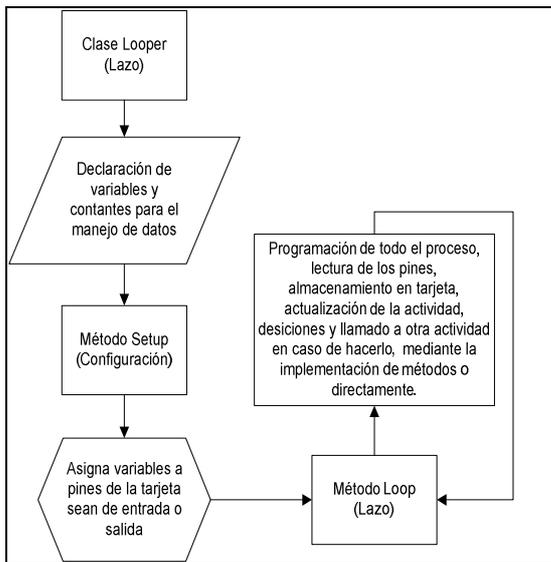


Figura 71. Ciclo de ejecución

Este ciclo es repetitivo para que según interacciones con el usuario el programa sepa que sentencias debe seguir.

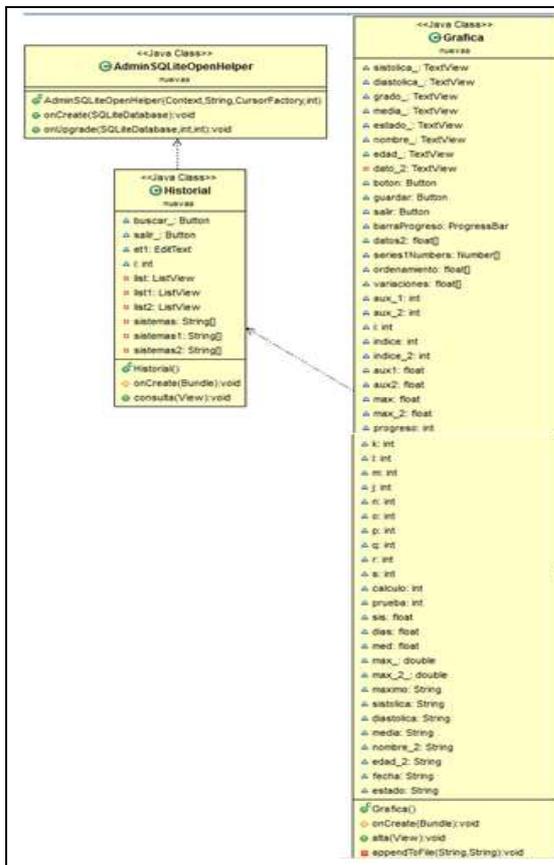


Figura 12. Diagrama UML parte 1

El diagrama UML(Lenguaje Unificado de modelado) usado para el diseño de sistemas hace uso de algunas formas predefinidas para el desarrollo; identificación de clases diagrama de secuencia de estados y otros esquemas que permiten una mejor lectura de un programa y su funcionamiento. Las clases usadas en la aplicación son:

- **Clase principal** → (Actividad principal) es una clase .java donde se implementan todas las sentencias de ejecución para el proceso de la medición desde interacciones para actualización a la pantalla como llamados a otras clases para el desarrollo de la aplicación.
- **Clase presentación** → Es una clase. Java la cual tiene asociada un archivo xml donde se tiene la portada o pantalla de inicialización de la aplicación.

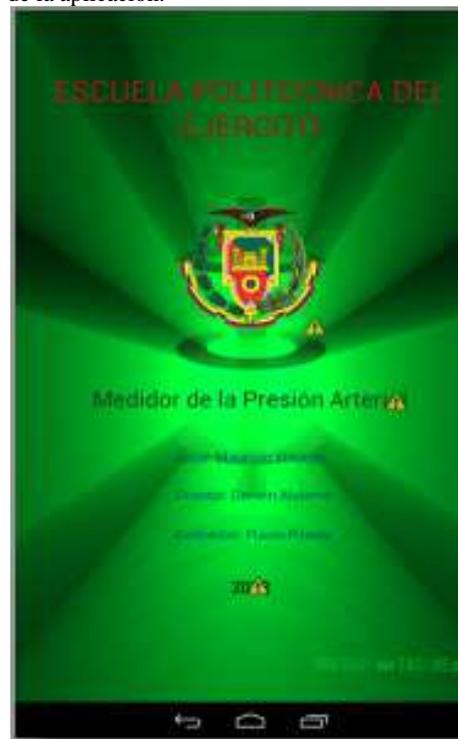


Figura 83. XML de la clase presentación

- **Clase looper** → Esta clase es propia de la tarjeta IOIO aquí se implementa todo el código que realizara la tarjeta de manera repetitiva y con ejecuciones reales hacia los pines de la tarjeta.
- **Clase tarea en segundo plano** → aquí se realizan sentencias que procesan información pero en segundo plano para no alterar la interacción con el usuario.
- **Clase historial** → la funcionalidad de esta clase radica en el almacenamiento de valores dentro de una base datos.

- **Clase AdmSQLServer** → Es la clase que implementa métodos para la utilización de SQL en Android permitiendo tener esta unión y además de proveer métodos para el uso de la base de datos.

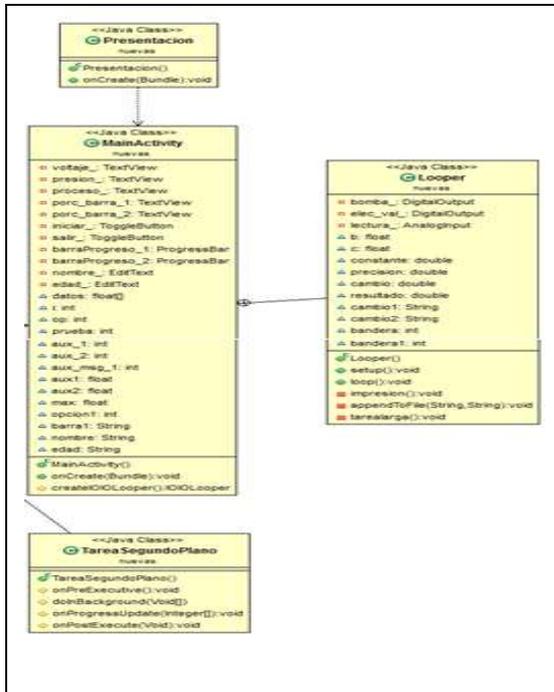


Figura 94. Diagrama UML parte 2

En cuanto a la interfaz gráfica del usuario se muestran elementos importantes durante el proceso de medición como etapa del proceso, lectura del voltaje, la presión actual del paciente y porcentaje de llenado y vaciado del brazalete.

