

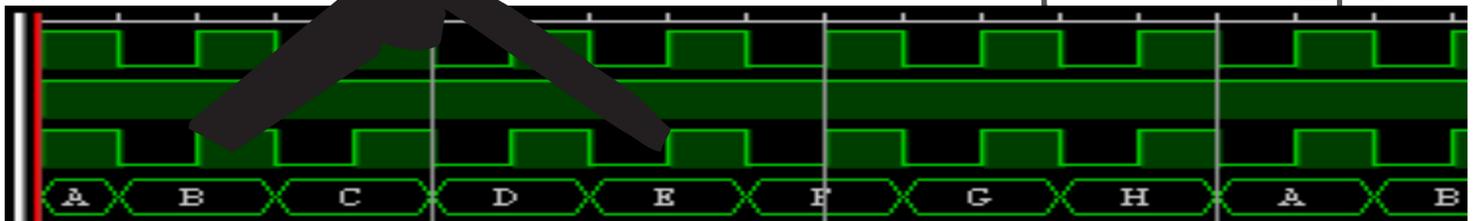
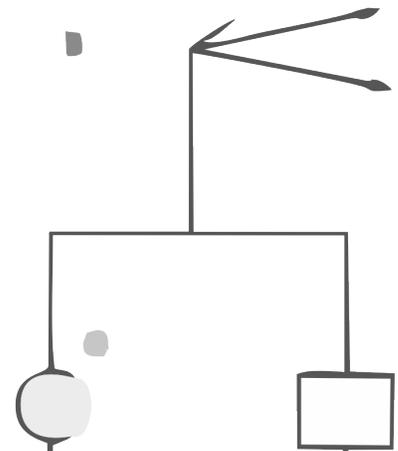
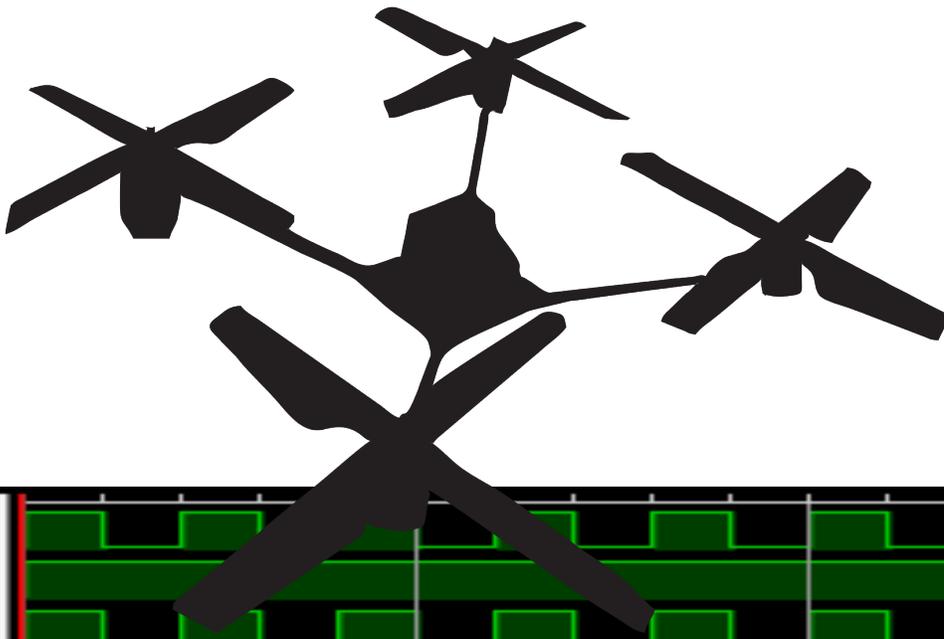


DIFU 100CI@



REVISTA DE DIFUSIÓN CIENTÍFICA

INGENIERÍA Y TECNOLOGÍAS



DIFU100ci@ (léase difuciencia) Vol. 10, No.1, mayo-agosto 2016, es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, "Francisco García Salinas", a través de la Unidad Académica de Ingeniería Eléctrica. Jardín Juárez No. 147, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Tel. (01 492) 925 6690. <http://difu100cia.uaz.edu.mx>. Editor responsable: Jorge Flores Troncoso, email: jflorest@uaz.edu.mx.

Reservas de Derechos al Uso Exclusivo No. 04-2015-060212215100-203, ISSN: 2007-3585, ambos otorgados por el Instituto Nacional de Derecho de Autor. Responsable de la última actualización de este número Jorge Flores Troncoso, López Velarde 801, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Fecha de última actualización 15 de septiembre de 2016. Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación. Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Universidad Autónoma de Zacatecas "Francisco García Salinas" a través de la Unidad Académica de Ingeniería Eléctrica.

Producción

**Universidad Autónoma de Zacatecas
"Francisco García Salinas"**

Diseño y Producción

Jorge Flores Troncoso

Diseño de Portada

José Ricardo Gómez Rodríguez

D.R. de la Presente Edición

**Universidad Autónoma de Zacatecas
"Francisco García Salinas"**

**López Velarde 801, Centro
98010 Zacatecas, Zac. México**

ISSN 2007-3585

DIFU100ci@ (léase difucencia) Vol. 10, No.1, mayo-agosto 2016, es una publicación cuatrimestral editada por la Universidad Autónoma de Zacatecas, "Francisco García Salinas", a través de la Unidad Académica de Ingeniería Eléctrica. Jardín Juárez No. 147, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Tel. (01 492) 925 6690. <http://www.uaz.edu.mx>, <http://difu100cia.uaz.edu.mx>. Editor responsable: Jorge Flores Troncoso, email: jflorest@uaz.edu.mx.

Reservas de Derechos al Uso Exclusivo No. 04-2015-060212215100-203, ISSN: 2007-3585, ambos otorgados por el Instituto Nacional de Derecho de Autor. Responsable de la última actualización de este número Jorge Flores Troncoso, López Velarde 801, Colonia Centro, Zacatecas, Zacatecas, C.P. 98000. Fecha de última actualización 15 de septiembre de 2016.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación. Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Universidad Autónoma de Zacatecas "Francisco García Salinas" a través de la Unidad Académica de Ingeniería Eléctrica.

Hecho en México
Made in Mexico

Directorio

Dr. Antonio Guzmán Fernández Rector
Ruben Ibarra Reyes Secretario General
Dr. Luis Alejandro Aguilera Galaviz Secretario Académico
Dr. Edmundo Guerrero Sifuentes Secretario Administrativo
Dr. Manuel Reta Hernández Coord. Investigación y Posgrado
Dra. Georgia Aralu González Pérez Coord. Depto. Editorial

Editores Ejecutivos

Jorge Flores Troncoso, Editor en Jefe, UAZ, México
Manuel Reta Hernández, UAZ, México
Claudia Sifuentes Gallardo, UAZ, México
Juvenal Villanueva Maldonado, CONACyT-UAZ, México

Editores Asociados

Salvador Ibarra Delgado, UAZ, México
Remberto Sandoval Aréchiga, UAZ México
Israel Alejandro Arriaga Trejo, CONACyT-UAZ, México
Jorge Simón Rodríguez, CONACyT-UAZ, México

Grupo Revisor

Oliver Joekisch, University of Applied Sciences, Germany
Ding Hongwei, Tongji University, China
Robert Weissbach, Penn State Erie, The Behrend College
Naim Logic, Arizona Public Service
Leonardo Acho Zuppa, U Politècnica de Catalunya, España
Miguel Andrés, U. de Valencia, España
Manuel Hernández Calviño, Universidad de La Habana, Cuba
Luis C. García Santander, Universidad de Concepción, Chile
José Ángel González Fraga, UABC, México
Omar Álvarez Xochihua, UABC, México
Ramón Parra Michel, CINVESTAV, México
Leonel Soriano Equigua, FIME-UColima, México
José Luis Álvarez Flores, FIME-UColima, México
Geminiano D. Martínez Ponce, CIO, México
Jaime Sánchez García, CICESE México
Salvador Villarreal Reyes, CICESE, México
Enrique Pacheco Cabrera, AEM, México
Ernesto García Domínguez, UAZ México
Rafael Villela Varela, UAZ México
Jorge de la Torre y Ramos, UAZ México

Contenido

Editorial	p. 1
Integración de la generación distribuida a la red eléctrica.	
Análisis en estado estable	pp. 2–7
Diseño de estructuras para vehículos aéreos no tripulados	pp. 8–11
Implementación en Hardware de un Entrelazador/Desentrelazador de Datos con Interface AXI Stream	pp. 12–18
Modelado Matemático y Control PD de un Quadrotor	pp. 19–27
Software Educativo Didáctico para la Oralización de Niños Hipoacúsicos	pp. 28–33

Editorial

La revista *DIFU*₁₀₀*ci*@ es una revista cuatrimestral que comenzó su publicación oficial en 2005. En mayo del 2012, la revista *DIFU*₁₀₀*ci*@ adquirió el ISSN.

Desde entonces, se pretende contribuir a la difusión del conocimiento de la comunidad académica tanto nacional como internacional mediante la difusión de resultados de investigación de alta calidad. La Revista se centra en obras originales, que incluyen principalmente los estudios experimentales, análisis numéricos, estudios de casos y revisiones bibliográficas que proporcionan una significativa contribución a las áreas de ingeniería y tecnología en todas las disciplinas (Electrónica, Eléctrica, Ciencias de la Computación, Mecatrónica, Robótica, Telecomunicaciones, Procesamiento de señales, Ingeniería Industrial, Ingeniería de Control, y Bioingeniería).

Desde el comienzo, la revista ha buscado la mejora de los artículos aceptados para su publicación por un proceso de evaluación por pares o árbitro de los manuscritos recibidos. Estas evaluaciones son llevadas a cabo por expertos de reconocido prestigio por sus conocimientos y logros académicos, con el objetivo de asegurar que las publicaciones seleccionadas están contribuyendo al estado del arte en diferentes áreas de interés. Además, desde su inicio, la revista se ha abierto a los estudiantes y académicos a través del Sistema Open Journal, facilitando todo el proceso de presentación y publicación.

Agradezco a los autores y revisores, que se esfuerzan para mejorar la calidad de los manuscritos. Exhorto a todos los investigadores, académicos y estudiantes en las áreas de ingeniería y tecnología para que continúen sometiendo sus artículos en nuestra revista y contribuir a la noble difusión de la ciencia y la tecnología.

Jorge Flores Troncoso
Editor en Jefe, Revista *DIFU*₁₀₀*ci*@
Universidad Autónoma de Zacatecas

Integración de la generación distribuida a la red eléctrica. Análisis en estado estable

Antonio José Ramón Martínez Rangel, Manuel Reta Hernández, Francisco Bañuelos Ruedas

Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica.
Av. López Velarde 801, Col. Centro, Zacatecas, Zac., México, 98000.
mtzr2008@gmail.com

2016 Published by *DIFU*_{100ci@} <http://difu100cia.uaz.edu.mx>

Resumen

La generación de energía eléctrica a partir de fuentes de generación distribuida (GD) ha ido incrementando su utilización de manera gradual las últimas décadas. Dentro de los sistemas eléctricos, la integración de fuentes de generación distribuida requiere de una atención especial, debido a la propia naturaleza de las energía renovables. Idealmente, la GD interconectada a la red eléctrica centralizada no debe ocasionar ningún problema, por lo que es necesario que siempre se mantengan los niveles de estabilidad en la red. El documento presenta el análisis en estado estable de una posible micro red conectada a la red eléctrica del Campus Siglo XXI de la Universidad Autónoma de Zacatecas; el objetivo es verificar el estado de operación de la micro red interactuando con la red principal del campus, y de los diferentes componentes de la red principal (líneas de distribución, transformadores y buses).

Palabras clave: Análisis en estado estable, Flujos de potencia, Generación distribuida, Micro red.

1. Introducción

Desde 1990 ha habido un especial interés en conectar fuentes de generación distribuida (GD) a las redes de distribución [1]. Esto implica entender la operación y el control de las diferentes plantas de generación. Usualmente es requerido un estudio para evaluar el rendimiento del sistema eléctrico con las nuevas fuentes de generación, bajo condiciones normales y anormales de operación. El rendimiento de la red junto con las fuentes de GD es estudiado con programas computacionales.

El objetivo principal de un sistema eléctrico es satisfacer la demanda de los consumidores; esto representa un gran reto para el sistema en la parte de diseño y en la parte operacional. En un sistema tan complejo se toman en cuenta algunas políticas operacionales que garanticen la estabilidad del sistema; por ejemplo, la magnitud de los voltajes en los nodos y la frecuencia del sistema deben mantenerse dentro de los límites establecidos, las formas de onda de corriente y voltaje deben mantener su forma sinusoidal, las líneas de transmisión y distribución deben operar bajo los límites térmicos y de estabilidad, y las interrupciones deben ser

mínimas. Además, los generadores debe satisfacer la demanda y las pérdidas, y operar dentro de los límites especificados de potencia activa y reactiva.

El estudio de flujos de potencia en los sistemas eléctricos permite determinar las condiciones de operación del sistema en estado estable, cuyo estado es determinado al encontrar los flujos de potencia activa y reactiva, la magnitud de voltaje y su ángulo de voltaje en todos los buses para ciertas condiciones de carga. Si el análisis arroja que las magnitudes de los voltajes están fuera de límites en uno o más puntos de la red se toman las medidas necesarias para regular el voltaje. Similarmen- te, si el análisis muestra que los flujos de una línea de transmisión están fuera de la capacidad de carga, se toman acciones de control para corregir el desbalance. Los estudios de flujos de potencia también permiten analizar las posibles expansiones futuras de la red [2].

2. Estado del arte

El problema de flujos de potencia reside en determi- nar el estado estable de los voltajes complejos de todos los buses de la red, además de calcular los flujos de potencia activa y reactiva de las líneas de transmisión y transformadores. El sistema eléctrico es representado por un conjunto de ecuaciones no lineales cuya solución numérica es encontrada mediante técnicas iterativas [3].

El problema de flujos de potencia parte del diagrama unifilar del sistema, de donde se pueden obtener los datos de entrada para la solución. Los datos de entrada consisten en obtener parámetros en los buses, líneas de transmisión y transformadores. Para cada bus del sistema hay cuatro variables asociadas, magnitud de voltaje V_k , ángulo de fase δ_k , potencia real neta P_k y la potencia reactiva suministrada al bus Q_k . En cada bus, dos de esas variables son especificadas como datos de entrada, y los otros dos datos son calculados por el programa de flujos de potencia. La figura 1 muestra un bus con las cuatro variables mencionadas.

Por conveniencia, la potencia entregada al bus k es separada en términos de potencia generada y potencia de carga como se indica en las Ec. (1) y (2).

$$P_k = P_{Gk} - P_{Lk} \quad (1)$$

$$Q_k = Q_{Gk} - Q_{Lk} \quad (2)$$

Para la selección de las variables de bus se debe analizar y entender la categorización de cada bus, de acuerdo con los tres tipos de buses o barras existentes [3]-[4].

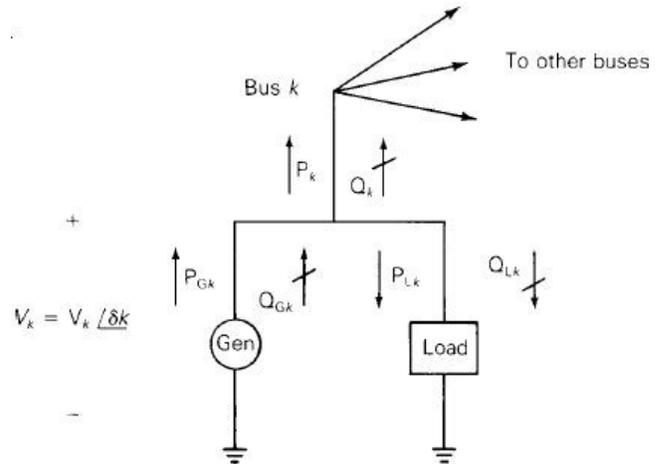


Figura 1. Bus de variables V_k , δ_k , P_k y Q_k

Bus slack o Bus de compensación. El bus slack es referenciado como un bus en el cual se conocen los valores conocidos de V_k y δ_k , típicamente $1 \angle 0^\circ$ en por unidad, y los valores de P_k y Q_k son calculados.

Bus de carga (PQ). Los valores de P_k y Q_k son conocidos y los valores V_k y δ_k son calculados. Estos nodos no cuentan con generadores conectados.

Bus de voltaje controlado (PV). P_k y V_k son los datos conocidos del bus y los valores de potencia reactiva y ángulo de fase son calculados (Q_k y δ_k).

Las líneas de transmisión se representan por su circuito π equivalente monofásico, con sus valores relevantes de impedancia serie y admitancia en derivación, los buses a los que la línea es conectada y el valor máximo de potencia reactiva [4]-[5].

El estudio de flujos de potencia comienza con la construcción de la matriz de admitancias " Y_{bus} ", la cual se construye a partir de los datos de admitancia de los elementos del sistema, y tiene como parámetros en la diagonal principal, los elementos Y_{kk} que corresponden a la suma de admitancias conectadas al bus k , mientras que en los elementos fuera de la diagonal son elementos Y_{km} [3].