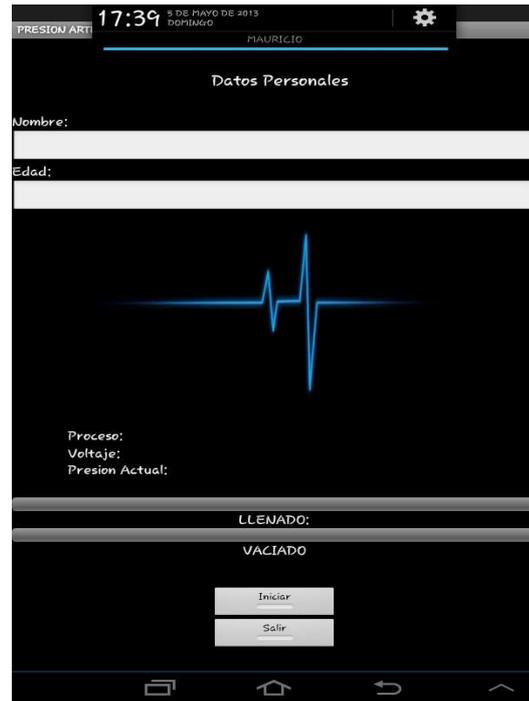


Figura 105. Interfaz Grafica del usuario

IV.- ANALISIS DE RESULTADOS

Dentro de las pruebas generales para cumplir con los requerimientos de diseño del dispositivo se encuentran:

- Capacidad para no reducir la presión de un 1mmHg en 10s cuando se encuentra inflado.
- Capacidad de cargarse y estar en funcionamiento durante al menos 16 mediciones consecutivas
- Aproximación para el cálculo de la presión sistólica y diastólica además de su correcta identificación de la presión media.

El proceso de la medición consiste en ubicar el dispositivo en la muñeca ajustando el sujetador lo suficiente para apretar un poco como lo muestra la figura 17.

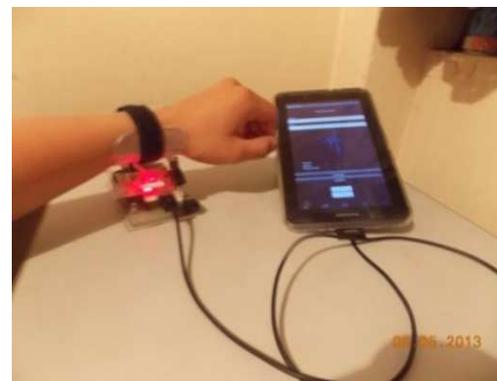


Figura 116. Colocación del Dispositivo

Ya que el dispositivo de medición es automático y controlado, simplemente se da click en el botón iniciar

y esto hará que el sistema comience la operación mostrando al final del proceso la siguiente gráfica con resultados.



Figura 127. Resultados-Medicación

La gráfica 18 muestra los resultados obtenidos en base a un registro de los datos de presión de 10 personas adultas a quienes se les realizó la medición de su presión arterial con 3 diferentes dispositivos de medición, el primero el prototipo, dispositivo comercial electrónico y dispositivo manual.

Paciente	Dispositivo	P. Sistólica	P. Diastólica	P. Media	Error Sis.	Error Dijas.
1	Prototipo	115	72	86	3	8
	Electrónico	128	85	89	10	5
	Manual	118	80			
2	Prototipo	92	62	72	8	8
	Electrónico	98	78	85	2	8
	Manual	100	70			
3	Prototipo	123	77	92	3	1
	Electrónico	132	75	90	12	3
	Manual	120	78			
4	Prototipo	124	71	89	4	3
	Electrónico	118	86	100	2	18
	Manual	120	68			
5	Prototipo	100	77	90	2	2
	Electrónico	101	72	92	3	3
	Manual	98	75			
6	Prototipo	125	73	92	2	5
	Electrónico	119	86	95	4	8
	Manual	123	78			
7	Prototipo	132	73	94	5	4
	Electrónico	132	86	100	5	9
	Manual	127	77			
8	Prototipo	148	78	112	8	2
	Electrónico	147	72	119	7	8
	Manual	140	80			
9	Prototipo	122	89	100	2	4
	Electrónico	123	72	102	3	13
	Manual	120	85			
10	Prototipo	117	58	85	3	4
	Electrónico	110	60	90	10	2
	Manual	120	62			
				Error Prototipo.	4	4,1
				Error Electrónico.	5,8	7,7

Figura 138. Tabla de resultados en Pacientes

Se puede observar las mediciones en los tres equipos, con el error respectivo del prototipo diseñado y del

medidor electrónico; ambos respecto al esfigmomanómetro manual.

Realizando una comparación en cada individuo de las dos presiones importantes sistólica y diastólica se puede obtener la gráfica de la figura 19

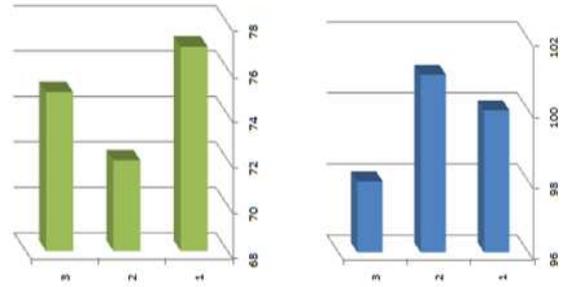


Figura 19. Presión sistólica- Diastólica (Individuo 1)

Los números en la Figura 19 representan los equipos:

#1 → prototipo



Figura 140. Prototipo basado en Android

#2 → Dispositivo electrónico



Figura 151. Dispositivo Electrónico

#3 → Esfigmomanómetro manual



Figura 162. Esfigmomanómetro

Los resultados en cada medición son obviamente diferentes respecto al módulo con el que se mide y al individuo que se le toma la presión, sin embargo sacando un error promedio se puede estimar el error de cada uno de los dispositivos.

Error Prototipo.	4	4,1
Error Electrónico.	5,8	7,7

Tabla 1. Resumen de errores

Se comparó el prototipo implementado con dispositivos electrónicos similares con la misma funcionalidad y existentes en el mercado y a su vez con la medición manual de la presión arterial, donde se observa que de manera eficiente el dispositivo tiene un margen de error de 4mmHg admisible en todos los dispositivos electrónicos y cumpliendo estándares como la norma EN1060-1 a 4 y AAMI (normativas que estandarizan que un medidor de presión arterial electrónico no debe tener fugas en 1mmHg durante al menos 10 segundos y que su error de precisión no debe exceder los 3 o 4mmhg) para dispositivos biomédicos a nivel mundial.

V.- CONCLUSIONES

El método más usado al nivel electrónico es el oscilométrico por la facilidad que presenta al calcular la presión media.

Para encontrar e identificar de manera correcta y obtener una buena aproximación, es mediante el análisis por etapas, es decir de 12 muestras en 12, su funcionalidad para este caso se da porque en cada paquete de muestras se encuentran las variaciones que suceden posteriormente se unen todos los paquetes para obtener la curva de pulso de índice oscilométrico, que constituye las variaciones en todo el periodo de desinflado, pero mejor reguladas por la aplicación del algoritmo en cada división.

La mejor tarjeta para la adquisición y procesamiento de los datos es la tarjeta IOIO por proveer alimentación de 3.3v con una corriente de hasta 1.5A regulables para activación de electroválvula y microbomba, otra característica fue el tamaño y la resolución de los pines de entrada analoga.

El diseño del algoritmo mostró una mejora sustancial en todo el procesamiento de la señal de la presión debido a que no usa amplificadores, filtros digitales,

aproximaciones por interpolación lineal o eliminación de un cierto rango de muestras por la variación que se produce en el principio de la toma de datos por la velocidad de salida del aire, con las pruebas realizadas se mostró la eficiencia del algoritmo para luego estimar las constantes de porcentajes de las alturas para calcular presión sistólica y diastólica en un valor de 0.55 y 0.7 respectivamente consiguiéndose un error de 4 mmHg con respecto al tensiómetro manual.

En las pruebas de medición de la presión arterial se debe acercar el dispositivo a la altura del corazón para evitar errores por la diferencia de presión que se da por el nivel de referencia.

Las importaciones tanto de librerías externas al paquete de instalación como para la operación de la tarjeta IOIO debe comprobarse la versión pues en anteriores no se incluye comunicación o permisos para la utilización de servicios como GPS, bluetooth, etc.

Hay que tener cuidado con la versión de Android del dispositivo móvil y la versión de SDK con la que se programa en la tarjeta pues si la primera es inferior entonces el programa no reconocerá órdenes del Smartphone.

Es importante no interrumpir en ningún momento el hilo principal de la actividad pues esto provoca una excepción en el programa, para esto se usan herramientas como tareas en segundo plano, servicios, notificaciones para adquirir datos o actualizar la interfaz.

REFERENCIAS

1. (s.f.). Recuperado el 27 de Marzo de 2013, de <http://osl.ulpgc.es/files/docs/cursos/Android/D1-2.pdf>
2. (2)Biomedico, P. (14 de Diciembre de 2010). Recuperado el 30 de Enero de 2013, de <http://portalbiomedico.com/equipamiento-biomedico/presion-no-invasiva-pni/-nibp/presion-no-invasiva-pni/nibp-conceptos-basicos.html>
3. (2012). (Medicina Colombia) Recuperado el 30 de Enero de 2013, de tensiómetro virtual: http://www.tensiovirtual.com/upload/81-001_a.pdf
4. Zambrana, A. C. (2004). Medicion de la tension Arterial. Valencia.
5. Geosalud. (2008). Geosalus.com. Recuperado el 30 de Enero de 2013, de <http://geosalud.com/hipertension/tecnicas.htm>
6. EFN. (s.f.). *efn.uncor.edu*. Recuperado el 30 de Enero de 2013, de

http://www.efn.uncor.edu/escuelas/biomedica/Plandeestudios/materias%20completas/Instrumentacion%20biomedica/Material/ago%202010/instrumentaci%F3n_presi%F3n_TEORICO.pdf

7. Electronica, R. (2007). Hipertensión y hipotensión arterial. Buenos Aires.
8. Espinoza, C. (6 de noviembre de 2012). *About.com*. Recuperado el 13 de Febrero de 2013, de Enfermedades del corazon: <http://enfermedadescorazon.about.com/od/nociones-basicas/a/El-Sistema-Circulatorio-Y-El-Corazon.htm>
9. Ferato.com. (2012). *Ferato*. Recuperado el 30 de enero de 2013, de http://www.ferato.com/wiki/index.php/Presi%C3%B3n_Arterial
10. Flores, D. A. (2009). *Clinica Homeopatia*. Recuperado el 30 de Enero de 2013, de https://homeopatiaflores.com/index.php?option=com_content&view=article&id=156&Itemid=152
11. Fluidos. (s.f.). Recuperado el 13 de Febrero de 2013, de <http://fluidos.eia.edu.co/hidraulica/articulos/flujoentuberias/flujosanguineo/flujosanguineo.html>
12. Galindo, D. M. (s.f.). *Monitoreo Transoperatorio basico*. Recuperado el 13 de Febrero de 2013, de http://www.anestesiaenmexico.org/RAM_95/RAM6/art/art_especial/art1.htm

Monitorización de Signos Vitales Mediante una Red de Dispositivos Móviles

Daniel Cilio, Katherine Herrera, Enrique V. Carrera

Resumen—El desarrollo e implementación de diferentes proyectos tecnológicos, apoyados en el correspondiente conocimiento médico, pueden contribuir a resolver varios problemas del sector de la salud. Si bien en los últimos años se han realizados enormes esfuerzos para desarrollar tecnologías aplicables en ambientes clínicos, el desarrollo de tecnologías para atención médica domiciliar podría reducir la presión que agobia a los hospitales actualmente. En el presente proyecto se realiza el diseño e implementación de un sistema para monitorización de signos vitales, el cual mide la frecuencia cardíaca, la oxigenación sanguínea y la temperatura corporal de una persona. La información obtenida de cada signo vital es muestreada y procesada por una plataforma digital para posteriormente ser enviada mediante un módulo Bluetooth hacia un dispositivo móvil para su análisis y visualización. El prototipo fue evaluado mediante una batería de pruebas para medición de signos vitales en diferentes pacientes.