La evaluación del estado estable de un sistema de potencia se enfoca en escribir las ecuaciones de cada bus de la red para mostrar la relación entre las potencias de generación P_G , la carga o demanda P_L , y las potencias intercambiadas (P_{calc}) a través de los elementos de transmisión conectados a los buses, donde la suma de las potencias debe de ser cero. Este enfoque permite llegar a la definición del error (ΔP_k), definido como

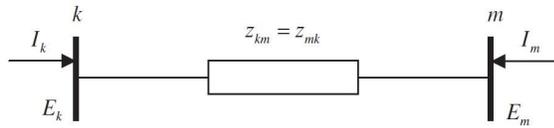


Figura 2. Impedancia equivalente

el valor programado ($P_{k,prog}$) menos el valor calculado ($P_{k,calc}$) [2]-[5] como se indica en las Ec. (3) y (4).

$$\begin{aligned} \Delta P &= P_{k,prog} - P_{k,calc} \\ &= P_{Gk} - P_{Lk} - P_{k,calc} = 0 \end{aligned} \quad (3)$$

$$\begin{aligned} \Delta Q &= Q_{k,prog} - Q_{k,calc} \\ &= Q_{Gk} - Q_{Lk} - Q_{k,calc} = 0 \end{aligned} \quad (4)$$

Para el desarrollo de las ecuaciones de flujo de potencia es necesario encontrar la relación entre las corrientes inyectadas y los voltajes de los buses. La figura 2 muestra dicho valor de corriente compleja del bus, denotada por I_k , expresada en términos del voltaje complejo de buses, E_k y E_m (Ec. (5) y (6)).

$$I_k = \frac{1}{Z_{km}}(E_k - E_m) = Y_{km}(E_k - E_m) \quad (5)$$

$$I_m = \frac{1}{Z_{mk}}(E_m - E_k) = Y_{mk}(E_m - E_k) \quad (6)$$

La potencia aparente compleja inyectada al bus k (Ec. (7)) consiste en las componentes de la potencia activa y reactiva y puede ser expresada en función del voltaje de nodo y la corriente inyectada al bus.

$$S_k = P_k + jQ_k = E_k I_k^* = E_k(Y_{kk}E_k + Y_{km}E_m)^* \quad (7)$$

Un sistema de potencia que consiste en más de dos buses y elementos de transmisión, se puede expresar en términos más generales en cuanto a los flujos netos de potencia inyectados al bus analizado (potencia activa y reactiva). Esto llama a las Ec. (3) y (4) a ser expresadas en términos más generales; con los flujos netos de potencia inyectados al bus k expresados como la suma del flujo de potencias de cada uno de los elementos de transmisión conectados a dicho bus. La figura 3 muestra los balances de potencia, activa y reactiva.

Las ecuaciones (3) y (4) son reescritas como se indica en las Ec. (8) y (9):

$$\Delta P_k = P_{Gk} - P_{Lk} - \sum_{i=1}^n P_{k,calc}^i = 0 \quad (8)$$

$$\Delta Q_k = Q_{Gk} - Q_{Lk} - \sum_{i=1}^n Q_{k,calc}^i = 0 \quad (9)$$

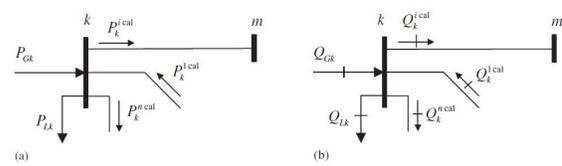


Figura 3. Balance de potencias en el bus k : (a) potencia activa, (b) potencia reactiva.

La potencia compleja que fluye a través de los elementos de transmisión se puede calcular a partir de los parámetros que interactúan entre dos nodos, como las admitancias, susceptancias, conductancias y voltajes. Los modelos correspondientes a la potencia activa y reactiva se muestran en las Ec. (10) y (11)

$$P_{k,calc} = V_k^2 G_{kk} + V_k V_m [G_{km} \cos(\theta_k - \theta_m) + B_{km} \sin(\theta_k - \theta_m)] \quad (10)$$

$$Q_{k,calc} = -V_k^2 B_{kk} + V_k V_m [G_{km} \sin(\theta_k - \theta_m) - B_{km} \cos(\theta_k - \theta_m)] \quad (11)$$

donde: V es el voltaje, G es la conductancia, B es la susceptancia y θ es el ángulo de fase.

2.1. Método de Newton-Raphson

El estudio de flujos de potencia es analizado comúnmente por el método Newton-Raphson, caracterizado por ser un método de rápida convergencia [2]. La base del método es la expansión de la serie de Taylor para una función de dos o más variables, para resolver el problema de flujos de potencia [5]. El método resuelve un conjunto de ecuaciones no lineales (Ec. (12)).

$$F(X) = \begin{cases} f_1(x_1, x_2, \dots) = 0 \\ f_2(x_1, x_2, \dots) = 0 \\ \vdots \\ f_N(x_1, x_2, \dots) = 0 \end{cases} \quad (12)$$

donde F es el conjunto de n ecuaciones no lineales y X es el vector de n variables de estado desconocidas.

El método consiste en determinar el vector de variables de estado X mediante la expansión de la serie de Taylor de $F(X)$ sobre una estimación inicial de $X(0)$ (Ec. (13)).

$$F(X) = F(X^{(0)}) + J(X^{(0)})(X - X^{(0)}) + \text{términos de orden mayor} \quad (13)$$

donde $J(X^{(0)})$ es una matriz de derivadas parciales de primer orden de $F(X)$ con respecto a X , evaluado en $X = X(0)$.

La Ec. (13) en forma expandida es expresada en (14).

$$\begin{aligned}
 & \begin{bmatrix} f_1(X^{(1)}) \\ f_1(X^{(1)}) \\ \vdots \\ f_n(X^{(1)}) \end{bmatrix} \approx \begin{bmatrix} f_1(X^{(0)}) \\ f_1(X^{(0)}) \\ \vdots \\ f_m(X^{(0)}) \end{bmatrix} + \\
 & + \left[\begin{array}{cccc} \frac{\partial f_1(X)}{\partial x_1} & \frac{\partial f_1(X)}{\partial x_2} & \cdots & \frac{\partial f_1(X)}{\partial x_n} \\ \frac{\partial f_2(X)}{\partial x_1} & \frac{\partial f_2(X)}{\partial x_2} & \cdots & \frac{\partial f_2(X)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(X)}{\partial x_1} & \frac{\partial f_n(X)}{\partial x_2} & \cdots & \frac{\partial f_n(X)}{\partial x_n} \end{array} \right]_{X=X^{(0)}} \begin{bmatrix} X_1^{(1)} - X_1^{(0)} \\ X_2^{(1)} - X_2^{(0)} \\ \vdots \\ X_n^{(1)} - X_n^{(0)} \end{bmatrix} \quad (14)
 \end{aligned}$$

donde $X^{(1)}$ representa al valor de las variables de estado calculadas después de la primera iteración.

La forma generalizada de la expresión anterior para el caso de la iteración (i) es expresada en Ec. (15).

$$F(X^{(0)}) \approx F(X^{(i-1)}) + J(X^{(i-1)})(X^{(i)} - X^{(i-1)}) \quad (15)$$

donde $i = 1, 2 \dots$. Además, si se asume que $X(i)$ está suficientemente cercano a la solución X^* , entonces $F(X^{(i)}) \approx F(X^*) = 0$.

Por tanto, la expresión se convierte en la Ec. (16), al resolver para $X^{(i)}$:

$$X^{(i)} = X^{(i-1)} - J^{-1}(X^{(i-1)})F(X^{(i-1)}) \quad (16)$$

La solución iterativa puede ser expresada como una función del vector de corrección. Por tanto, la Ec. (17) se reescribe como se expresa en la Ec. (18), y las estimaciones iniciales se actualizan con la Ec. (19).

$$\Delta X^{(i)} = X^{(i)} - X^{(i-1)} \quad (17)$$

$$\Delta X^{(i)} = -J^{-1}(X^{(i-1)})F(X^{(i-1)}) \quad (18)$$

$$X^{(i)} = X^{(i-1)} + \Delta X^{(i)} \quad (19)$$

Los cálculos se repiten tantas veces como sea necesario usando la mayoría de los datos que se tengan disponibles de X , hasta que el error ΔX esté dentro de la tolerancia prescrita (usualmente es un valor de $1e^{-12}$) [3].

3. Caso de estudio

El caso de estudio analiza una posible micro red conectada a la red eléctrica principal del Campus Siglo XX de la Universidad Autónoma de Zacatecas (UAZ). Las micro redes son sistemas eléctricos limitados por fuentes de GD en una área determinada. Usualmente, esta clase de sistemas incluye fuentes de generación renovables y tradicionales, un sistema de cargas o demanda y opcionalmente almacenamiento [6]-[7]. Además, presentan una alternativa para la generación de energía limpia y sustentable, y los beneficios son diversos. La GD permite generar energía cerca de la carga, y puede ser conectada en un amplio rango de niveles de voltaje, desde 120/240 V a 230 kV [8]. Los beneficios de una micro red para el usuario implican el incremento en la confiabilidad, aumento en la calidad de la energía, reducción de las interrupciones, uso eficiente y menor costo de la energía. Para el operador de la red centralizada representan la reducción de pérdidas en líneas de transmisión y distribución, abasto en zonas remotas, liberación de capacidad del sistema, mayor control de la energía, mayor regulación de voltaje, disminución en la inversión y reducción de fallas en el sistema [9]. Además, el uso de fuentes de energía renovable son de fácil adaptación a las condiciones del sitio y contribuyen a la disminución de gases contaminantes.

La micro red propuesta está conformada por tres fuentes de GD, un sistema solar fotovoltaico (SFV), celdas de combustible (CC) y una microturbina. Las fuentes de GD suministran el 80% de la demanda total, y el resto es tomado de la red de Comisión Federal de Electricidad (CFE).

El análisis se realizó con NEPLAN® para encontrar el comportamiento de la micro red conectada a la red eléctrica principal del campus. La figura 4 muestra el diagrama unifilar de la red eléctrica principal del campus cuyos datos de líneas de distribución, parámetros nodales y datos de los transformadores se detallan en [10].

Actualmente, las instalaciones del campus tienen una demanda de 1,153,320 kWh/año, con una carga total instalada es de 2,340.3 kW. La red eléctrica consta de la acometida de CFE, 17 buses, 9 líneas de distribución subterráneas, 8 transformadores y 8 cargas.

3.1. Resultados

La simulación de la red eléctrica principal con los módulos de GD conectados a la red son analizados en este apartado. El SFV se conectó al nodo 11, en el que está conectada la carga más grande. Los generadores

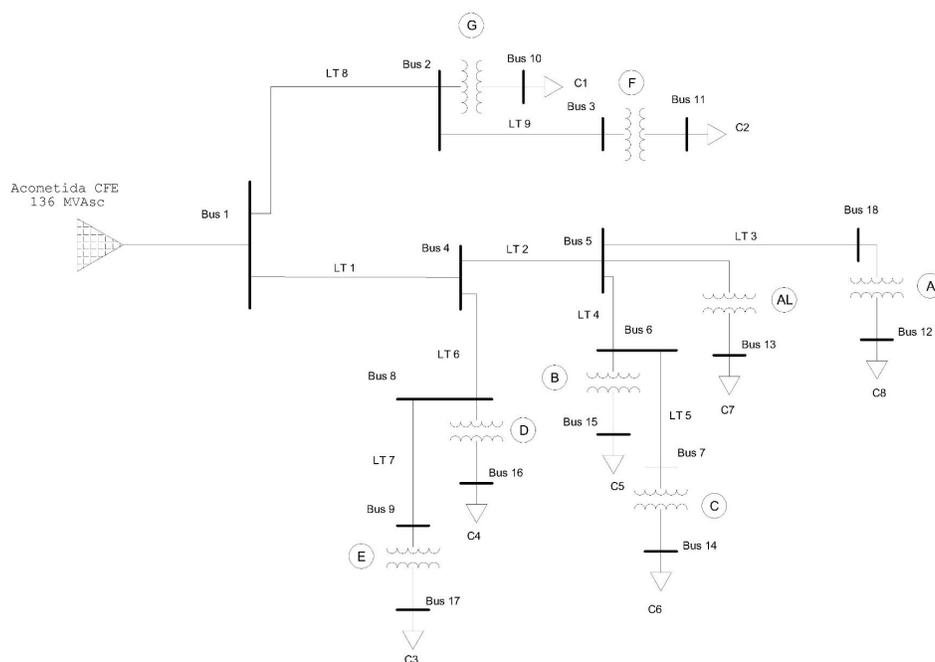


Figura 4. Diagrama unifilar de la red eléctrica del campus.

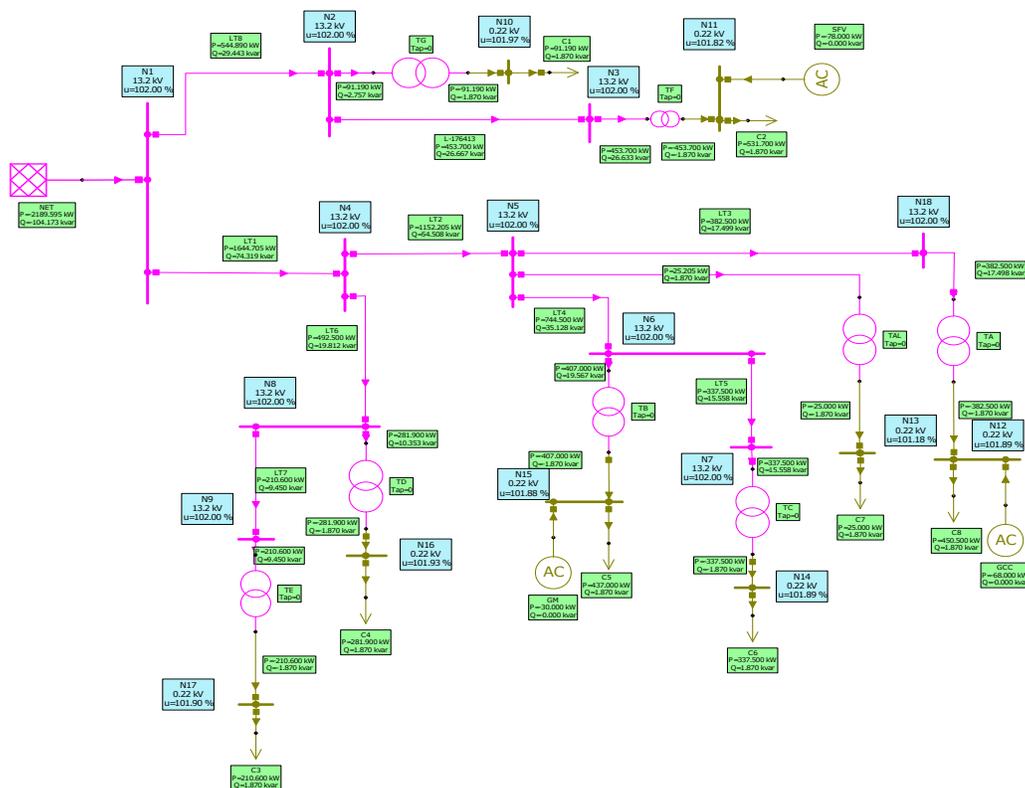


Figura 5. Resultados de flujos de potencia con GD.

de celdas de combustible y microturbina se conectan a los nodos 12 y 15 respectivamente.

Los resultados con GD conectada muestran al nodo slack (acometida de CFE) que aporta 2,189.5 kW y 104.2 kVar. El SFV aporta 78 kW, el sistema de celdas de combustible aportan 68 kW y la microturbina aporta 30 kW. La generación total de potencia activa en la red es de 2,365.5 kW. Los niveles de penetración de cada fuente de generación son: SFV 3.29 %, celdas de combustible 2.87 % y microturbina 1.27 %. Las pérdidas totales en la red fueron de 205 W. La figura 5 muestra los resultados de flujos de potencia con GD conectada a la red.

4. Conclusiones

De los resultados del análisis en estado estable en el escenario de carga y generación máxima, se observa que la micro red presenta un buen comportamiento. Los sistemas de GD conectados a la red principal de distribución operan sin problemas, el sistema es seguro, la regulación de voltaje está dentro de los límites permitidos y las líneas de transmisión y transformadores no tienen problemas de sobrecarga. Para expansiones futuras de la red, el sistema existente sería capaz de agregar nuevas fuentes de GD a la micro red e incrementar la carga sin ningún problema, aunque para asegurarse de tal propósito se realizaría un nuevo estudio de flujos de potencia.

Desde el punto de vista técnico para este caso, la integración de las fuentes de GD es conveniente por los beneficios adquiridos. De esta manera el usuario tendrá el control de sus generadores, decidirá cuando comprar energía a CFE y cuando venderle, así como determinar los tiempos de mantenimiento de la micro red.

Referencias

- [1] N. Jenkins, J.B. Ekanayake and G. Strbac, *Distributed Generation*, London, UK: The institution of Engineering and Technology, pp. 1-20, 2010.
- [2] E. Acha, C.R. Fuerte-Esquivel, H. Abriz-Perez, C. Angeles-Camacho, "Conventional Power Flow," in *FACTS Modelling and Simulation in Power Networks*, WILEY, England, pp. 93-150, 2004.
- [3] J. D. Glover, M.S. sarma and T.J. Overbye, "Power Flows," in *Power Systems Analysis & Design*, CENGAGE Learning, pp. 295-366, 2012.
- [4] E. J. Jiménez Tenorio, "Incorporación de modelos WECS a un programa de flujos dinámicos de potencia," Tesis Maestría, UNAM, México, D.F., 2013.
- [5] J. J. Grainger, W.D. Stevenson Jr, "Soluciones de flujos de potencia," in *Análisis de sistemas de potencia*, McGraw-Hill, México, 1996.

- [6] M. Soshinskaya, W.H.J. Crijns-Graus, J.M. Guerrero, J.C. Vasquez, "Microgrids: Experiences, barriers and success factors," *Renewable and Sustainable Energy Reviews*, vol. 40, pp. 659-672, Ago. 2014.
- [7] Siemens AC, "Microgrids white paper," Siemens AC, Munich, Germany, 2011.
- [8] A. Rodríguez, "La generación distribuida y su posible integración al sistema interconectado nacional," *Taller sobre Generación Distribuida*, CREG, 2009.
- [9] C. E. Lang, "Generación distribuida, energía de calidad," [Online]. *CONUEE*, 2013. http://www.conuee.gob.mx/wb/CONAE/generacion_distribuida_1?page=1. Consulta: 18 de mayo del 2016.
- [10] A. J. R. Martínez-Rangel, "Análisis y diseño de una micro red para el Campus Siglo XXI de la Universidad Autónoma de Zacatecas," Tesis de Maestría, Unidad Académica de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas, Zacatecas, a presentarse en noviembre de 2016.

Diseño de estructuras para vehículos aéreos no tripulados

M. García-Sánchez^a, J. Flores-Troncoso^a, S. Ibarra-Delgado^a, R. Sandoval-Arechiga^a, F. Ortega-García^b

^aCentro de Investigación y Desarrollo en Telecomunicaciones Espaciales

^cUniversidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica.

Av. López Velarde 801, Col. Centro, Zacatecas, Zac., México, 98000.

mi_garcias@uaz.edu.mx, jflorest@uaz.edu.mx, salvador6407@hotmail.com,

rem.sandoval@gmail.com, francisko_7rock@hotmail.com

2016 Published by DIFU_{100ci}@ <http://difu100cia.uaz.edu.mx>

Resumen

En este trabajo se muestra el diseño y construcción de una estructura para vehículos aéreos no tripulados (VANT), como parte del desarrollo de trabajos de investigación en fotogrametría y estabilización en vehículos aéreos no tripulados.

Palabras clave: Drones, UAVs, VANTs, Diseño estructural.

1. Introducción

El sueño de volar. Los iniciadores habían estado volando globos y dirigibles desde 1783, pero un vuelo con algo más pesado que el aire "fue visto como la búsqueda de charlatanes", dijo Jim Tobin, autor de conquistar el aire: Los hermanos Wright y la Gran Carrera de Volar. "La idea del vuelo humano era el estándar imposible. Sin embargo, el sueño era demasiado convincente como para ignorarlo. Ya que en 1804, el inglés George Cayley identificó tres áreas de estudio para futuras investigación: la necesidad de conseguir despegar de la tierra, la propulsión para moverse una vez en el aire, y el control para permanecer en el aire[1].

Los primeros vuelos- En toda la historia de la invención probablemente no hay sin igual en la manera ostentosa en la que los hermanos Wright, de Dayton, Ohio, que dieron paso al mundo su invención en la primera

época exitosa de la máquina voladora. Su logro marcó un enorme paso hacia adelante en el arte de volar, que no era muy conocido, y era tan brillante que la duda en cuanto a que la verdad de la historia fue oculta; sobre todo porque los inventores se negaron, ya sea para conocer sobre la máquina o para hacer cualquier declaración sobre sus amplios detalles. El Científico Americano, sin embargo, escribió a sus diecisiete testigos oculares que habían visto los diferentes vuelos y que recibieron estas cartas de estos distinguidos residentes locales comentando el logro que habían alcanzado mas por encima de la duda que inicialmente se tenía. Desafortunadamente, el mundo aeronáutico extranjero no le dio la importancia a los hechos de este modo, que se dieron a conocer; y cuando Alberto Santos-Dumont hizo su vuelo durante algunos cientos de pies en Europa, le dieron el crédito por haber hecho el primer vuelo con éxito[2].

Alternamente al desarrollo de vehículos aéreos de

ala fija, en la última mitad del siglo XIX muchos inventores se enfocaron en el desarrollo de helicópteros. Este fue un progreso práctico, pero no un vehículo exitoso. El problema fue que dejaban de lado qué el vehículo fuera barato, creíble y liviano. Se sabe que existieron un sin número de intentos de uso de la máquina de vapor. W.H. Phillips (Inglaterra, 1842) fue quien construyera el modelo de la máquina de vapor. Viscomte Gustave de Ponton d'Amercourt (Francia, 1863) construyó un modelo pequeño de la máquina de vapor, él inventó la palabra "helicóptero". Aphonse Penaud (Francia, 1970) Experimentó con estos modelos. Enrico Forlanini (Italia, 1978) construyó una máquina voladora con un peso de 3.5 kg, está era accionada por una máquina de vapor. Thomas Edison (Estados Unidos, 1880), experimentó con estos modelos. Él reconoció que el problema era que se estaba omitiendo la máquina adecuada. Edison concluyó que no sería capaz de volar hasta que existiera una máquina con una capacidad de 1 o 2 kg/hp. Estos fueron los modelos que daban el inicio de la búsqueda del problema de adecuar un suministro de energía sostenible para el vuelo[3].

Alrededor de 1910 cuando la carrera de la aviación parecía ser muy promisor, el ruso Igor Ivánovich Sikorsky sucede el hecho que desde su niñez a principios de esa década había estado influenciado por los modelos de máquinas voladoras plasmados en los diseños de Leonardo da Vinci. Para abril de 1909, Sikorsky había estudiado docenas de planos de aviones que había encontrado en París, en esos momentos fascinado, él creía convincentemente en la esperanza del concepto del "helicóptero". En ese tiempo el conoció a los pioneros de la aviación, los capitanes Ferdinand Feber y Louis Bleriot, siendo el primero quien advirtió a Sikorsky lo siguiente "No desperdices tu tiempo en helicópteros, el avión será mas valorado", Sikorsky entonces compró una máquina Anzani de 25 caballos de potencia, luego regreso a Kiev en Ucrania donde construyó su primer modelo de hélice no tripulado, nombrado H1. Después de una mejora a su prototipo, desarrolló el H2, pero por alguna razón decidió enfocarse a aeroplanos[4].

Desde 1929 Sikorsky retomó su idea inicial y diez años después ya tenía su primer prototipo nombrado el VS-300, un helicóptero capaz de ser controlado por la persona que lo tripulaba, marcando un tiempo record de vuelo de 90 minutos, al observar la filmación del staff y el movimiento percibido de la aeronave observó algunos problemas de control que enseguida corrigió[5].

Hoy en día tenemos distintos tipos de artefactos voladores en tamaño y en peso, impulsados por combustible o por baterías siendo estos últimos los que generalmente son conocidos como vehículos aéreos no tripulados

o drones. Son muchas las preocupaciones para poder mantener en vuelo por determinados periodos de tiempo a estos vehículos. En este documentos se busca mostrar el diseño y construcción de una estructura para un helicóptero de cuatro motores, comúnmente conocido como cuadrotor o dron¹. Con el objetivo de minimizar su peso y además optimizar su funcionalidad para contener dispositivos necesarios para el control de vuelo.

2. Estructuras y materiales

El desarrollo más notable en las estructuras de los vehículos aéreos será la reducción de tamaño (miniaturización) y el uso de materiales multifuncionales. Aunque existen muchos avances en materiales de manufactura que están permitiendo la accesibilidad a los vehículos aéreos, no todos los beneficios son exclusivos de estos. La próxima generación de estas aeronaves requerirán de materiales livianos y de bajo costo. El diseño y construcción de un vehículo es direccionado por las consideraciones de un rango de modos de fallas, tales como excesiva deformación elástica, flexiones, ondulaciones, fracturas, fatigas, corrosión o daño por impacto. Se han identificado cuatro áreas que serán esenciales para futuros desarrollos en vehículos aéreos no tripulados. Estas cuatro áreas requerirán de investigación y desarrollo básica hasta la realización de pruebas. A continuación observamos estas cuatro áreas:

- Definición del diseño del entorno en el cual el futuro VANT operará, incluyendo definición de carga, requerimientos de confiabilidad y aeroelasticidad.
- Reducción de costos de manufactura para los componentes de la aéreo-estructura, incluyendo avances en materiales compuestos y materiales multifuncionales.
- Mejoramiento del proceso de diseño para soportar la reducción del tiempo de ciclo, prototipos rápidos y bajos costos de fabricación.
- Monitoreo del buen funcionamiento, controles no vedosos, detección de tecnologías, incluyendo sistemas microelectromecánicos, nuevos sensores y actuadores.

Cada uno de estas áreas requerirá de una mejor comprensión del proceso y fenómenos implicados, así como una predicción más confiable de interacciones entre los elementos del proceso o dispositivo[6].

¹Del origen griego, **dhren**, imitación de sonido producido por un abejorro o abeja zángano

3. Proceso de diseño

Los vehículos aéreos no tripulados requieren de una estructura para distribuir sobre esta los componentes necesarios para la operación del aeronave. Así mismo conforman el soporte principal de la aeronave llamado también armadura.

La armadura es uno de los principales tipos de estructuras que se usan en la ingeniería. Esta proporciona una solución práctica y económica para muchas situaciones de ingeniería, en especial para el diseño de puentes y edificios. La mayoría de las estructuras reales están hechas a partir de varias armaduras unidas entre sí para formar una armadura espacial. Cada armadura está diseñada para soportar aquellas cargas que actúan en su plano y, por tanto, pueden ser tratadas como estructuras bidimensionales. Los elementos de una armadura, por lo general, son delgados y sólo pueden soportar cargas laterales pequeñas; por eso todas las cargas deben estar aplicadas en los nodos y no sobre los elementos. Las armaduras rígidas no siempre son simples, incluso cuando parecen estar hechas de triángulos. El procedimiento de agregar elementos a una armadura en el que la resultante será rígida cada vez que se agreguen nuevos elementos, éstos se unen a nodos ya existentes y además se conectan entre sí en un nuevo nodo. Una armadura que se construye de esta forma recibe el nombre de armadura simple. Cuando varios elementos rectos se unen en sus extremos para formar una configuración tridimensional, la estructura obtenida recibe el nombre de armadura en el espacio o espacial[7].

Los brazos del cuadricoptero son de las partes más importantes, ya que estos son extensiones de la estructura y además en ellos van ensamblados los motores, estos deben de ser lo suficientemente resistentes, soportar la torsión de la carga, al mismo tiempo ligeros y aerodinámicos para no tener problemas de estabilidad al momento de elevar el cuadricoptero.

La dimensión promedio para que un cuadricoptero maniobre en forma eficiente es de 450 milímetros, aunque existen medidas que redundan alrededor de esta, la mayoría de la literatura refiere a la ya mencionada medida[8][9]. Con esta definición estándar en la que las medidas se toman del eje de giro de cada motor hacia el motor del lado opuesto, se procede a diseñar la estructura en forma de "X", aunque existen otras configuraciones para cuatro motores, como lo son "+" o "V" se procedió con la selección ya que es un diseño muy común.

3.1. Diseño

Con la ayuda del software Sketchup² creamos los modelos de los brazos para que posteriormente fueran generados en una impresora 3D. A continuación se muestran las figuras 1 y 2, en las que se puede apreciar el diseño obtenido. Se puede observar como se han dejado espacios semi-ocultos bajo la estructura en los que se pueden adaptar dispositivos como pueden ser los controladores de velocidad.

4. Diseño del sistema

El diseño de la estructura permite proteger los cables de alimentación de los motores, quedando de igual forma que el controlador de velocidad. El diseño de la cara superior esta construido de forma que perita el flujo de aire, minimizando la fricción entre los nodos de la estructura.

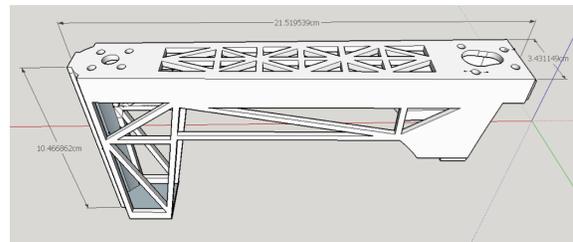


Figura 1. Modelo de brazo vista lateral

Inicialmente se busca una funcionalidad en el diseño, la siguiente parte es la elección del material que se usará en la construcción, en este caso se eligió un polímero termoplástico del tipo ABS, el cuál tiene un módulo de Young en un rango de 1.1 y 2.9 KPa, con este material se busca tener una vibración mínima en la estructura[10]. Esta vibración es una resonancia natural de frecuencia que se provoca por el funcionamiento de los motores y se calcula con la siguiente formula:[11].

$$f = \sqrt{(3EI/L^3 M)}/2\pi \text{ Donde:}$$

f es la frecuencia natural. E es el módulo de Young. I es el momento de inercia de la sección. M es la masa del sistema de motor. L es la longitud del brazo.

²Software de versión de pruebas de la empresa google®.

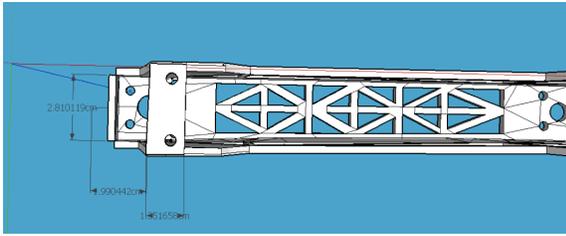


Figura 2. Modelo de brazo vista inferior

La construcción de una estructura propia podrá ser implementada en los prototipos ya existentes (ver figura 3), y será compatible en su totalidad para colocar todos los dispositivos requeridos por el vehículo.



Figura 3. Prototipo de pruebas armado

4.1. Conclusiones

En el presente trabajo se ha mostrado el diseño y construcción de una estructura para dron, en este sentido, se puede observar como se pueden obtener estructuras más adaptadas a las funcionalidades específicas de cada proyecto de investigación en particular. Con lo anterior se puede referir que tendremos mas protección en los elementos que puede contener el vehículo aéreo no tripulado. De esta forma podemos dejar un esquema básico para el diseño adaptable para futuras investigaciones.

Referencias

- [1] Curry, A. 135, 8. TAKING FLIGHT, pp. 135, 2003.
- [2] 50, 100 & 150 Years Ago, Scientific American, Vol. 295, 2006.

- [3] J. Johnson *et al*, "HELICOPTER THEORY". Dover Publications, Inc., 1980.
- [4] I. Sikorsky , "The Sikorsky Legacy". Arcadia Publishing, 2007.
- [5] K. Moore , Sikorsky's blurred vision, Professional Engineering, pp, pp.63, 2001.
- [6] Uninhabited Air Vehicles: Enabling Science for Military Systems, Publication NMAB, pp. 50-51, 2000.
- [7] Beer, F., E. Jhonston y E. Eisenberg , Mecánica vecyorial para ingenieros: estática, Mc Graw Hill, 8^o Ed., 2007.
- [8] "Drones - ACG Drone", ACG Drone, 2016. [Online]. Available: <http://www.acgdrone.com/drones/>. [Accesado: 19- jul- 2016].
- [9] "Cuadrícóptero hechos a mano", Horizontech.ru, 2016. [Online]. Available: <http://www.horizontech.ru/ru/uav/poleznye-stati/54-rama-dlya-kvadrokoptera-svoimi-rukami>. [Accesado: 19- jul- 2016].
- [10] Materials data book, Cambridge University Engineering Department, 2003.
- [11] Fernandez, A., Basic structural dynamics in multicopters, Aerospace Engineer at AIRBUS, 2014.

Implementación en Hardware de un Entrelazador/Desentrelazador de Datos con Interface AXI Stream

Laura García Luciano, Ian Camacho Pichardo, Salvador Ibarra Delgado, Jorge Flores Troncoso, Remberto Sandoval Árechiga

Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica.
Av. López Velarde 801, Col. Centro, Zacatecas, Zac., México, 98000.

35162643@uaz.edu.mx, ian.cam57@gmail.com, sibarra@uaz.edu.mx, jflorest@uaz.edu.mx, rem.sandoval@gmail.com.

2016 Published by *DIFU*_{100ci} @ <http://difu100cia.uaz.edu.mx>

Resumen

En éste artículo se muestra la implementación de un IP Core de un algoritmo de dispersión de errores llamado “Entrelazador”, y de su decodificador llamado “Desentrelazador”. Un IP Core (Núcleo de Propiedad Intelectual) es un módulo de hardware de aplicación específica, el cual es encapsulado para ser usado como dispositivo en algún sistema que lo necesite. Específicamente hablando, éste entrelazador/desentrelazador como unidad lógica es totalmente funcional y rehusable, pues al estar empaquetado puede fácilmente ser colocado en cualquier sistema que lo requiera. En un sistema de comunicaciones que consta de un transmisor y un receptor, existen distintas fases por las cuales viajan los mensajes, cada uno dividido en tramas. El algoritmo estudiado dispersa los errores en el mensaje para su fácil supresión en etapas posteriores. Éste IP Core convive con otros IP Cores que tienen distintas funciones en un sistema de comunicaciones, por lo que la salida del entrelazado es la entrada que alimenta otro proceso, o dicho de otro modo, a otro IP Core. Entonces, es necesario contar con un modo estándar que hará que se comunique el IP Core de Entrelazado/Desentrelazado con la cadena del transmisor y receptor respectivamente. Razón por la cual se hace uso del protocolo AXI, permitiendo la intercomunicación con el IP Core que esté antes en la cadena y con aquel que esté después de la cadena.