I. INTRODUCCIÓN

El número de personas mayores de 65 años se duplicaría entre 1997 (357 millones) y 2025 (761 millones). Igualmente, el promedio de vida de las personas con enfermedades crónicas como problemas del corazón, cáncer y Alzheimer está aumentando. Esta situación contribuye a generar una elevada demanda por sistemas de asistencia médica, no solamente en términos de cuidados hospitalarios, sino de monitorización y cuidados rutinarios a gran escala [1].

El desarrollo e implementación de diferentes proyectos tecnológicos, apoyados en el correspondiente conocimiento médico, pueden contribuir a resolver varios de estos problemas en el sector de la salud. En particular, la utilización de dispositivos móviles y sistemas de comunicación inalámbricos puede mejorar el diálogo continuo entre pacientes y el personal médico. Esto permitiría brindar servicios de salud de mejor calidad y más productivos, eliminando el tiempo que un paciente pierde desplazándose desde su hogar hacia una estación particular de control médico.

Dentro de los procedimientos rutinarios realizados en pacientes está la monitorización de sus signos vitales. Esta monitorización puede realizarse aplicando un procesamiento de las señales biomédicas en una zona específica del cuerpo humano. Dentro de las señales biomédicas más importantes se destacan las señales bio-eléctricas y las señales bio-ópticas, las mismas que permiten hacer mediciones como

D. Cilio es ingeniero en Electrónica y Telecomunicaciones por la Universidad de las Fuerzas Armadas – ESPE, Sangolquí, Ecuador dacilio@espe.edu.ec

K. Herrera es ingeniera en Electrónica y Telecomunicaciones por la Universidad de las Fuerzas Armadas – ESPE, Sangolquí, Ecuador kvherrera@espe.edu.ec

E. V. Carrera es profesor del Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas – ESPE, Sangolquí, Ecuador evcarrera@espe.edu.ec

la del electrocardiograma (ECG) y oxigenación en la sangre (SpO_2), respectivamente.

En este proyecto se presenta el diseño e implementación de un sistema de monitorización de signos vitales con características flexibles tanto en su hardware como en su software, permitiendo una fácil lectura de las señales biomédicas más importantes. Si bien el software está orientado al sistema operativo Android en la actual implementación, el código fuente puede reutilizarse para adaptar la aplicación a diferentes plataformas. Adicionalmente, una estrategia de diseño es limitar el tamaño físico del hardware para obtener dispositivos de monitorización compactos que mejoren la movilidad de los pacientes. Finalmente, toda la plataforma de monitorización fue evaluada mediante una batería de pruebas para medición de signos vitales en diferentes pacientes y bajo distintas circunstancias.

II. FUNDAMENTO TEÓRICO

Los signos vitales son señales o reacciones que se perciben y miden rutinariamente por los profesionales de la salud para evaluar las funciones básicas del organismo. Los signos vitales constituyen una herramienta valiosa como indicadores del estado funcional del paciente. Una interpretación adecuada y oportuna de estos indicadores ayuda a los profesionales de la salud a decidir posibles tratamientos. En particular, la determinación de los signos vitales tiene extrema importancia en los servicios de emergencia, donde los pacientes presentan una gran variedad de cuadros clínicos.

Los signos vitales generalmente cambian con la edad, el sexo, el peso, la tolerancia al ejercicio y las enfermedades [2]. Los rangos normales de los signos vitales para un adulto sano promedio mientras está en reposo son:

- Temperatura: 37°C (98,6°F).
- Pulso: 60 a 100 latidos por minuto.
- Presión arterial: 90/60 mm/Hg hasta 120/80 mm/Hg.
- Respiración: 12 a 18 respiraciones por minuto.

A. El Electrocardiograma

El ECG es el registro gráfico, en función del tiempo, de las variaciones de potencial eléctrico generadas por el conjunto de células cardíacas y recogidas en la superficie corporal. Las variaciones de potencial eléctrico durante el ciclo cardíaco producen las ondas características del ECG haciendo posible su trazado característico. Estas variaciones son detectadas por electrodos ubicados en la piel. Mediante el electrocardiógrafo son amplificadas, filtradas y registradas en papel las forma de ondas y deflexiones que representan la magnitud y dirección de la actividad eléctrica cardíaca [3].

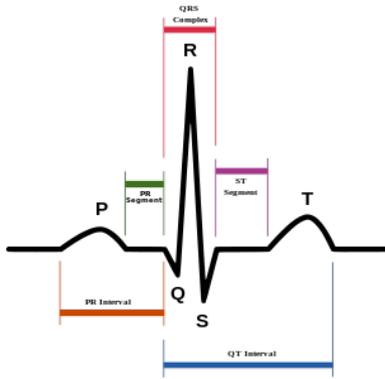


Fig. 1. Gráfico de un ECG con etiquetas de onda e intervalos.

Los electrodos se colocan sobre el cuerpo del paciente, sujetándolos y conectados al electrocardiógrafo mediante cables. Las derivaciones de un ECG utilizan diferentes combinaciones de electrodos para medir distintas señales procedentes del corazón. En forma figurada, cada derivación es como una ‘fotografía’ de la actividad eléctrica del corazón, tomada desde un ángulo diferente. En electrocardiografía, el término ‘derivación’ se refiere a la medida del voltaje entre dos electrodos.

1. *La Onda ECG:* El trazado típico de un electrocardiograma registrando un latido cardíaco normal consiste en una onda P, un complejo QRS y una onda T como se puede apreciar en la Fig. 1. La sístole mecánica o contracción ventricular comienza justo después del inicio del complejo QRS y culmina justo antes de terminar la onda T. La diástole, que es la relajación y relleno ventricular, comienza después que culmina la sístole correspondiendo con la contracción de las aurículas, justo después de iniciarse la onda P [4].

B. La Oximetría de Pulso

La oximetría de pulso o medición del SpO_2 es un procedimiento no invasivo para medir de forma continua la saturación arterial de oxígeno (SaO_2). Habitualmente, el sensor es colocado en un dedo de la mano, aunque a veces se utilicen otras ubicaciones como el lóbulo de la oreja o un dedo del pie.

La oximetría de pulso se fundamenta en la espectrofotometría y la pletismografía. La espectrofotometría permite calcular la concentración de una determinada sustancia en una solución a partir de su absorción óptica a una longitud de onda determinada. Por otro lado, la pletismografía permite medir los cambios en el flujo sanguíneo o el volumen de aire en diferentes partes del cuerpo [5]. De esta forma, la sustancia bajo análisis se ilumina y se mide la absorción de luz a longitudes de onda específicas al pasar por un lecho vascular arterial pulsátil, y a partir de esa medida, se calcula la concentración de oxígeno. Dicha técnica analítica también establece que para analizar dos sustancias en solución se necesitan, como mínimo, dos longitudes de onda.

En el caso de la sangre, hay dos sustancias relevantes a la oxigenación que son: la hemoglobina reducida (Hb) y la oxihemoglobina (HbO_2). Como se tiene dos sustancias,

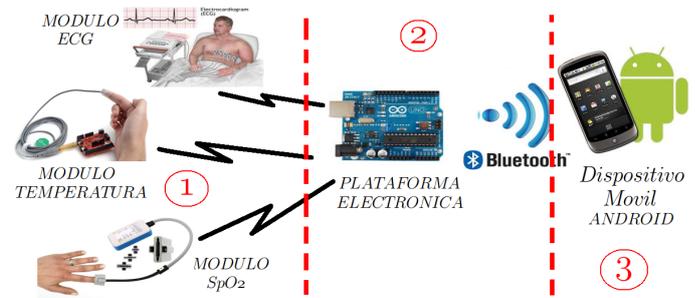


Fig. 2. Esquema general del sistema de monitorización.

los oxímetros requieren mínimo dos longitudes de onda: típicamente una roja y una infrarroja. Esto conlleva a que la luz deba atravesar parte de la sangre oxigenada (arterial), sangre reducida (venosa), y otros tejidos como huesos, uñas y piel [6].

C. La Temperatura Corporal

La temperatura corporal normal, de acuerdo con la Asociación Médica Americana, puede oscilar entre los $36,5^{\circ}C$ y $37,2^{\circ}C$. La temperatura corporal se puede determinar en tres zonas: la axila, la boca y el recto. Las dos últimas opciones son las que nos dan una idea precisa de la temperatura real del organismo, ya que el termómetro se aloja en una de sus cavidades (temperatura interna vs. temperatura externa) [7].

La temperatura del cuerpo puede ser anormal debido a la fiebre (temperatura alta) o a la hipotermia (temperatura baja). De acuerdo con la Asociación Médica Americana, se considera que hay fiebre cuando la temperatura corporal es mayor a $37^{\circ}C$. Por otro lado, la hipotermia presenta temperaturas corporales por debajo de los $35^{\circ}C$.

III. SISTEMA DE MONITORIZACIÓN

El sistema para monitorización de signos vitales cuenta con tres módulos, como se observa en la Fig. 2. Estos tres módulos generales son:

1. Los sensores que permiten la adquisición de las señales de los signos vitales.
2. La plataforma digital que permite el muestreo y procesamiento de las señales, así como el envío de los datos vía Bluetooth hacia un dispositivo móvil.
3. El dispositivo móvil que permite la visualización y análisis de los signos vitales.

A. Los Sensores

1. *Temperatura Corporal:* Para obtener la temperatura corporal no se requiere una etapa de acoplamiento debido que los sensores utilizados tienen salidas digitales que permiten comunicarse directamente con la plataforma digital, la misma que posteriormente enviar su valor vía Bluetooth al dispositivo móvil.

Para este proyecto se consideró utilizar el integrado SHT15: sensor de temperatura y humedad digital [8]. El sensor digital SHT15 está totalmente calibrado y ofrece una alta precisión y excelente estabilidad a largo plazo y

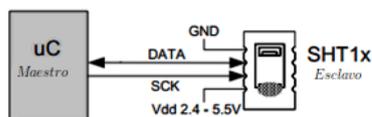


Fig. 3. Esquema de conexión del sensor de temperatura.

a muy bajo costo. Los 4 pines del sensor de temperatura son conectados como se muestra en la Fig. 3 a la plataforma digital; de esta manera, se administran directamente la señal de reloj y las señales con los datos de interés.

2. *Electrocardiograma*: Para la recepción de los datos del ECG se utiliza la configuración de 3 derivaciones, debido a que permite obtener señales aceptables para su análisis. Además, para esta configuración, los circuitos requeridos para su acoplamiento son relativamente pequeños, lo cual ayuda a mejorar la movilidad del sistema. Así, se colocan tres electrodos en puntos específicos del paciente, como se describe en la Fig. 4.

Para que la recepción de la señal sea óptima, se incluyen etapas de amplificación y filtrado, como muestra la Fig. 5. Al finalizar esta etapa de procesamiento por hardware, la señal será entonces muestreada por la plataforma digital.

La etapa de pre-amplificación utiliza un amplificador de instrumentación que permite mejorar la señal proveniente de los electrodos que se encuentra en el rango de los μV . En la etapa de filtrado se eliminan las frecuencias no deseadas de la señal, para ello se usan filtros pasa-banda (combinación de filtros pasa-altos y pasa-bajos). Finalmente, la etapa de post-amplificación genera una señal en el rango de los V para que pueda ser procesada adecuadamente por la plataforma digital.

En la presente implementación, los electrodos seleccionados son electrodos auto-adhesivos reusables, que permiten una buena recepción de los voltajes ECG en reposo. Los electrodos están conectados al brazo derecho (RA), al brazo izquierdo (LA) y a la pierna derecha (RL) del paciente (Fig. 4). El circuito diseñado posee una retroalimentación con la señal de la pierna derecha que es utilizada como referencia para reducir el ruido. Esta señal ingresa a un amplificador instrumental, en este caso el AD620, junto con la señales del brazo izquierdo y el brazo derecho para la determinación de las componentes principales de la señal del ECG.