Palabras clave: IP Core, Entrelazado, Desentrelazado, FPGA, AXI Stream.

1. Introducción

En un sistema de comunicaciones que se compone básicamente de un transmisor, un canal y un receptor, existen módulos específicos que tratan la información desde su origen analógico/digital hasta ser transmitida por un canal y después al ser recibida

existen también módulos que corresponden a los procesos inversos según corresponda y obtener el mensaje original, o la información original analógica/digital.

Casi todo el tiempo las perturbaciones están presentes, el ruido por ejemplo, y dependiendo de algunas condiciones, a veces se presentan ráfagas de errores. Para ésto, hay una etapa en el transmisor que se encarga de

disipar los errores en ráfagas llamada “**Entrelazado**” y así, otro módulo se encarga de corregir los errores de forma eficiente.

El algoritmo de entrelazado en términos generales, revuelve o coloca en posiciones distintas conjuntos de bits de tamaño previamente establecido, y es por ello que los errores quedan separados entre sí. Entonces, el entrelazado del mensaje antes de la transmisión y el desentrelazado después de la recepción causa que una ráfaga de errores sea dispersada y se convierta en errores aleatorios como lo muestran en [1].

Existen dos tipos de algoritmos de entrelazado, uno es el entrelazado de bloque y el otro es el entrelazado convolucional. Para [2], un entrelazador convolucional es en la mayoría de casos, implementado en hardware por su gran popularidad y simpleza, en [3] se muestra un entrelazador convolucional que se usa para el análisis de modelos.

Un “*Entrelazado de bloques*” acepta símbolos codificados en bloques, los permuta y su salida alimenta al módulo que le sigue en la cadena que conforma un transmisor, el modulador. En el receptor, el “*Desentrelazador de bloques*” realiza la operación inversa, es decir, acepta los símbolos desde el demodulador, los procesa o reacomoda y alimenta a los módulos que le siguen para obtener el mensaje original.

Un aspecto muy importante en el desarrollo del proyecto, es el protocolo AXI que se usa para poder comunicar el entrelazador/desentrelazador con la memoria de la tarjeta de desarrollo y así poder comprobar su funcionamiento. El protocolo AXI Stream es utilizado para intercomunicar de forma rápida y eficiente la memoria externa por medio del DMA(Direct Acces Memory) con los IP Cores desarrollados.

2. Entrelazado / Desentrelazado

El entrelazador de bloques implementado, permuta símbolos de tamaño de un byte que conforman el mensaje. El IP Core de entrelazado recibe como entrada 256 bytes, mismos que guarda en memoria como un arreglo de M-renglones x N-columnas. Como primera instancia el IP Core intercambia dos columnas entre sí a conveniencia, y luego los muestra en salida por columnas hasta terminar con los 256 bytes y vuelve a comenzar mientras exista información a procesar. El desentrelazador, recibirá entonces los 256 bytes permutados y con su algoritmo inverso se obtendrá la información original.

Para ejemplificar el proceso de acomodo de un mensaje, se tiene una matriz de M=2 y N=4 y se intercambiarán las columnas 2 y 3 mostradas en la figura 1

a	b	c	d
e	f	g	h

Figura 1. Matriz ejemplo de 2 x 4

Ahora, el algoritmo de entrelazado queda reflejado en la ecuación ec.(1):

$$renglones(k - 1) + i - 1 = columnas(renglones - i) + j - 1 \tag{1}$$

Entonces, cuando el mensaje se almacenan en la memoria, se realiza por renglones seguidos, en éste ejemplo la memoria queda llena como se muestra en las figuras 2 y 3.

a	b	c	d	e	f	g	h
---	---	---	---	---	---	---	---

Figura 2. Contenido de una matriz de 2 x 4 acomodado en una memoria de 8 símbolos de 8 bytes cada uno.

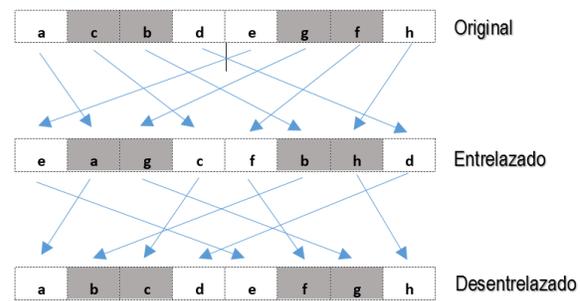


Figura 3. Proceso de Entrelazado y Desentrelazado

Para el desarrollo de éste proyecto, el entrelazador/desentrelazador recibe paquetes de 256 bytes y devuelve el mismo tamaño del paquete a los procesos siguientes. El valor de M y N están ligados al tamaño de la memoria donde se almacenará, por tanto los valores multiplicados en conjunto deberán dar como resultado 256, por ejemplo (32)(8) = 256.

3. Bus AXI

3.1. Origen

El AXI(Advance eXtensible Interface) es un protocolo que determina una interfaz de conexión entre IP Cores (Núcleos de Propiedad Intelectual). Un bus AXI, es un protocolo de bus que soporta fases separadas de dirección/control y datos, transferencias de datos no alineados usando un flash de byte, transacciones basadas en ráfagas con sólo emitir la dirección de inicio, separa los canales de datos de escritura y lectura para habilita el bajo costo del DMA, habilidad para emitir para

emitir múltiples direcciones de salida, sin chocar con la ejecución de las transacciones, y fácil unión de las etapas.

3.2. Protocolos de bus on-chip AMBA AXI

Los protocolos ARM (*Advanced Risk Machine*) AMBA (*Advanced Microcontroller Bus Architecture*) son un estándar para comunicación *on-chip*. AMBA es un estándar abierto para la administración y conexión de bloques funcionales en un *System-on-chip* (SoC). Facilita el desarrollo de diseños multi-procesador con un gran número de controladores y periféricos.

AXI *Advanced eXtensible Interface* es parte de ARM AMBA, una familia de buses para microcontroladores introducido por primera vez en 1996. La primera versión de AXI fue incluida en AMBA 3.0, lanzado en 2003. AMBA 4.0, lanzado en 2010, incluye la segunda versión de AXI, AXI4 [9].

Hay tres tipos de interfaces AXI4:

- **AXI4**: para requerimientos de mapeo de memoria de alto rendimiento.
- **AXI4 Lite**: para comunicación de mapeo de memoria simple o de bajo rendimiento (por ejemplo, desde y a registros de estado y control).
- **AXI4 Stream**: para transmisión a alta velocidad de datos.

3.3. AXI Stream

El protocolo AXI4 Stream actúa como un canal unidireccional para un flujo de datos con un *handshake* entre 2 y 9 señales. Lo que quiere decir que la información o datos son transmitidos de maestro a esclavo [7]. Para el caso más simple con un *handshake* entre dos señales, se requieren las señales:

- **TDATA**: El vector de datos.
- **TVALID**: Indica que los datos están presentes y son válidos para el esclavo.

Pero en la mayoría de los casos se utiliza una señal adicional:

- **TREADY**: Bandera presentada por el esclavo para indicar que está listo para recibir los datos.

Es posible prescindir de la señal TREADY y en este caso, se toma en cuenta que la transmisión es sin confirmación del esclavo.

Las otras señales que pueden ser utilizadas en la transferencia son:

- **TID**: Identifica la transmisión de la trama actual de datos(en caso de que diferentes tramas sean transmitidas entrelazadas).
- **TDEST**: Funciona como una dirección de destino en el caso de varios esclavos o diferentes destinos en el esclavo.
- **TLAST**: Indica el final de una trama, útil cuando las tramas varían en tamaño.
- **TKEEP**: Para especificar si todos los datos transmitidos en la trama deben tomarse en cuenta o no.
- **TSTRB**: Indica si los datos en cierta posición son útiles. El byte de posicionamiento puede darse el lujo de no actualizar cierta parte de la información que posee el esclavo.
- **TUSER**: Datos en paralelo a la información principal que pueden incluir información adicional para el esclavo.

Considerando el caso más simple, un ejemplo de una transacción con las tres señales básicas se representa en la figura 2.7. La información es transferida cuando TVALID y TREADY se encuentran en estado alto (*handshake*).

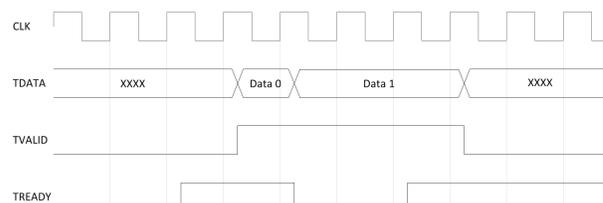


Figura 4. Transferencia de datos mediante protocolo AXI4 Stream. Imagen obtenida de [8].

4. Implementación del IP Core en Hardware y Software

La implementación de un IP Core se conforma de dos etapas que trabajan en conjunto. En la primera etapa, cuando se tiene un diseño funcional, se codifica en un lenguaje descriptor de hardware en algún software de desarrollo, que ya finalizado se empaqueta de manera que el IP Core está listo para añadirse al sistema que lo requiera. Además, se necesita crear una plataforma de hardware, que contendrá las conexiones entre el SoC Zynq 7020, el IP Core empaquetado y la comunicación AXI Stream para el rápido y eficiente acceso a memoria externa.

En la segunda etapa, con la ayuda de una herramienta de desarrollo de software se puede acceder y manipular el hardware, puesto que ahora ocupan de forma individual, una dirección de memoria. Finalmente, cuando se programe la tarjeta de desarrollo con el hardware y software construido, podremos controlar las acciones que se deseen desde la PC mediante el puerto serie.

4.1. Implementación en Hardware

Hablar de una implementación en hardware implica conocer los recursos disponibles. Se necesita de un software en el cual se pueda describir el hardware y los conocimientos del lenguaje descriptor a usar. También se debe contar con un FPGA que tenga la capacidad de albergar la aplicación, en éste caso el IP Core de Entrelazado/Desentrelazado. Es entonces que la implementación puede realizarse, desde que se diseña el algoritmo, se presenta como un IP Core, se descarga al dispositivo lógico programable, hasta la manipulación del mismo a través de una terminal.

4.1.1. Descripción del Hardware

La descripción del hardware se llevó a cabo con el uso del lenguaje verilog usando la herramienta de diseño electrónico ISE Design Suite 14.7 de Xilinx. En ésta última se plasmó el diseño del Entrelazador/Desentrelazador de datos, que muestra el esquemático de la figura 5.

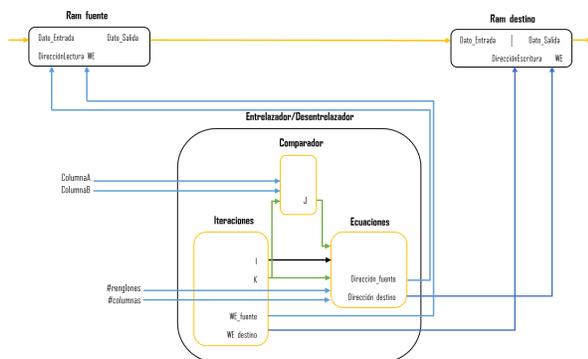


Figura 5. Esquemático del algoritmo de Entrelazado/Desentrelazado.

El algoritmo de entrelazado se forma de los siguientes componentes:

1. Aquel que intercambia las columnas, llamado "intercambia_columnas".
2. Aquel que calcula la dirección de lectura y escritura de una memoria origen, llamado "calculo_direccion".

3. Aquel que controla el inicio y fin del proceso de entrelazado/desentrelazado, llamado "FSM_Interleaver".

Sin embargo, el algoritmo no trabaja por si solo, necesita una fuente de donde tomar los datos que va a procesar, a alguien que le diga en que momento hacerlo, también necesita saber donde colocar los datos procesados y avisar que dichos datos están listos; y es por ello que la arquitectura se compone del correspondiente algoritmo y a su vez de dos memorias tipo RAM de doble puerto. Una memoria contendrá los bytes que procesará el algoritmo, y la segunda memoria contendrá los bytes conmutados (como se vió en la figura 5).

intercambia_columnas Éste componente se encarga de intercambiar las columnas escogidas, de modo que recibe como entrada un contador que recorre todas las columnas, y cuando ese contador es igual a COL_A la salida $j = COL_B$ ó cuando es igual a COL_B la salida $j = COL_A$; pero cuando es diferente a esas dos columnas, la salida j es igual a la entrada j_{col} .

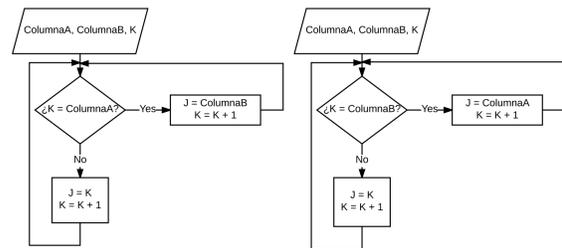


Figura 6. Toma de decisiones para intercambiar las columnas.

calculo_direccion Éste componente contiene una ecuación (ver 1) que representa la relación entre la dirección de lectura de una memoria origen, y la dirección de escritura de una memoria destino. La ecuación necesita saber el número de columnas COL y renglones REN totales; dos contadores, uno que recorra los renglones i_{ren} y otro que recorra las columnas j_{col} y un indicador que muestre si se está recorriendo una columna intercambiada.

FSM_Interleaver Éste componente se encarga dirigir a los dos últimos, así como proporcionarles las cuentas que recorren renglones y columnas, habilitar/deshabilitar la escritura en una memoria destino y avisar que el proceso del Interleaver terminó. Tiene como entradas una señal de reloj clk , una señal de reset rst , una señal de comienzo $start$, una señal de comenzar un nuevo proceso después de haber terminado uno anteriormente $restart$ y el número de columnas COL y renglones REN de la

matriz de 256 elementos. Las salidas, como se mencionaba anteriormente, son las cuentas para los renglones i_{ren} y para las columnas j_{col} , una señal que se conectará al WE de una memoria destino para habilitar/deshabilitar la escritura en ella, así como el fin de su proceso indicado con $done = 0$.

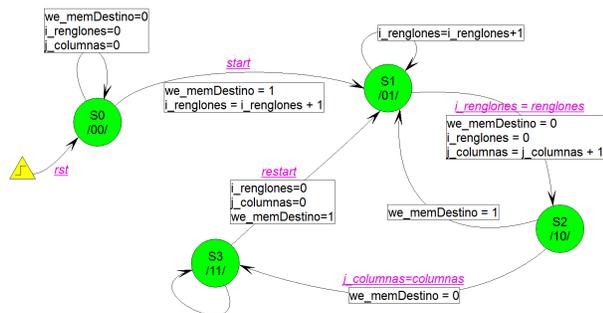


Figura 7. Máquina de estados que contiene el algoritmo de entrelazado/desentrelazado.

Cuando finalmente se tiene todo el sistema descrito en verilog, es posible crear el IP Core con ayuda de otra herramienta e incorporarlo en la plataforma de hardware para que pueda ser alimentado y probado en solitario.

4.1.2. Plataforma de Hardware

La plataforma de hardware se desarrolla en la herramienta de software Vivado 2016.1 de Xilinx, y se compone del System on Chip (SoC) Zynq 7020 que es el procesador del FPGA, el IP Core de Entrelazado, el IP Core de Desentrelazado y del protocolo de comunicación AXI Stream que hace posible la interacción con los datos guardados en memoria y entre los Cores (ver Figura 8). Una forma de poder apreciar de mejor manera el desarrollo de la plataforma, es tener al sistema en bloques como lo maneja Vivado por defecto, para así ver claramente las conexiones entre los mismos y detectar errores casi de forma inmediata.

En la Figura 8) se identifican perfectamente las partes necesarias para arrancar nuestro sistema de Entrelazado y Desentrelazado.

1. El *Zynq7ProcessingSystem* es quien contiene el sistema de procesamiento, los controladores de memoria, los protocolos de comunicación.
2. El *Interleaver_AXI2* es el IP Core que trabaja como Entrelazador con interfaz AXI.
3. El *Interleaver_AXI3* es el IP Core que trabaja como Desentrelazador con interfaz AXI.
4. El *AXIDirectMemoryAccess* es quien interactúa directamente con la memoria sin la intervención del

procesador, haciendo eficiente en tiempo la lectura y escritura de datos en el memoria.

Finalmente se encuentran bloques como el *reset* activo en bajo y el reloj *clk*, que alimentan a todo el sistema.

4.2. Plataforma de Software

La plataforma de software es desarrollada en la herramienta de software *Xilinx Software Development Kit 2016.1* usando el lenguaje de alto nivel C para realizar la interacción con el hardware a través de una terminal de tipo serial (ver Figura 9).

Desde ésta plataforma se configura la comunicación serial para así interactuar con el software e indicarle mediante el teclado en que momento transmitir entre otras configuraciones.