Posteriormente, la señal obtenida pasa por un sistema de filtros que permite eliminar gran parte del ruido que aún interfiere con la señal ECG. La señal buscada se encuentra en el orden de $1-2Hz$, por lo que se aplican filtros pasa-bajos de $33Hz$ y pasa-altos de $0,16Hz$ para eliminar las frecuencias no deseadas.

Finalmente, después de filtrar y eliminar el ruido de la señal, se realiza una nueva amplificación de la misma para obtener rangos de voltaje aceptables. La señal obtenida en el prototipo posee $3V_{pp}$ y puede ser fácilmente muestreada por la plataforma digital. El sistema completo de amplificación y filtrado se muestra en la Fig. 6.

3. *Oximetría de Pulso*: Todos los sensores de SpO_2 poseen el mismo principio de funcionamiento, existiendo diferencias en cuanto al tipo conector que poseen y que

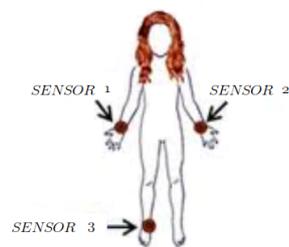


Fig. 4. Posición de los electrodos para la medición del ECG.

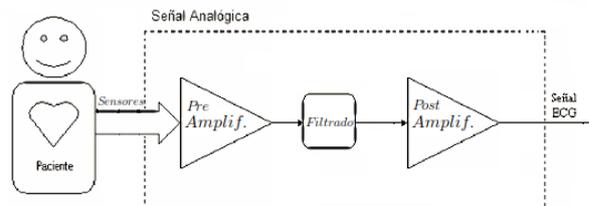


Fig. 5. Esquema general del módulo de pre-procesamiento ECG.

depende principalmente de su forma de conexión a equipos de marcas específicas. El conector más usado es el DB9, el cual facilita una manipulación flexible de la señales. En este caso, para obtener la señal de SpO_2 se necesitan 2 circuitos: uno que permita la generación de pulsos PWM para controlar el encendido y apagado de los LEDs rojo e infrarrojo, y un circuito de acoplamiento de la señal a la salida del fotodetector. Este último circuito está compuesto por una etapa de amplificación de transimpedancia (*i.e.*, conversor corriente-voltaje), y una etapa de filtrado y eliminación de voltaje DC. El esquema general del módulo SpO_2 se muestra en la Fig. 7.

El circuito de control de los LEDs determina el encendido y apagado de los mismos mediante una señal PWM. Simultáneamente, obtenemos una corriente resultante a la salida del fotodetector, la misma que se la transforma en voltaje por medio de un amplificador de transimpedancia. Una vez obtenido el voltaje proporcional a la corriente entregada por el fotodetector, la señal entra a una etapa de filtrado para eliminar frecuencias no deseadas. Finalmente, la señal pasa a una etapa de amplificación para obtener valores en el rango de los V y de esa forma pueda ser muestreada por la plataforma digital. La plataforma digital procesa la señal para obtener el valor de SpO_2 correspondiente y además envía ese valor por medio de Bluetooth hacia el dispositivo móvil para su posterior visualización.

En la actual implementación se utilizó el sensor reusable de la marca Nellcor [9] cuya correcta utilización se observa en la Fig. 8. Este tipo de sensor es perfecto para realizar el control del paciente en tiempo real y realizar un seguimiento a corto plazo, además de ser cómodo y diseñado para reducir la interferencia de la luz ambiental.

Como se mencionó, para obtener la señal final del SpO_2 es necesario realizar un acoplamiento de la señales tomadas por el sensor, así como un circuito de control que permita encender y apagar los LEDs en un determinado período de

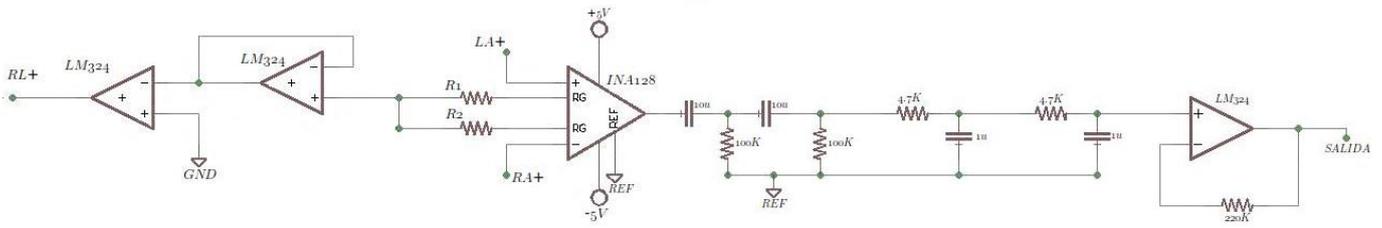


Fig. 6. Circuito de acoplamiento para generación de la señal ECG.

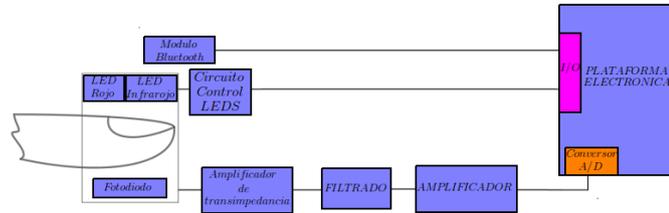


Fig. 7. Esquema general del módulo de SpO_2 .



Fig. 8. Sensor reusable para medición de SpO_2 de marca Nellcor.

tiempo. En el sensor Nellcor, los pines 2 y 3 del conector DB9 controlan el encendido y apagado de los LEDs, debido a que se encuentra conectado el cátodo de un LED con el ánodo del otro, esto nos asegura que solo un LED puede encontrarse encendido con una simple señal PWM. Los pines 5 y 9 nos dan la señal de salida del fotodetector como corriente en valores de nA . Los pines 6 y 7 representan la tierra del sistema, mientras que los pines 1, 4 y 8 no son utilizados.