5. Resultados

El primer paso para observar el funcionamiento del algoritmo implementado es simular desde el más mínimo bloque en el diseño y comprobar la salida esperada. Por tanto se proseguirá mostrando los resultados en la implementación de hardware (ver Figura 8).

Mediante la herramienta de software SDK, se descargó la plataforma de hardware en la tarjeta de desarrollo ZedBoard (ver Figura 10).

El proceso en general consiste en entrelazar una cadena de letras (cada letra representa 1 byte=8 bits) guardadas en la memoria de la tarjeta de desarrollo y desentrelazar la cadena conmutada, para después guardarla en la misma memoria, y obtener de un solo paso la correcta salida que verifica la adecuada funcionalidad de los dos IP Cores respectivamente.

Una terminal serial fue usada para que se determinara qué hacer y cuándo en el sistema de Entrelazado/Desentrelazado de datos. En la plataforma de software se definieron distintos comandos para dicho fin y los siguientes son los que se usan por defecto:

comando e Mediante éste comando se carga el contenido de un archivo a memoria (ABCDEFGHABCDEFGH...).

comando b Mediante éste comando se define el tamaño en bytes del paquete que se va a procesar, para nuestro caso se define en 256.

comando m Mediante éste comando se puede desplegar el contenido de un segmento de la memoria (memoria_origen [0]) del cual se alimentará el IP

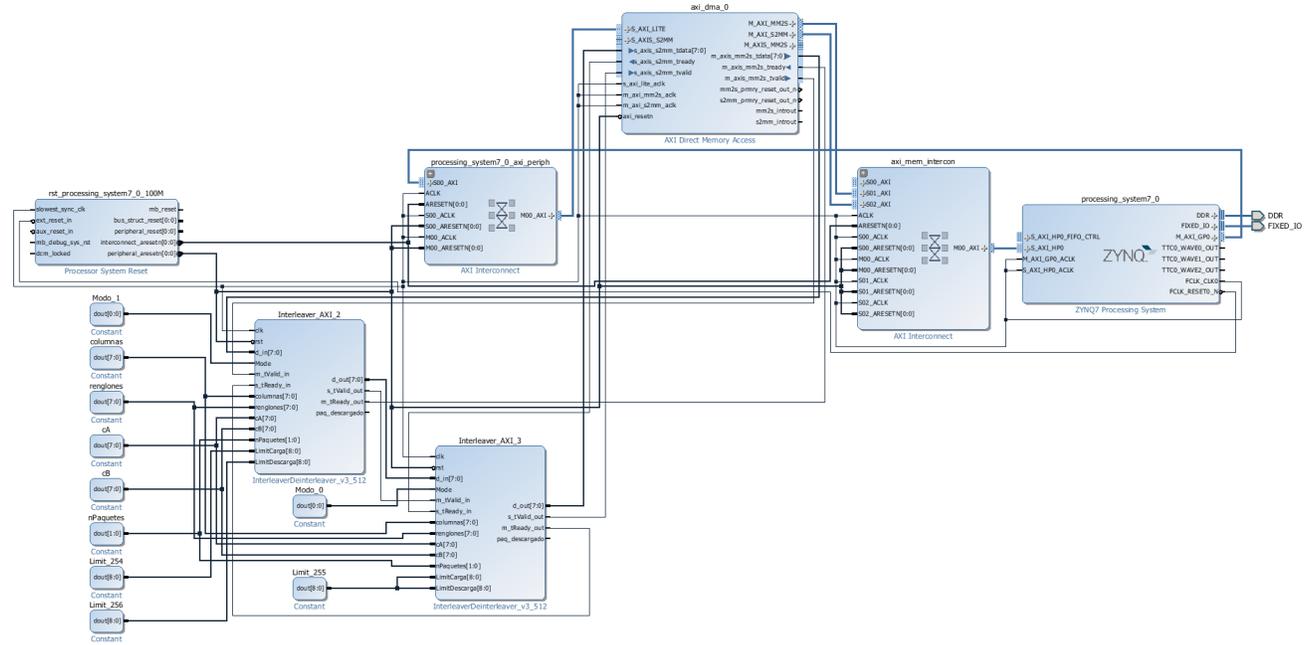


Figura 8. Diagrama de Bloques de la Plataforma de Hardware

```

// ***** Banco de Prueba de IPcores que usan AXI Stream *****
--> Bytes = 256
--> COMANDOS: e(nviar->0), r(ecibir->0), m(memoria), l(impiar), t(transferir), b(bytes)
--> m
--> Zona: o(rgen), d(estino) --> o
--> Número de líneas (75 bytes cada una) a mostrar: 10
Dirección:          Data:
1100000 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100040 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100080 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100120 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100160 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100200 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100240 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100280 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH

```

Figura 9. Código en la Plataforma de Software Vivado SDK

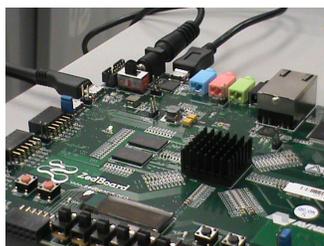


Figura 10. Tarjeta de desarrollo ZedBoard

Core de Entrelazado, y otro segmento de la memoria(memoria_destino [d]) que contendrá el paquete procesado.

comando t Mediante éste comando se realiza la transmisión de la infomación.

Al cargar el contenido del archivo a memoria y definir los 256 bytes se puede teclear el comando t para iniciar la transmisión, que de ser correcta, se podrá revisar la memoria destino y observar el resultado. En la Figura 11 se observa el procedimiento antes descrito y también que la cadena original(en memoria origen) es igual a la cadena obtenida(en memoria destino), demostrando que el procesamiento fue exitoso.

```

***** Banco de Prueba de IPcores que usan AXI Stream *****
--> Bytes = 256
--> COMANDOS: e(nviar->0), r(ecibir->0), m(memoria), l(impiar), t(transferir), b(bytes)
--> m
--> Zona: o(rgen), d(estino) --> o
--> Número de líneas (75 bytes cada una) a mostrar: 10
Dirección:          Data:
1100000 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100040 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100080 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100120 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100160 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100200 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100240 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH
1100280 --> ABCDEF GHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGHABCEFGH

```

Figura 11. Terminal que muestra el contenido de la memoria origen y destino.

Sin embargo, también se puede observar lo que pasa en tiempo de ejecución directamente en el SoC que la infomación sea la que arroja la terminal, y es por ello que muestro la carga de bytes desde memoria a

la memoria RAM dualPort del IP Core Entrelazador(ver Figura 12).

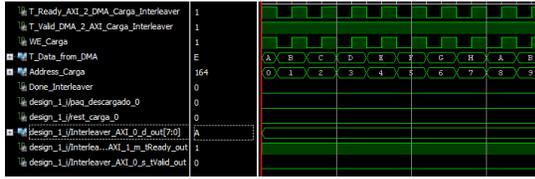


Figura 12. Carga de bytes en el IP Core de Entrelazado

Y de igual manera, como el proceso de entrelazado es transparente, debido a que sólo se muestra el resultado final, en la siguiente figura(ver Figura 13) se encuentra parte de los bytes entrelazados.

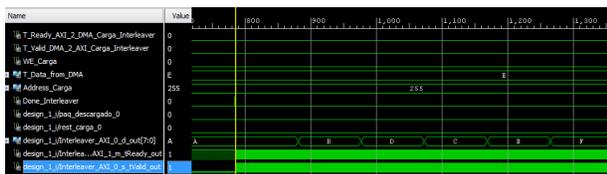


Figura 13. Salida IP Core Entrelazado

6. Conclusiones

Con los recursos de software y hardware, fue posible desarrollar la implementación en hardware del IP Core de Entrelazado/Desentrelazado. El Zynq, tiene la capacidad requerida para contener todo un sistema de telecomunicaciones, además, gracias al procesador y por facilidad, se pudo manipular el IP Core desde la PC mediante una terminal de tipo serial.

En cuanto a las herramientas de software usadas, se puede decir que cuentan con las funciones necesarias y más, en Vivado por ejemplo, su interfaz gráfica nos permitió hacer todo tipo de pruebas en el menor tiempo posible, haciendo una ventaja potencial por disminuir el tiempo de desarrollo.

Finalmente fue posible verificar que la comunicación entre IP Cores fue exitosa pues con ayuda de la herramienta “debug” llamada ChipScope, la información pasa de uno al otro sin pérdidas y en el momento requerido, así como los resultados esperados.

Referencias

- [1] Sklar, Bernard “*Digital Communications. Fundamentals and Applications*”. Prentice Hall, 2001.
- [2] Abdelmohsen, A. Khater Mohamed M, Khairy and S.E.D, Habib. “*Efficient FPGA Implementation for the IEEE 802.16e Interleaver*”. 2009 International Conference on Microelectronics.
- [3] Upadhyaya, BK and Sanyal, SK. “VHDL modeling of convolutional interleaver-deinterleaver for efficient FPGA implementation”. International Journal of Recent Trends in Engineering, Academy Publisher, Finland. Vol. 2, Num. 6, p. 66-68, 2009.
- [4] Andrews, Kenneth Heegard, Chris and Kozen, Dexter. “A Theory of Interleavers”. Cornell University, 1997.
- [5] Eric Tell and Dake Liu. “A HARDWARE ARCHITECTURE FOR A MULTI MODE BLOCK INTERLEAVER”.
- [6] VanCourt, Tom and Herbordt, Martin “Application-Specific Memory Interleaving for FPGA-Based Grid Computations: A General Design Technique, “*Field Programmable Logic and Applications, 2006. FPL’06. International Conference on*”. IEEE, p.1-7,2006.
- [7] Toussaint, Cedric y Benharbone, William “AXI4-Stream” https://wiki.electronicens.cnrs.fr/index.php/FPGA_CPLD_Guides:_AXI4-Stream 2015 Consultado Enero, 2016
- [8] Analytics Engines “Introduction to AXI” <http://www.analyticsengines.com/developer-blog/introduction-to-axi/> 2015 Consultado Enero, 2016
- [9] Xilinx, Inc AXI Reference Guide http://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf 2015 Consultado Enero, 2016

Modelado Matemático y Control PD de un Quadrotor

H. Yáñez-Badillo^a, R. Tapia-Olvera^b, F. Beltrán-Carbajal^c, O. Aguilar-Mejía^a, I. López-García^c

^aUniversidad Politécnica de Tulancingo, Departamento de Posgrado.
Ingenierías No.100, Col. Huapalcalco, C.P. 43629, Tulancingo, Hgo., México.
Likan_oi@hotmail.com, omar.aguilar@upt.edu.mx

^bUniversidad Nacional Autónoma de México, Facultad de Ingeniería, División de Ingeniería Eléctrica.
Av. Universidad 3000, Cd. Universitaria, Delegación Coyoacán, C.P. 04510 Ciudad de México, México.
rtapia@fi-b.unam.mx

^cUniversidad Autónoma Metropolitana, Unidad Azcapotzalco, Departamento de Energía.
Av. San Pablo No. 180, Col. Reynosa Tamaulipas, C.P. 02200, Delegación Azcapotzalco, Ciudad de México, México.
{fbeltran, ilg}@azc.uam.mx

2016 Published by *DIFU*_{100ci}@ <http://difu100cia.uaz.edu.mx>

Resumen

Se presenta el modelado matemático y un enfoque de control proporcional derivativo (PD) de un helicóptero de cuatro rotores, el cual se conoce en la literatura con el nombre de Quadrotor. Se describe también algunas estrategias de control y aplicaciones actuales del Quadrotor desarrolladas por varios grupos de investigación. El modelo matemático de este sistema mecatrónico se desarrolla mediante la formulación de Newton-Euler. Posteriormente se plantea un controlador PD para seguimiento de trayectorias de referencia. Se incluyen algunos resultados de simulación computacional que muestran el desempeño adecuado del esquema de control.

Palabras clave: Quadrotor, Helicóptero de cuatro rotores, Modelado matemático, Control PD.

1. Introducción

Un robot aéreo es un sistema capaz de sostener el vuelo sin control directo del humano y es capaz de ejecutar una tarea específica [1]. Los robots aéreos o vehículos aéreos no tripulados UAV (*Unmanned Aerial Vehicles*) se han vuelto rápidamente comunes y abarcan un enorme rango de tamaños y formas [2]. Encontramos a los de ala fija, los dirigibles y los vehículos aéreos provistos de rotores. Estos últimos tienen una amplia gama de configuraciones y su principal característica es que permiten el despegue y aterrizaje vertical. Aquí, se incluyen al helicóptero convencional, los de tipo coaxial y múltiples rotores, donde se identi-

can las configuraciones de cuatro, seis y ocho rotores.

Un Quadrotor es un sistema sub-actuado (*underactuated*), con seis grados de libertad y cuatro rotores independientes. Se requieren técnicas de control eficientes para la navegación y estabilización del Quadrotor. En la literatura existen diversas propuestas para llevar a cabo estas tareas. La elección de una técnica adecuada de control depende principalmente de la tarea que se le asigne al robot. Esta configuración no es nueva, su existencia data del año 1922. Un año previo, la *US Army Air Corps* encomendó al Dr. Bothezat el desarrollo de una máquina de vuelo vertical [3]. En la figura 1 se muestra el prototipo de Bothezat, el cual tenía un peso de 1700 kg, y su primer vuelo fue el 22 de Octubre de 1922.



Figura 1. El Quadrotor de Bothezat [3].

En años recientes el interés de los investigadores y especialistas de aeronáutica se ha incrementado considerablemente, lo que ha propiciado el desarrollo de nuevas configuraciones de drones, el cual es el término más común con el que quizás se hace referencia a los robots aéreos. El uso actual de nuevas tecnologías para el diseño, adquisición y procesamiento de señales, suministro de energía, manufactura, control, navegación, etc., ha beneficiado el desarrollo de los drones, con tamaños cada vez más compactos y capacidades de procesamiento elevadas, que van de la mano con las nuevas tendencias para su implementación. Debido a esto, el interés de los investigadores en las diversas áreas de desarrollo sigue en constante crecimiento, para su mejora, optimización y exploración de nuevas áreas de interés relacionadas al Quadrotor.

Las aplicaciones para los robots aéreos, incluido el Quadrotor, son variadas y en constante crecimiento, destacando las siguientes [1]:

- Sensado remoto.
- Atención en desastres.
- Vigilancia.
- Búsqueda y rescate.
- Transportación.
- Asistencia para descargas.
- Adquisición de imágenes.

Todas éstas lideradas por el desarrollo en el contexto militar. Un ejemplo de la asistencia para descargas se tiene cuando se requiere sofocar un incendio y el vehículo aéreo descarga sustancias químicas para contrarrestarlo, o en tareas de fumigación. En México,

el uso de UAV's de manera formal, nos direcciona a las paraestatales como PEMEX y CFE, que hacen uso de estos robots para el monitoreo de sus instalaciones y líneas de suministro en coordinación con elementos de la Secretaría de la Defensa Nacional. Una constante actualización y administración de recursos en estas actividades, demandan el desarrollo de nuevos prototipos con diversas características de operación, para solventar las necesidades particulares de cada aplicación.

En este artículo se describen algunos esquemas de control automático reportados en la literatura para un Quadrotor. Se presenta también el modelado matemático del dron a través de la formulación de Newton-Euler. Además, se describe un enfoque de control PD para tareas de seguimiento de trayectorias especificadas para este helicóptero de cuatro rotores. Se incluyen algunos resultados de simulación computacional para mostrar el desempeño adecuado del esquema de control para seguimiento de trayectorias de posición constantes.

2. Esquemas de Control Automático Aplicados a un Quadrotor

En el proceso de diseño de algún esquema de control de movimiento para un Quadrotor se han realizado algunas consideraciones fundamentales. La más importante es considerar a un robot como un cuerpo rígido. Se requieren de dos marcos de referencia para describir su comportamiento dinámico. Un marco solidario al móvil y otro inercial fijo a la superficie terrestre. Así, los siguientes puntos son comunmente tomados en cuenta en el diseño del control de movimiento para un Quadrotor:

- El centro de masa del robot es coincidente con el origen del marco de referencia del cuerpo.
- La estructura se supone rígida y simétrica.
- La aspas son rígidas.
- El empuje y arrastre son función del cuadrado de las velocidades angulares.

En la literatura es posible encontrar diversas propuestas basadas en controladores lineales y no lineales, todas ellas para lograr un alto nivel de desempeño [4]. Entre las principales técnicas de control resaltan controladores proporcional derivativo (**PD**), proporcional integral derivativo (**PID**), backstepping, H_∞ , estimadores de estados (filtros de Kalman), regulador lineal cuadrático (**LQR** por sus siglas en inglés), redes neuronales artificiales (**ANN** por sus siglas en inglés), algoritmos de control difuso, por mencionar algunos. El modelo no lineal es de gran utilidad, debido a que proporciona mayor

información del comportamiento físico del vehículo [5]. Sin embargo, el modelo linealizado se emplea ampliamente en la literatura, debido a que en general presenta buenos resultados de regulación alrededor de estados de operación de equilibrio deseados. Este modelo se utiliza con condiciones definidas de vuelo (hover), siendo estables y efectivos sólo en rangos de operación que corresponden a ángulos razonablemente pequeños. Dentro de las ventajas de estos diseños basados en controladores lineales se pueden mencionar: a) presentan un desempeño aceptable y b) tienen una estructura simple. Este último aspecto es el más relevante, considerando que el obtener un modelo dinámico exacto del sistema puede ser una tarea compleja [6]. El controlador **PID** funciona calculando los errores o diferencia entre la salida medida y el valor de referencia asignado, definiendo las entradas de control para ajustar la velocidad de los motores. Aunque este método de control ha sido propuesto inicialmente para sistemas lineales, también se ha utilizado para el control de sistemas no lineales [7]. En [6, 8, 9, 10, 11, 12] los estudios se enfocan en determinar un control óptimo a través de algoritmos de control convencionales PD y PID, cada trabajo tiene su propia metodología de sintonización. En [9] se presenta un algoritmo **PD** aumentado para mejorar la respuesta transitoria principalmente ante perturbaciones, la actitud del vehículo se analiza mediante el uso de cuaternios para evitar singularidades *Gimbal-Lock*.

En [8] se desarrolla un algoritmo **PID** y se hace referencia a un modelo simplificado obtenido por el método de *Euler-Lagrange*. En este estudio no se consideran los efectos giroscópicos obteniendo un modelo relativamente simple, lo cual permite contar con un primer acercamiento en el modelado, simulación e implementación de un robot de estas características. En [13] se considera una versión del modelo incluyendo efectos giroscópicos, realizando un análisis con controladores **PD**, backstepping y control inverso, el último tiene como objetivo el control del sistema a través de su dinámica inversa.

El algoritmo LQR es una forma de control óptimo lineal que tiene como objetivo reducir la magnitud de la entrada de control sin afectar el desempeño del algoritmo de control. En [6] se emplea este regulador para obtener los parámetros de ajuste que minimizan las alteraciones no deseadas (altitud) y al mismo tiempo limitar la energía de la acción de control usando un algoritmo que minimice una función de costo o índice de desempeño con factores de ponderación. En [14] se compara el desempeño del controlador **LQR** contra un controlador no lineal basado en el análisis de Lyapunov, el modelo se desarrolla de la formulación *Euler - La-*

grange. En [12] se encuentran algoritmos **PID** y **LQR**, primero se presentan las constantes del controlador PID derivadas de la teoría LQR, posteriormente se sintoniza el algoritmo **PID** con un procedimiento **ITAE** (*integral of time-weighted absolute error*) que permite minimizar el error.

De igual manera los controles convencionales se utilizan como complementos de controladores más complejos y robustos. En [15, 16] se fundamenta el diseño de controladores no lineales del tipo backstepping. En [17] se implementa un controlador backstepping especificando los parámetros empleados en la etapa de simulación, donde se proponen condiciones iniciales adversas para el vuelo estable del robot, los resultados son interesantes considerando que se tiene una buena respuesta ante perturbaciones considerables. En [18] el algoritmo de modos deslizantes se utiliza como observador de estado para estabilizar el sistema, adicionalmente se diseña un observador-estimador que pretende desprestigiar las perturbaciones inherentes del sistema con el menor número de sensores, esto se logra mediante una estrategia de linealización en la retroalimentación.

Es importante citar que en la literatura donde se desarrollan algoritmos complejos de control, su desempeño se compara bajo diversas circunstancias con controladores convencionales analizando su capacidad de controlabilidad. En [19, 20] se implementa una estrategia de modos deslizantes y posteriormente se discuten beneficios del controlador frente a las características de operación que entrega un controlador PID. En [21] se contempla el uso de un controlador por modos deslizantes, se proporcionan detalles de las simulaciones que validan su desempeño considerando algunos objetivos como la estimación de variables aerodinámicas y el seguimiento de trayectorias. Por su parte en [5, 7, 22, 23, 24] se optimizan controladores convencionales con algoritmos adaptativos. En [5] se presentan consideraciones del entrenamiento de la red neural, en línea y fuera de línea, definiendo dos capas (SHL por sus siglas en inglés) para estimar parámetros desconocidos en el modelo. En [7] un controlador neuro-PID realiza el cálculo de los coeficientes óptimos para un correcto desempeño, se enfatiza una correcta caracterización de los valores primarios por ejemplo la regla de aprendizaje, en este tipo de algoritmos se busca disminuir la demanda de operaciones en la etapa de procesamiento. En [12] se lleva a cabo un control adaptativo con una estrategia complementada utilizando modos deslizantes.

Las diferentes representaciones del modelo matemático para esta clase de robots convergen a un cierto nivel de complejidad, el cual depende de los factores que se reconocen pueden afectar de manera importante

el comportamiento dinámico del sistema en un escenario de operación específico. En [25] se hace un análisis de contribuciones previas [11, 17, 26] concluyendo su desarrollo con un mayor número de variables que describen la dinámica del sistema, además, se tiene una descripción de las consideraciones que se deben tomar en cuenta en la etapa de implementación del control en el sistema Quadrotor. Finalmente, se puede apreciar que el uso de estimadores y observadores de estado es una opción importante para este tipo de tareas de control. En [27] se tiene un diseño con filtros, en este caso un filtro de Kalman lineal, así como la versión no lineal del filtro de Kalman extendido (EKF por sus siglas en inglés), que se diseña con una estrategia LQ para lograr un mejor desempeño.

En el presente artículo se describe un enfoque de control PD para tareas de movimiento planificado para un Quadrotor. Las ecuaciones de movimiento usadas en la síntesis de este esquema de control se presentan en la siguiente sección.

3. Modelo Matemático de un Quadrotor

3.1. Movimiento de un Quadrotor

Para que el robot consiga moverse, se deben variar las velocidades angulares de cada actuador y debido a la anulación de momentos se produce el movimiento ya sea rotacional o traslacional. Cuando las velocidades angulares son iguales para los cuatro motores y la fuerza de empuje es mayor a la fuerza ejercida por la gravedad el robot consigue moverse hacia arriba (véase la figura 2a); se debe considerar que los motores 1 y 3 giren conforme las manecillas del reloj mientras que el par restante giren en sentido en contra de las manecillas del reloj. Se distinguen tres tipos de movimientos angulares para el Quadrotor, como se observa en la figura 2b: el movimiento de *roll* donde las velocidades angulares de los motores 1 y 3 se mantienen constantes, para el caso del motor 2 se debe aumentar y disminuir para el motor 4, este movimiento permite el desplazamiento en la dirección *y*.

El movimiento de *pitch* es similar al *roll*, la diferencia radica en las condiciones variables de operación que actúan en el otro par de motores, por consiguiente el movimiento se consigue en dirección *x*. El movimiento de *yaw* resulta de variar de manera uniforme la velocidad de alguno de los pares de actuadores para conseguir un movimiento conforme o en contra de las manecillas del reloj del cuerpo del robot. En la figura 3 se observan las direcciones *x*, *y* y *z* para los desplazamientos del Quadrotor.

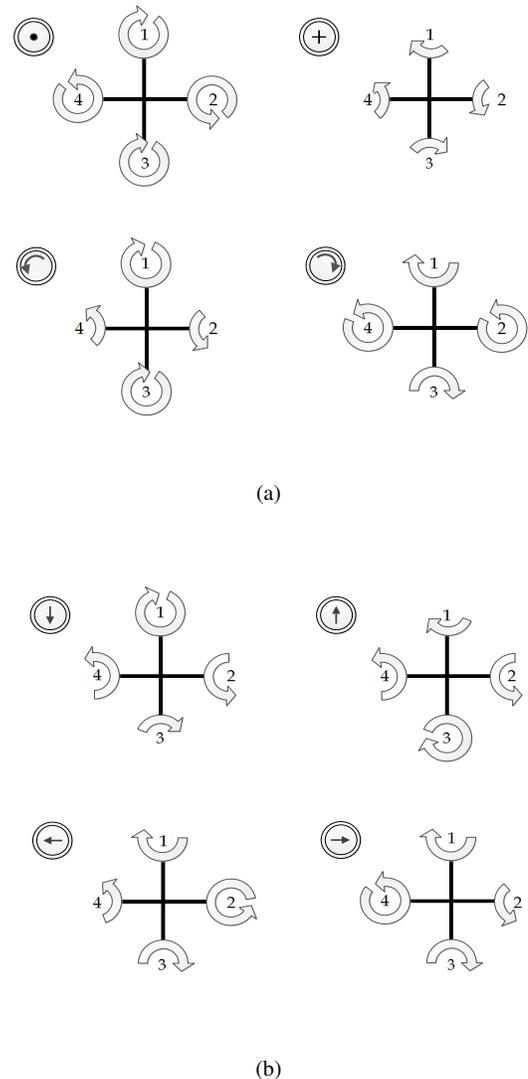


Figura 2. Movimiento del Quadrotor

Para representar el comportamiento dinámico del Quadrotor es necesario establecer coordenadas generalizadas que describan por completo la posición y actitud de un marco de referencia solidario al móvil O_{ABC} , respecto a un marco de referencia inercial fijo O_{NED} , como se muestra en la figura 3.

$$\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix}; \quad \eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}; \quad \mathbf{q} = \begin{bmatrix} \xi \\ \eta \end{bmatrix}$$

donde ξ y η son los vectores de posición lineal y angular respectivamente respecto del marco de referencia inercial fijo. Así, \mathbf{q} es el vector de coordenadas generalizadas que describe completamente la posición y orientación del cuerpo rígido mediante las posiciones

espaciales y los ángulos de *Euler*, y se observan en la figura 3. Se ha considerado la configuración donde se gira entorno a los ejes coincidentes al cuerpo un ángulo ϕ, θ y ψ para x, y y z respectivamente, refiérase a [14, 28].

3.2. Formulación Newton-Euler

En mecánica clásica, las ecuaciones de Newton - Euler describen la dinámica traslacional y rotacional de un cuerpo rígido. Estas leyes relacionan el movimiento del centro de gravedad de un cuerpo rígido con la suma de las fuerzas y torques (o momentos) actuantes **en el cuerpo** y se presentan como

$$m\dot{\mathbf{v}}_B + (\boldsymbol{\Omega}_B \times m\mathbf{v}_B) = \mathbf{F}_B \quad (1)$$

$$\mathbb{J}\dot{\boldsymbol{\Omega}}_B + (\boldsymbol{\Omega}_B \times \mathbb{J}\boldsymbol{\Omega}_B) = \boldsymbol{\tau}_B \quad (2)$$

Aquí, m es la masa del robot, \mathbf{v}_B es el vector de velocidad lineal del robot, $\boldsymbol{\Omega}_B$ es el vector de velocidades angulares (p, q, r) , \mathbb{J} es la matriz de inercias, \mathbf{F}_B y $\boldsymbol{\tau}_B$ son las entradas principales de fuerza u_1 , (que engloba las fuerzas F_1, F_2, F_3 y F_4 , figura 3) y par u_2, u_3 y u_4 , que corresponden con los movimientos de *roll*, *pitch* y *yaw* respectivamente. Así, $u_i, i = 1, 2, \dots, 4$ se utilizan como entradas de control del sistema, las cuales se relacionan con las velocidades angulares de cada rotor mediante la expresión (3). El subíndice B denota que las ecuaciones de movimiento se describen con respecto del marco de referencia móvil.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \quad (3)$$

Las constantes b y d son factores de empuje y arrastre y relacionan idealmente a las entradas principales de fuerza y par con el cuadrado de las velocidades angulares de cada motor $\Omega_1, \Omega_2, \Omega_3$ y Ω_4 , estos coeficientes pueden ser aproximados mediante la teoría del elemento pala.

Debido a que para un observador en el marco de referencia inercial espacial o fijo, la fuerza centrífuga es nula, se consideró expresar el movimiento traslacional en esta referencia y la dinámica angular se expresa respecto al marco solidario al móvil, expresado en (4) y (5). Considerando el vector de gravedad \mathbf{G} y los efectos giroscópicos $\boldsymbol{\Gamma}$ de manera particular, se tienen las siguientes expresiones

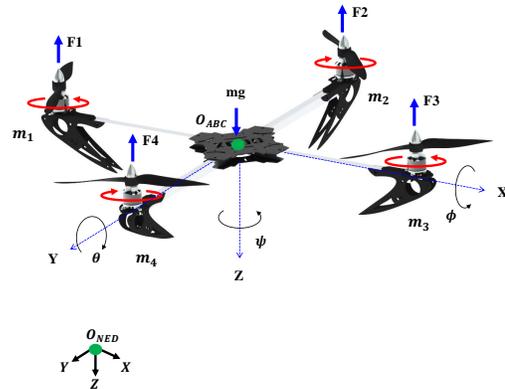


Figura 3. Diagrama esquemático de un helicóptero de cuatro rotores.

$$m\ddot{\boldsymbol{\xi}} = \mathbf{G} + \mathbf{R}\mathbf{F}_B \quad (4)$$

$$\mathbb{J}\dot{\boldsymbol{\Omega}}_B = \boldsymbol{\tau}_B - (\boldsymbol{\Omega}_B \times \mathbb{J}\boldsymbol{\Omega}_B) - \boldsymbol{\Gamma} \quad (5)$$

donde \mathbf{R} es la matriz de rotación que permite definir la orientación del helicóptero, donde s y c se usan para denotar a las funciones *sin* y *cos*, respectivamente.

$$\mathbf{R} = \begin{bmatrix} c\theta c\psi & -s\psi c\phi + c\psi s\phi s\theta & s\psi s\phi + c\psi s\theta c\phi \\ s\psi c\theta & s\psi s\phi s\theta + c\psi c\phi & -s\phi c\psi + s\psi s\theta c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (6)$$

Desarrollando la expresión (4) y considerando el amortiguamiento aerodinámico debido a la resistencia del aire: k_x, k_y y k_z , las ecuaciones que describen el movimiento de traslación se presentan como

$$\ddot{x} = \frac{u_1}{m} (\sin \theta \cos \phi \cos \psi + \sin \phi \sin \psi) - \frac{k_x}{m} \dot{x} \quad (7)$$

$$\ddot{y} = \frac{u_1}{m} (\sin \theta \sin \psi \cos \phi - \sin \phi \cos \psi) - \frac{k_y}{m} \dot{y} \quad (8)$$

$$\ddot{z} = \frac{u_1}{m} (u_1 \cos \phi \cos \theta) - g - \frac{k_z}{m} \dot{z} \quad (9)$$

Para el análisis de la dinámica angular se considera la ecuación (5), y debido a que las componentes de las velocidades angulares no son vistas de la misma manera según este ubicado el observador en alguno de los marcos de referencia ya definidos, se utilizan las relaciones cinemáticas (10), (11) y (12), véase [28].

$$\dot{\boldsymbol{\eta}} = \mathbf{w}_n^{-1} \boldsymbol{\Omega}_B \quad (10)$$

$$\boldsymbol{\Omega}_B = \mathbf{w}_n \dot{\boldsymbol{\eta}} \quad (11)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (12)$$

Si se requiere trabajar con un modelo no lineal simplificado se puede considerar como en [19, 20] que $p \approx \dot{\phi}$, $q \approx \dot{\theta}$ y $r \approx \dot{\psi}$, que se formaliza en las expresiones (13), (14) y (15), que son las utilizadas en este estudio para simulación

$$\ddot{\phi} = \frac{J_{yy} - J_{zz}}{J_{xx}} \dot{\theta} \dot{\psi} - \frac{J_r}{J_{xx}} \dot{\theta} \Omega_n + \frac{u_2}{J_{xx}} \quad (13)$$

$$\ddot{\theta} = \frac{J_{zz} - J_{xx}}{J_{yy}} \dot{\phi} \dot{\psi} - \frac{J_r}{J_{yy}} \dot{\phi} \Omega_n + \frac{u_3}{J_{yy}} \quad (14)$$

$$\ddot{\psi} = \frac{J_{xx} - J_{yy}}{J_{zz}} \dot{\phi} \dot{\theta} + \frac{u_4}{J_{zz}} \quad (15)$$

donde Ω_n es la suma algebraica de las velocidades angulares *ec.*(16), que contribuyen a los efectos giroscópicos provocados por la rotación de las aspas de los motores, su momento de inercia se expresa mediante J_r . Debido a que la estructura del robot se considera simétrica, solamente se consideran los elementos de la diagonal del tensor de inercias J_{xx} , J_{yy} y J_{zz} .

$$\Omega_n = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (16)$$

4. Un Esquema de Control PD

Debido a que la dinámica del sistema está descrita por ecuaciones diferenciales no lineales acopladas de segundo orden, un esquema de control PD puede ser usado para estabilizarlo alrededor de un estado de operación de equilibrio deseado [2], ciertamente con algunas imitaciones en su desempeño dinámico. En las ecuaciones (17)-(20) se presenta la estructura para las cuatro entradas de control principales del sistema, la compensación de la gravedad para u_1 y las componentes del tensor de inercias han sido consideradas.

$$u_1 = [\ddot{z}_d - k_{Dz}(\dot{z} - \dot{z}_d) - k_{Pz}(z - z_d) + g] \frac{m}{c\phi c\theta} \quad (17)$$

$$u_2 = [\ddot{\phi}_d - k_{D\phi}(\dot{\phi} - \dot{\phi}_d) - k_{P\phi}(\phi - \phi_d)] J_{xx} \quad (18)$$

$$u_3 = [\ddot{\theta}_d - k_{D\theta}(\dot{\theta} - \dot{\theta}_d) - k_{P\theta}(\theta - \theta_d)] J_{yy} \quad (19)$$

$$u_4 = [\ddot{\psi}_d - k_{D\psi}(\dot{\psi} - \dot{\psi}_d) - k_{P\psi}(\psi - \psi_d)] J_{zz} \quad (20)$$

donde el subíndice d hace referencia a la trayectoria deseada, las constantes proporcionales k_P y derivativas k_D son específicas para cada controlador. Para el seguimiento de trayectorias en las direcciones de los ejes x y y , es necesario establecer control sobre los ángulos θ y ϕ , respectivamente. Debido a esto se proponen dos

referencias internas simplificadas θ_d y ϕ_d , las cuales se deducen de las ecuaciones (7) y (8):

$$\theta_d = \sin^{-1} \left(\frac{m}{u_1} u_x - \frac{\sin \phi \sin \psi}{\cos \phi \cos \psi} \right) \quad (21)$$

$$\phi_d = \sin^{-1} (u_x \sin \psi - u_y \cos \psi) \quad (22)$$

Por lo tanto, para asegurar el seguimiento de trayectorias se emplean los pseudocontroladores (entradas de control auxiliar) u_x y u_y

$$u_x = \ddot{x}_d - k_{Dx}(\dot{x} - \dot{x}_d) - k_{Px}(x - x_d) \quad (23)$$

$$u_y = \ddot{y}_d - k_{Dy}(\dot{y} - \dot{y}_d) - k_{Py}(y - y_d) \quad (24)$$

Nótese que las ganancias de control k_{Pi} y k_{Di} , $i = z, \phi, \theta, \psi, x, y$, se deben seleccionar apropiadamente de tal manera que el conjunto de ecuaciones diferenciales ordinarias que conforman la dinámica del error en lazo cerrado sea asintóticamente estable.