Para el control de los LEDs es necesario señales PWM a $1kHz$ y 25% del ciclo de trabajo. Para obtener las señales deseadas se tiene un temporizador 555 que genera una onda de $2kHz$ de frecuencia y un ciclo de trabajo del 50%. Esta señal entra en un flip-flop JK (74LS73) que actúa como un divisor de frecuencia y cuya señal de salida permite el control de un demultiplexor (74LS155) el cual permite enviar un pulso hacia cada uno de los LEDs. El circuito para el control de las señales PWM se muestra en la Fig. 9.

A la salida del fotodetector (*i.e.*, pin 9 del conector DB9) se tiene una señal en el valor de los nA , se requiere un amplificador de transimpedancia que transforma y amplifica la corriente entregada por el sensor a una señal de voltaje proporcional a la corriente.

La señal obtenida a la salida del amplificador de transimpedancia se encuentra en el rango de los $0,5Hz$ a los $5Hz$, se usa un filtro pasa-bajos de $6Hz$, y un pasa-altos de $0,8Hz$ y se amplifica la señal con una ganancia de 31 para finalmente

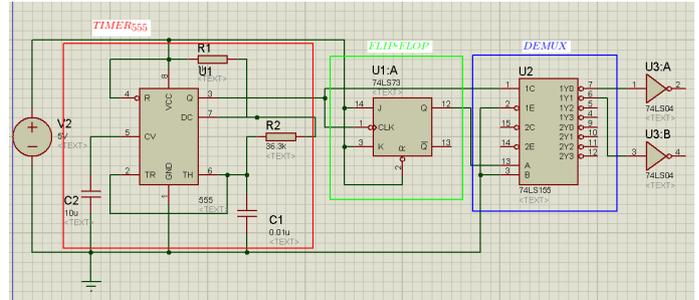


Fig. 9. Circuito generador de las señales PWM.

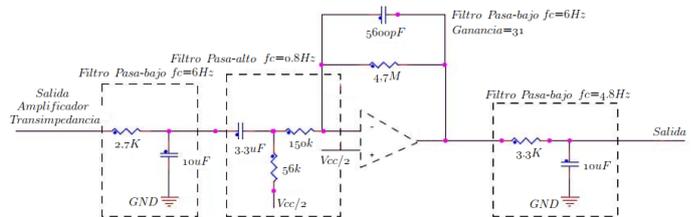


Fig. 10. Circuito de acoplamiento para la señal SpO_2 .

usar un filtro pasa-bajos de $4Hz$. El circuito final se muestra en la Fig. 10.

La señal final resultante posee una frecuencia cercana a $1Hz$ y una amplitud cercana a los $2V$, la cual es muestreada por la plataforma digital para su posterior procesamiento.

B. La Plataforma Digital

La plataforma digital, al igual que los sensores, debe cumplir con determinadas especificaciones que permitan un óptimo procesamiento de los datos. Las características más importantes son:

- Tener un software y hardware flexibles, y fáciles de utilizar.
- Poseer un microprocesador de propósito general que permita manipular eficientemente los datos y señales requeridas.
- Tener entradas y salidas tanto analógicas como digitales que permitan utilizar diferentes tipos de sensores para cada uno de los módulos a ser implementados y también permitan accionar diferentes actuadores.
- Las entradas analógicas deben proveer una buena resolución de muestreo (al menos 10 bits).
- Debe ser posible alimentar al sistema con una fuente externa no regulada y con voltajes variables.

Tabla I
PULSACIONES POR MINUTO EN REPOSO.

Paciente	Valor medido	Valor real
1	88	84
2	81	83
3	60	63
4	67	66
5	77	81
6	66	69
7	93	95
8	73	72
9	82	80
10	75	72

Tabla II
ERROR EN LA MEDICIÓN DE PULSACIONES EN REPOSO.

Paciente	Error relativo	Error absoluto
1	4	4.76
2	2	2.41
3	3	4.76
4	1	1.51
5	4	4.94
6	3	4.35
7	2	2.10
8	1	1.39
9	2	2.50
10	3	4.17

- El tamaño de los componentes debe ser lo más compacto posible para la optimización de espacio.

Para este proyecto se ha seleccionado la plataforma Arduino, la cual tiene fácil acceso a sus entradas y salidas, y es ideal para la conexión de sensores o dispositivos externos. Esta plataforma es adecuada para proyectos como el planteado debido a su ambiente de desarrollo simplificado y programación directa de su hardware [10].

C. El Dispositivo Móvil

El dispositivo móvil permite la visualización de las gráficas correspondientes al ECG, las medidas de SpO_2 y la temperatura corporal. Todas estas señales son enviadas mediante un módulo Bluetooth desde la plataforma digital para su posterior recepción en el dispositivo móvil.

Para la presentación de los datos transmitidos se tiene una variedad de dispositivos móviles como son teléfonos celulares, computadores portátiles, tabletas, etc. En la actual implementación se ha seleccionado el sistema operativo Android para el desarrollo de la aplicación. Esta selección se debe a que Android brinda una plataforma ampliamente difundida y un mercado abierto para la distribución de la misma.

IV. EVALUACIÓN DEL SISTEMA

Para el análisis de resultados se utilizó la siguiente metodología: se tomó los signos vitales a una muestra de 10 personas, controlando las características y estado de las personas para limitar posibles alteraciones a los parámetros normales de sus signos vitales.

A. El Electrocardiograma

Se midieron las pulsaciones (*i.e.*, latidos por minutos) de cada persona en estado de reposo mediante el sistema de monitorización, para luego validar el pulso de forma manual. Los resultados obtenidos para la muestra de 10 pacientes en el dispositivo móvil se muestran en las tablas I y II.

Según la información que aparece en la tabla II, el error porcentual es en media 3.3%. Este porcentaje es considerado como aceptable al tomar en consideración la simplicidad y bajo costo de la plataforma utilizada.

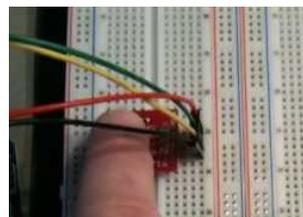


Fig. 11. Medición del valor de temperatura mediante el sensor SHT15.

B. La Temperatura

Para realizar las mediciones de temperatura corporal se coloca el dedo índice sobre el sensor como se observa en la Fig. 11. Este esquema mide de forma externa la temperatura pudiendo generar errores significativos. Debido a lo anterior, la temperatura corporal medida es comparada con los valores arrojados por un termómetro clínico que nos da una lectura mucho más exacta de este signo vital.

Al tener errores considerables entre la medición del sistema de monitorización y el termómetro clínico, calculamos un factor de corrección promedio igual a 1.07, que fue utilizado para corregir las mediciones y cálculo del error. Los datos de error absoluto y relativo para cada medición de temperatura usando el factor de corrección calculado se muestran en las tablas III y IV.

Vemos que el error porcentual medio de las mediciones de temperatura una vez aplicado el factor de corrección es de 0.95%. Este es un error bastante aceptable que genera una medición bastante buena del signo vital temperatura.

C. La Oximetría de Pulso

En la tabla V se muestran los valores de SpO_2 tomados para la muestra de 10 personas. El rango normal del valor de

Tabla III
VALORES DE TEMPERATURA CON FACTOR DE CORRECCIÓN.

Paciente	Temperatura corregida	Temperatura real
1	37.05	36.5
2	36.27	37.2
3	36.62	36.3
4	35.83	36.2
5	37.27	37.6
6	36.47	36.6
7	36.97	36.8
8	36.65	36.4
9	36.53	36.4
10	37.22	36.9

Tabla IV
ERROR EN LA TEMPERATURA CON FACTOR DE CORRECCIÓN.

Paciente	Error Relativo	Error Absoluto
1	0.55	1.51
2	0.93	2.50
3	0.32	0.88
4	0.37	1.02
5	0.33	0.88
6	0.13	0.35
7	0.17	0.46
8	0.25	0.69
9	0.13	0.36
10	0.32	0.87

Tabla V
VALORES MEDIDOS PARA SpO_2 .

Paciente	SpO_2
1	94
2	89
3	92
4	94
5	86
6	87
7	90
8	90
9	93
10	88

SpO_2 se encuentra entre el 85 % y el 95 %, por lo que los valores obtenidos se pueden considerar adecuados. Además, se verificó la coincidencia del sincronismo existente entre la señal entregada por el sensor de SpO_2 y la señal ECG.

V. CONCLUSIONES

De acuerdo al análisis de los errores obtenidos, los datos medidos a partir del ECG y temperatura corporal se ubican en rangos del 4 % y 1 % respectivamente, lo que garantiza una adecuada precisión de las mediciones. Las mediciones del SpO_2 se encuentran dentro de los rangos normales para cada paciente, además que coincide la validación de sincronismo al comparar dicha señal con la del ECG. Estos márgenes de error son considerados especialmente aceptables al considerar la simplicidad y bajo costo de la plataforma utilizada.

La plataforma digital utilizada (*i.e.*, Arduino) para la medición de los signos vitales es de fácil manejo y transporte, permitiendo movilidad para su utilización en diferentes lugares y circunstancias. La obtención de datos mediante esta plataforma de monitorización ayudará a realizar un diagnóstico preventivo ágil de los pacientes.

REFERENCIAS

[1] Enrique V. Carrera Pamela Morales. Monitoreo de condiciones de salud mediante una red de dispositivos móviles. Technical report, Escuela Politécnica del Ejército, 2012.

[2] Medline Plus. Signos Vitales. <http://www.nlm.nih.gov/medlineplus/spanish/ency/article/002341.htm>, year = 1997, note =.

[3] D C Godoy León. Proyecto diseño y construcción de una plataforma de telemedicina para el monitoreo de bioseñales. <http://www.dalcame.com/wdescarga/ecg12.pdf>, 2008. [Online; accessed 12-Feb-2012].

[4] Francis Cozal. Electrocardiograma. <https://es.wikipedia.org/wiki/Electrocardiograma>, 2013. [Online; accessed 08-Feb-2013].

[5] Seco González A. Noguero Casado MJ. Pulsioximetría. <http://www.fisterra.com/material/tecnicas/pulsioximetria/pulsioximetria.pdf>. [Online; accessed 08-Feb-2013].

[6] Santiago Lopez. Pulse Oximeter Fundamentals and Design. http://www.freescale.com/files/32bit/doc/app_note/AN4327.pdf, 2012. [Online; accessed 08-Feb-2013].

[7] University Of Chicago. Signos Vitales: Temperatura Corporal, Pulso, Frecuencia Respiratoria y Presión Sanguínea. <http://www.uchospitals.edu/online-library/content=S03963>, 2013. [Online; accessed 01-Marzo-2013].

[8] Sensirion. SHT15. <http://www.sensirion.com/en/products/humidity-temperature/humidity-sensor-sht15/>. [Online; accessed 01-Marzo-2013].

[9] NONIN. Sensores Spo2 Nonin. <http://www.nonin.com/PulseOximetry/Sensors>. [Online; accessed 01-Marzo-2013].

[10] MCM. Arduino Uno vs Raspberry Pi vs BeagleBone Black. <http://blog.mcmelectronics.com/post/Arduino-Uno-Raspberry-Pi-and-BeagleBone-Black#.UeXP8c9qYxA>, 2013. [Online; accessed 16-Jul-2013].

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

Universidad de las Fuerzas Armadas – ESPE

El Departamento de Eléctrica y Electrónica (antigua Facultad de Ingeniería Electrónica) fue creado el 25 de abril de 1977 e inicia sus labores en octubre del mismo año en la modalidad presencial, con sus planes y programas de estudio encaminados a la formación del Ingeniero Electrónico. A partir de octubre de 1990, ofrece dos carreras con perfiles profesionales definidos: Ingeniería Electrónica en Telecomunicaciones e Ingeniería Electrónica en Computación.

En la propuesta actual del Departamento se integran en un sólo perfil estas aspiraciones, ofreciendo las siguientes carreras: Ingeniería en Electrónica y Telecomunicaciones, Ingeniería en Electrónica, Automatización y Control, e Ingeniería en Electrónica, Redes y Comunicación de Datos.

Además oferta los siguientes programas de posgrado: Maestría en Redes de Información y Conectividad, y Maestría en Gerencia de Redes y Telecomunicaciones.



Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas – ESPE
Copyright © 2013