5. Resultados en Simulación

El desempeño del esquema de control PD fue evaluado mediante simulación computacional usando la plataforma de *Simulink*. El conjunto de valores de los parámetros del Quadrotor se describen en la Tabla 1.

Tabla 1. Parámetros del sistema

Parámetro	Valor	Unidades
g	9.81	m/s^2
m	0.973	kg
l	0.225	m
J_r	0.0000286	Nms^2
J_{xx}	0.00866	Nms^2
J_{yy}	0.00866	Nms^2
J_{zz}	0.000000229	Nms^2

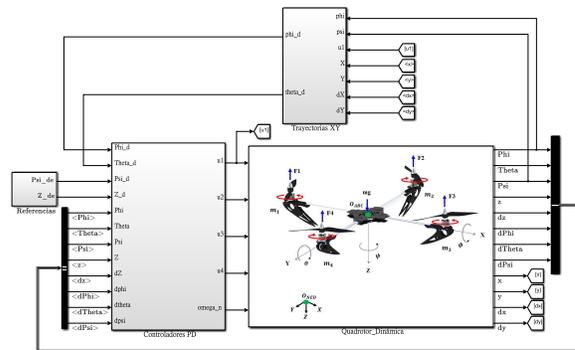


Figura 4. Diagrama de bloques del sistema en Simulink.

En las figuras 4 y 5 se presentan los bloques programados en *Simulink*, donde se observan las variables

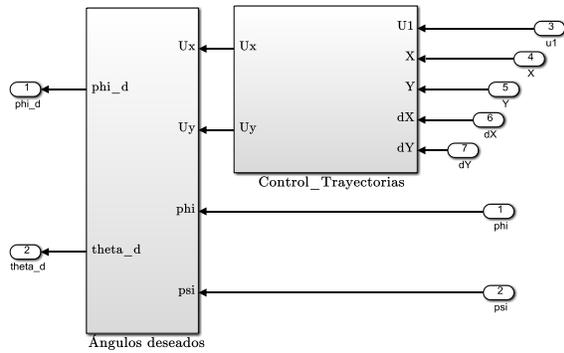


Figura 5. Bloque generador de trayectorias de movimiento en las direcciones x y y .

retroalimentadas para la implementación del esquema de control.

Para verificar el desempeño de los controladores de movimiento, se especificaron las características para cada trayectoria de referencia. Posterior a ello se presentan las respuestas del sistema en lazo cerrado. En éstas, las líneas continuas representan las señales reales y las líneas discontinuas las referencias de movimiento planificadas para el Quadrotor.

La trayectoria de referencia de posición para la dirección x esta determinada por la expresión (25). En la figura 6 se puede observar el seguimiento satisfactorio en lazo cerrado de la trayectoria establecida para el movimiento en el eje x .

$$x = \begin{cases} 0 \text{ m} & t < 6 \text{ s} \\ 4 \text{ m} & t \geq 6 \text{ s} \end{cases} \quad (25)$$

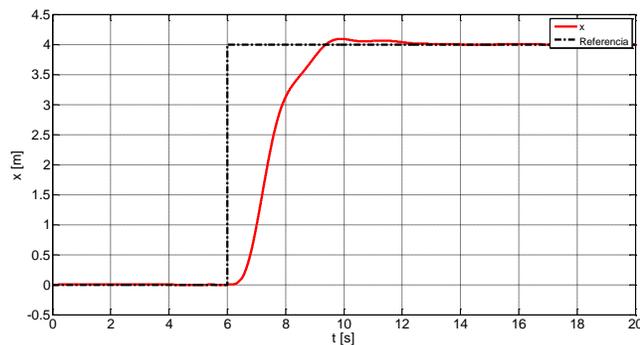


Figura 6. Seguimiento en lazo cerrado de la trayectoria de referencia en dirección x .

Para el movimiento en la dirección y se establecieron dos cambios de referencia descritos por (26). La figura 7 muestra el seguimiento adecuado en lazo cerrado para esta planificación deseada para la posición en la

dirección y .

$$y = \begin{cases} 0 \text{ m} & t < 0.2 \text{ s} \\ 1.5 \text{ m} & 0.2 \text{ s} \geq t < 15 \text{ s} \\ 1 \text{ m} & t \geq 15 \text{ s} \end{cases} \quad (26)$$

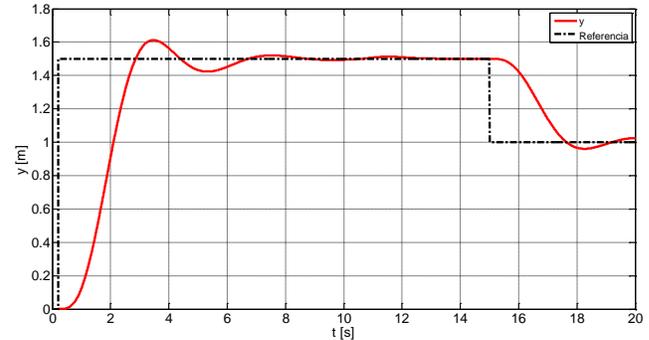


Figura 7. Seguimiento en lazo cerrado de la trayectoria de referencia en dirección y .

Con respecto al movimiento en la dirección z solo se presenta un cambio en la trayectoria de referencia como se describe por (27). El seguimiento aceptable de esta trayectoria se puede observar en la figura 8.

$$z = \begin{cases} 0 \text{ m} & t < 10 \text{ s} \\ 3 \text{ m} & t \geq 10 \text{ s} \end{cases} \quad (27)$$

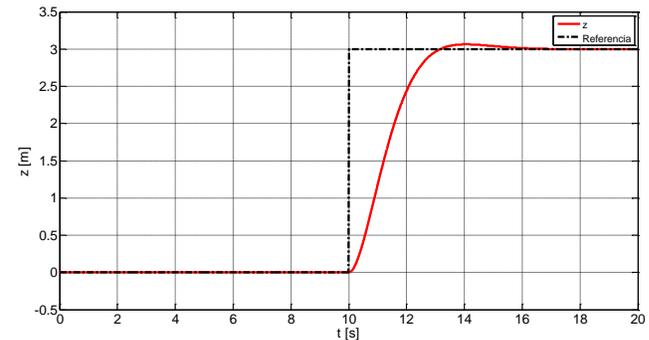


Figura 8. Seguimiento en lazo cerrado de la trayectoria de referencia en dirección z .

Las trayectorias para ϕ y θ están dadas por las expresiones (21) y (22), las cuales son generadas por el bloque correspondiente mostrado en la figura 5, por lo que están sujetas a condiciones operativas del seguimiento de trayectorias para x y y . El seguimiento razonable en lazo cerrado de estas trayectorias de referencia generadas en línea, de acuerdo a los movimientos reales en las direcciones x y y , se describe en las figuras 9 y 10.

Ahora, la trayectoria de referencia de posición angular para ψ se define de manera independiente a los demás

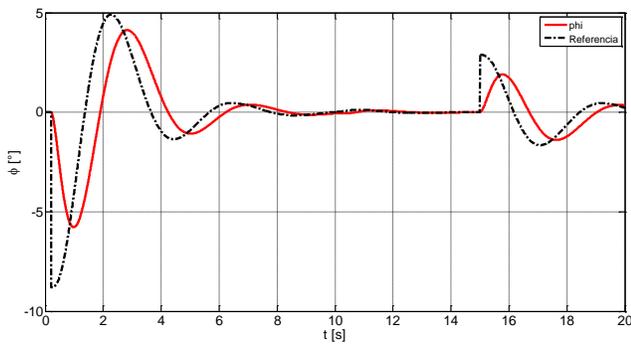


Figura 9. Seguimiento en lazo cerrado de la trayectoria de referencia ϕ .

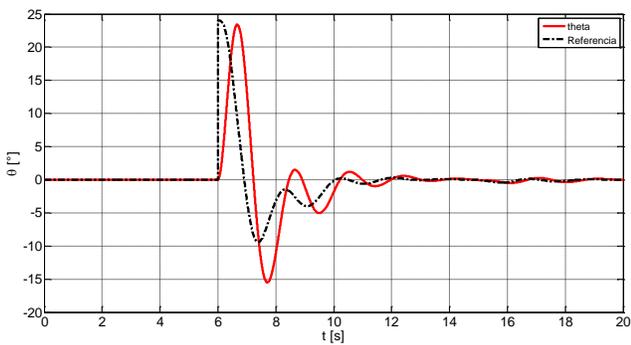


Figura 10. Seguimiento en lazo cerrado de la trayectoria de referencia θ .

movimientos por la expresión (28). En la figura 11 se verifica el seguimiento asintótico de esta trayectoria planificada.

$$\psi = \begin{cases} 0^\circ & t < 10 \text{ s} \\ 10^\circ & 10 \text{ s} \leq t < 16 \text{ s} \\ -5^\circ & t \geq 16 \text{ s} \end{cases} \quad (28)$$

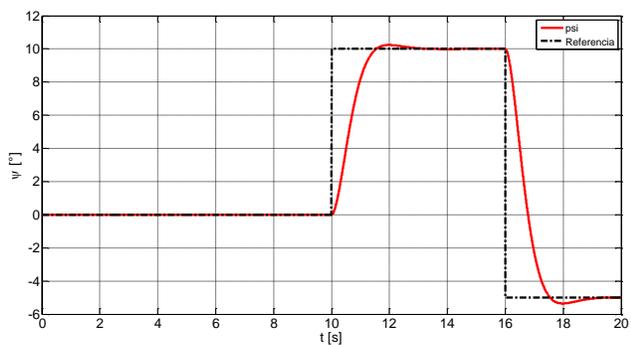


Figura 11. Seguimiento en lazo cerrado de la trayectoria de referencia ψ .

Finalmente, en las figuras 12 y 13 se presentan las

respuestas de los controladores del tipo PD correspondientes a las trayectorias de movimiento establecidas para el Quadrotor.

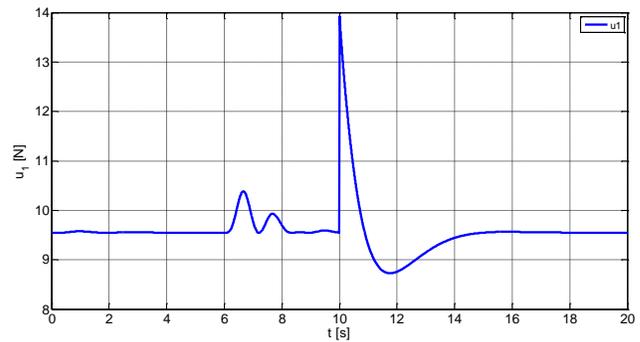


Figura 12. Señal de control de fuerza u_1 .

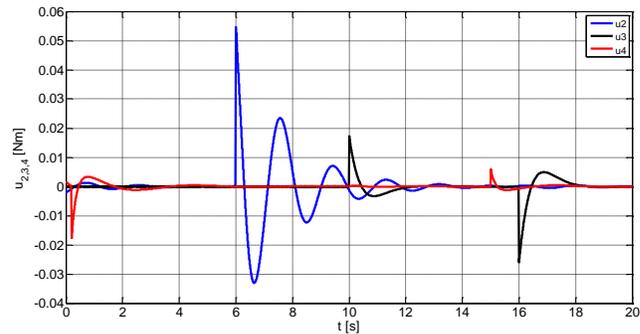


Figura 13. Señales de control de par u_2 , u_3 y u_4 .

6. Conclusiones

En este artículo se ha descrito el modelo matemático y un esquema de control PD para un Quadrotor. Algunos enfoques de control y aplicaciones de este vehículo aéreo fueron también presentados. El modelo matemático fue obtenido por el método de Newton-Euler y usado para la síntesis del esquema de control de seguimiento. El desempeño del control fue evaluado de manera satisfactoria mediante simulación computacional. Los resultados permiten observar que el desempeño del controlador PD presentado, cumple de manera aceptable con el seguimiento de las trayectorias planificadas para el Quadrotor. No obstante, se observan ligeros sobreimpulsos y retardos en las respuestas temporales del sistema. Futuros trabajos de investigación están orientados a la síntesis de esquemas de control robustos y eficientes para mejorar significativamente el seguimiento de movimientos planificados para un helicóptero de

cuatro rotores. Se tiene interés en considerar el problema de control robusto con respecto a perturbaciones endógenas y exógenas, debidas a dinámicas no modeladas e interacciones del Quadrotor con su entorno de operación. El problema del seguimiento de trayectorias más complejas para el movimiento de un Quadrotor será también abordado. Finalmente, se pretende proponer alternativas para atenuar el ruido que puede existir en las señales de medición y en las entradas de control, el cual podría deteriorar significativamente el desempeño de cualquier política de movimiento en lazo cerrado planificada para un Quadrotor.

Referencias

- [1] B. Siciliano and O. Khatib. *Handbook of Robotics*, 3rd Edition. Springer-Verlag Berlín Heidelberg, 2008, ISBN 978-3-540-23957-4.
- [2] P. Corke. *Robotics, Vision and Control*, 1st Edition. Springer-Verlag Berlín Heidelberg, 2011, ISBN 978-3-642-20143-1.
- [3] P. Castillo *et al*, *Modelling and Control of Mini-Flying Machines*. Springer-Verlag London, 2005, ISBN 1-85233-957-8.
- [4] K.Y. Chee and Z.W. Zhong, "Control, navigation and collision avoidance for an unmanned aerial vehicle". *Sensors and Actuators A: Physical* 190, pp. 66-76, 2013.
- [5] H. Boudjedir *et al*, "Adaptive neural network control based on neural observer for a Quadrotor unmanned aerial vehicle". *Advanced Robotics* 28(17), pp. 1151-1164, 2014.
- [6] B. Meng-Leong *et al*, "Low-Cost Microcontroller-based Hover Control Design of a Quadcopter". *Proc. of the International Symposium on Robotics and Intelligent Sensors (IRIS)* 41, pp. 458-464, 2012.
- [7] M.Fatan *et al*, "An Adaptive Neuro PID for Controlling the Altitude of Quadcopter Robot". *Proc. of the IEEE 18th International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 662-665, 2013.
- [8] A. L.Salih *et al*, "Flight PID controller design for a UAV Quadrotor". *Scientific Research and Essays* 5(23), pp. 3660-3667, 2010.
- [9] N. Johnson *et al*, "Enhanced proportional-derivative control of a micro Quadcopter". *Proc. of the ASME Dynamics Systems and Control Conference*, pp. 1-5, 2013.
- [10] B. Erginer and E. Altug, "Modeling and PD Control of a Quadrotor VTOL Vehicle". *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 894-899, 2007.
- [11] S. Bouabdallah *et al*, "PID vs LQ Control Techniques Applies to an Indoor micro Quadrotor". *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems* 3, pp. 2451-2456, 2004.
- [12] L. M.Argentim *et al*, "PID, LQR and LQR-PID on a Quadcopter Platform". *Proc. of the IEEE International Conference on Informatics, Electronics & Vision (ICIEV)*, pp. 1-6, 2013.
- [13] I. C. Dikmen *et al*, "Attitude Control of a Quadrotor". *Proc. of the 14th IEEE International Conference on Recent Advances in Space Technologies*, pp. 722-727, 2009.
- [14] P.Castillo *et al*, "Modelado y estabilización de un Helicóptero con cuatro rotores". *Revista Iberoamericana de Automática e Informática Industrial*, pp. 41-57, 2007.
- [15] A.A. Mian *et al*, "Backstepping based PID Control Strategy for an Underactuated Aerial Robot". *Proc. of the 17th IFAC World Congress* 17, pp. 15636-15641, 2008.
- [16] A.A. Mian and W. Daobo, "Modeling and Backstepping-based Nonlinear Control Strategy for a 6 DOF Quadrotor Helicopter". *Chinese Journal of Aeronautics* 21(3), pp. 261-268, 2008.
- [17] S. Bouabdallah and R. Siegwart, "Backstepping and Sliding-mode Techniques. Applied to an Indoor Micro Quadrotor". *Proc. of the IEEE International Conference on Robotics & Automation*, pp. 2247-2252, 2005.
- [18] A. Benallegue *et al*, "High-order sliding-mode observer for a Quadrotor UAV". *International Journal of Robust and Nonlinear Control* 18(4-5), pp. 427-440, 2008.
- [19] Z. En-Hui *et al*, "Second order sliding mode control for a Quadrotor UAV". *ISA Transactions* 53(4), pp. 1350-1356, 2014.
- [20] Jing-Jing X.and En-Hui Z. "Position and attitude tracking control for a Quadrotor UAV". *ISA Transactions* 53(3), pp. 725-731, 2014.
- [21] L. Luque *et al*, "Robust block second order sliding mode control for a Quadrotor". *Journal of the Franklin Institute* 349(2), pp. 719-739, 2012.
- [22] T. Dierks and S. Jagannathan, "Output Feedback Control of a Quadrotor UAV Using Neural Networks". *IEEE Transactions on Neural Networks* 21(1), pp. 50-66, 2010.
- [23] C. Nicol *et al*, "Robust Neural Network Control of a Quadrotor helicopter". *Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1233-1238, 2008.
- [24] B.J. Emran and A. Yesildirek, "Robust Nonlinear Composite Adaptative Control of Quadrotor" *International Journal of Digital Information and Wireless Communications (IJDWC)*, pp. 45-57, 2014.
- [25] S.Bouabdallah and R.Siegwart, "Full Control of a Quadrotor". *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 153-158, 2007.
- [26] S.Bouabdallah, "Design and Control of an Indoor Micro Quadrotor". *IEEE International Conference on Robotics & Automation*, 2004.
- [27] P. Bauer *et al*, "LQ Servo control design with Kalman filter for a Quadrotor UAV". *Periodica Polytechnic Transportation Engineering* 36(1-2), pp. 9-14, 2008.
- [28] H. Goldstain *et al*, *Classical Mehanics*, 3rd Edition Addison Wesley, 2000.

Software Educativo Didáctico para la Oralización de Niños Hipoacúsicos

Miguel Angel Navarrete Sánchez^a, Osbaldo Vite Chávez^a, Roberto Olivera Reyna^a, Reynel Olivera Reyna^a, Eduardo García Sánchez^a, Claudia A. Rivera Romero^b, Elvia Palacios Hernández^b, Antonio Marín Hernández^c

^aUniversidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica Campus Jalpa
Libramiento Jalpa Km. 156+380, Fraccionamiento Solidaridad, Carretera Guadalajara-Salttillo, Jalpa Zacatecas México, CP 99601.
<http://campusjalpa.uaz.edu.mx/>

^bUniversidad Autónoma de San Luis Potosí. Facultad de Ciencias. Departamento de Electrónica.
Av. Salvador Nava Mtz. s/n Zona Universitaria, San Luis Potosí, Mexico, C. P. 78290.
galia.fc.uaslp.mx/~epalacios

^cUniversidad Veracruzana. Departamento de Inteligencia Artificial.
Sabastián Camacho No. 5, Zona Centro, Xalapa, Veracruz, México. C.P. 91090
<https://www.uv.mx/dia/>

2016 Published by DIFU_{100ci}@ <http://difu100cia.uaz.edu.mx>

Resumen

Actualmente, en la intervención educativa, se clasifican a los niños con dificultades auditivas, en sordos profundos y en hipoacúsicos, los cuales presentan una audición limitada. A pesar de ello, logran desarrollar un nivel de lenguaje y comunicación propio, por debajo de los niños sin estas dificultades. Con la ayuda de diferentes dispositivos y herramientas computacionales, se les puede apoyar para que se integren a la sociedad con mayor facilidad en todos los ámbitos al desarrollar estas habilidades de lenguaje y comunicación. Por lo anterior, en este trabajo, se presenta una herramienta de software educativo, que tiene como objetivo fortalecer el proceso de la enseñanza-aprendizaje del lenguaje y de la oralización para niños hipoacúsicos que cursan el nivel básico. El software fue generado utilizando el modelado de proceso para desarrollar entornos didácticos interactivos computacionales (EDIC). En el presente trabajo se puede apreciar la culminación del desarrollo de software como herramienta para su instalación en instituciones educativas de nivel básico y en los hogares.

Palabras clave: Enseñanza-Aprendizaje, Hipoacúsico, Oralización, Software Educativo.

1. Introducción

Una persona con capacidades diferentes se puede definir como “toda persona que por razón congénita o adquirida presenta una o más deficiencias de carácter físico, mental, intelectual o sensorial permanente o temporal” [1], y que al interactuar con el

entorno social, son restringidos en igualdad de condiciones con respecto a las demás personas. La mayor parte de ellos se comunican con pensamientos, necesidades, sentimientos e ideas a través del habla y estas son percibidas a través de la audición y visión; actualmente en la intervención educativa, se clasifica a los niños con dificultades auditivas en: hipoacúsicos, que

presentan una audición limitada, pero que, con prótesis o sin ella, tienen una audición bastante funcional para la vida diaria permitiendo la adquisición del lenguaje oral por vía auditiva, ver Fig. 1 y sordos profundos, donde su audición no funciona [2].

La Dirección General de Planeación y Estadística Educativa (DGPyEE) de la Secretaría de Educación Pública (SEP) presenta las principales cifras del Sistema Educativo Nacional 2014-2015, sobre la población total atendida en educación especial, que es de 563 mil 976 mexicanos, esto, incluyendo a los alumnos que presentan necesidades educativas especiales temporales o permanentes, que pueden o no estar asociadas a una discapacidad o aptitudes sobresalientes. Para el caso particular de la hipoacusia, se atendió a 8 mil 383 alumnos, teniendo un incremento a comparación del periodo 2013-2014, que fue de 7 mil 964 [3].

Hoy en día, el software educativo se visualiza, como uno de los pilares del sistema educativo a distancia, y se perfila como la herramienta principal, utilizarán las próximas generaciones de educandos [4], es por ende, que en este trabajo, se desarrolló un software como complemento didáctico para el proceso de la enseñanza-aprendizaje del lenguaje y oralización de niños Hipoacúsicos, que permita al usuario con esta discapacidad, disponer de su propio diccionario multimedia, esto, con elementos multimedia, como: imágenes, videos, texto y audio, apoyados de sus maestros en la escuela, de maestros de lenguaje y comunicación que los apoyan de las Unidades de Servicio de Apoyo a la Educación Regular (USAER) y de los Centros de Atención Múltiple (CAM), y de sus padres de familia en el hogar, adaptándose de forma pertinente el material, a las necesidades de aprendizaje de contenidos de cada alumno. En la actualidad, existen una gran cantidad de herramientas multimedia que tienen como objeto apoyar el proceso enseñanza-aprendizaje, las cuales, se consiguen a un precio no tan accesible en algunos casos, aunque también pueden ser gratuitas o de libre distribución, pero sin la certeza de que cumplan con las necesidades de los usuarios. En el desarrollo de software educativo, es conveniente la participación de un equipo de expertos multidisciplinarios en diferentes áreas del conocimiento para cumplir con los requerimientos y necesidades que marcan los usuarios [5, 6]. El desarrollo y generación de software multimedia educativo ha avanzado gracias al uso de los lenguajes de programación, lenguajes de autor y sistemas de autor [7], que permiten generar aplicaciones más amigables y funcionales, y que en base a los requerimientos, necesidades y características planteadas, facilitan su desarrollo y generación. Sin embargo, en la mayoría de los casos, el diseñador no



Figura 1. Niño hipoacúsico con aparato, tomada de [8].

cuenta con amplios conocimientos sobre programación y el uso de los lenguajes de programación y de autor, no sabe utilizarlos. Sin embargo, esto no se presenta cuando el software es diseñado por un grupo multidisciplinario, en el que participa un experto de programación. Aunque es importante mencionar, que existen los sistemas de autor, que no utilizan comandos y sentencias propias de los lenguajes de programación y de autor, sino que trabajan a través de una interfaz más amigable, utilizando menús, iconos, plantillas, etc., que permiten generar y desarrollar el software educativo, de una manera más sencilla y práctica, y que no se requiere ser un experto en el área de computación.

2. Método

Para poder desarrollar software multimedia de calidad, es necesario, como ya se había mencionado anteriormente, asegurarnos que reúna los requisitos y necesidades del cliente y/o usuarios al que va dirigido, para lo cual, se debe seguir un proceso para su generación, denominado metodología, la cual, puede ser constituida como una combinación de los métodos existentes [9] en ingeniería de software (método de evolutivo, desarrollo en espiral, proceso incremental, ciclo de vida, en cascada, en reutilización, desarrollo concurrente, basada en componentes, proceso unificado, etc.) que permiten

describir las etapas o fases de producción [5, 14].

2.1. Metodología del desarrollo

En la literatura existe una gran cantidad de metodologías propuestas para la generación de software multimedia [4, 10, 11, 12, 13, 14]. Sin embargo, en este trabajo, se utilizó la metodología denominada El modelado del proceso para desarrollar Entornos Didácticos Interactivos Computacionales (EDIC) [15] que especifica claramente las actividades a desarrollar en cada una de sus fases o etapas, permitiendo que los sub-productos generados en cada una, por medio de un seguimiento lógico, formen un software educativo de calidad. Esto por la creciente necesidad de elaborar más y mejores recursos digitales que apoyen al proceso de enseñanza-aprendizaje. La metodología EDIC está dividida en seis etapas ver Fig. 2, las cuales se describen más adelante, y permiten enfocarse en generar una descripción detallada de la información que se debe ir generando en cada una de ellas, esto, con el fin de facilitar el proceso de implementación de la metodología. Cabe señalar que todo el proceso es iterativo, por lo que una vez que se concluyen todas las etapas, se puede regresar a la etapa que se considere requiere mayor detalle o presente errores. Así mismo, al final del proceso, se realiza un análisis de mejora continua, que permite detectar áreas de oportunidades y mejoras continuas en el proceso, esto, para refinar las prácticas aprendidas y adoptadas durante el proceso.

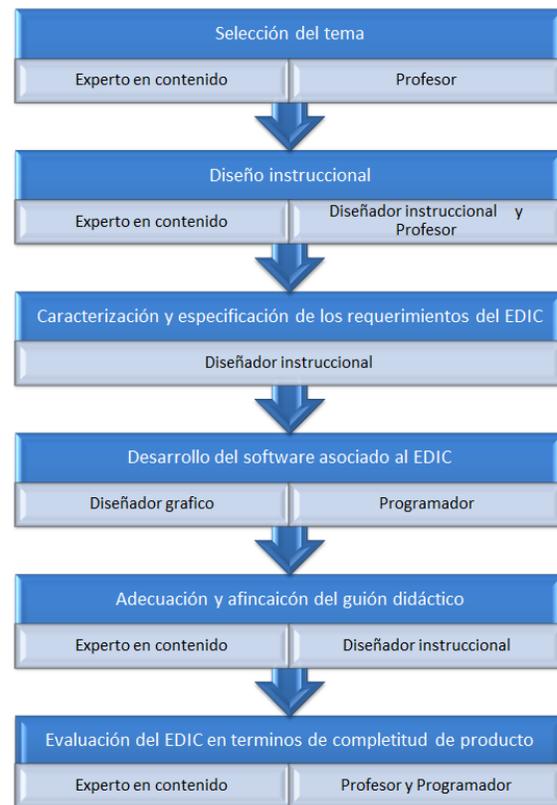


Figura 2. Etapas del modelo de procesos.

- A Etapa 1 Selección del tema. Se basa en el concepto que un profesor prioriza como fundamental, y se apoya en los expertos en contenido para determinar el nivel de profundidad del mismo; esto, para fomentar la comprensión por los estudiantes. La información se contempla por:
 - 1) Instrumento de especificaciones del tema.
 - 2) Evaluación diagnóstica realizada al inicio del curso.
 - 3) Ficha descriptiva del tema a cubrir.
- B Etapa 2 Diseño instruccional. Es el proceso a seguir para obtener el guion didáctico preliminar, participan en dicho proceso de análisis y desarrollo, el experto en contenido, el profesor y el diseñador instruccional.
 - 1) Instrumento del guion didáctico.
 - 2) Ficha descriptiva del tema a cubrir.
 - 3) Instrumento preliminar del guion didáctico.
- C Etapa 3 Caracterización y especificación de los requerimientos del EDIC. Se forman los requerimientos de la actividad académica, y se esquematizan

los diversos escenarios funcionales y adaptables en el software de apoyo por medio del diseñador instruccional.

- 1) Instrumento preliminar del guion didáctico.
 - 2) Documentos de especificación de requerimientos funcionales del software de apoyo.
- D Etapa 4 Desarrollo del software asociado al EDIC. Es considerada la metodología de diseño, implantación e implementación, en la que participan el programador y diseñador gráfico, sus actividades son:
 - 1) Documento de especificación de requerimientos del software de apoyo.
 - 2) Prototipo funcional del software para EDIC.
 - E Etapa 5 Adecuación y afinación del guion didáctico. Se realiza la revisión y prueba piloto, donde se identifican los elementos que requieran adecuación, participan los expertos en contenido y el diseñador instruccional. La información es contemplada por:
 - 1) Prototipo funcional del software para EDIC.
 - 2) Guion didáctico terminado.
 - 3) Software funcional.
 - F Etapa 6 Evaluación del EDIC en términos de completitud de producto. Una vez que se ha probado el



Figura 3. Caso de uso de SORHI.

EDIC en el aula, se realiza una evaluación donde participan los expertos en contenido, el profesor y el programador; se utilizan instrumentos de medición que cuantifican elementos tecnológicos, pedagógicos, de contenido y estéticos. Las actividades de la etapa son:

- 1) Guion didáctico terminado.
- 2) Software funcional.
- 3) Documento para evaluación de completitud.
- 4) Guiones didácticos.
- 5) Software para EDIC.

3. Resultados

La herramienta didáctica generada, denominada SORHI (Software como complemento didáctico para la oralización de niños hipoacúsicos) conjuga el código fuente con el diseño de interfaces que cumplen con los requerimientos que marcan las etapas del EDIC, obteniendo como resultado, la primera parte del desarrollo del software educativo. Su funcionalidad se visualiza en la Fig. 3, y que se describe de la siguiente manera: Primeramente, al iniciar el software aparece la interfaz de inicio de sesión, solicitando el nombre de usuario y la clave de acceso; se verifica su autenticidad y después muestra la pantalla principal; a continuación se visualiza la información global de SORHI, como se puede



Figura 4. Interfaz principal modo administrador.

observan en la Fig. 4. En este punto existen dos vistas, el modo administrador (profesor) y por default el de usuario (alumno):

1. Modo Administrador (profesor). Al desplegar la pantalla principal, aparecen inhabilitados algunas opciones de botón en el menú, en la parte superior, con el objetivo de que la interfaz inicie en modo alumno, al pulsar sobre el botón administrador, la aplicación solicita la contraseña, para activar el modo administrador, con el cual, se tiene acceso a las opciones; agregar (*cargar un archivo de imagen, audio y escribir un texto*), modificar y eliminar término. Además de todas las funciones con las que cuenta la interfaz. Ésta distinción se efectúa para evitar que los alumnos obtengan acceso a las funciones en las que realizan acciones no deseadas o, que modifiquen la información.
2. Modo Usuario (Alumno). Al desplegar la pantalla principal, como se mencionó anteriormente, se accede al modo Usuario. La funcionalidad de la ventana del menú principal, se lleva a cabo, al dar un clic a cualquiera de los botones, exceptuando aquellos destinados para el administrador. Las opciones de botones activados para el modo usuario son: buscar término, ayuda, flechas de navegación, escuchar audio, visualización de imagen y su texto descriptivo, además, una sección de pequeños botones con texto, que comprenden las letras del abecedario, los cuales permiten acceder a una lista de términos relacionados a la letra correspondiente, y finalmente la opción salir de SORHI, como se observa en la Fig. 5.

El software educativo, tiene características particulares en cuanto a la comunicación con el usuario, las cuales no se logran cuantificar mediante métricas, porque están relacionadas con conductas de aprendizajes [4]. Pero, las reglas en la construcción de un programa son las mismas, ya sea educativo, comercial, de



Figura 5. Interfaz principal modo usuario.

investigación, etc. Una vez generado el código fuente, el software debe probarse para descubrir tantos errores como sea posible antes de ser un producto final. Las pruebas se desarrollaron desde dos perspectivas diferentes:

1. Caja blanca. La lógica de programa interno se verificó con casos de prueba que:
 - Garantizaron que todas las rutas independientes dentro de un módulo se revisarán al menos una vez.
 - Se revisaron todas las decisiones lógicas en sus lados verdadero y falso.
 - Se ejecutaron todos los bucles en sus fronteras y dentro de sus fronteras operativas.
 - Se revisaron estructuras de datos internas para garantizar su validez.
2. Caja negra. Los requerimientos de software permiten derivar un conjunto de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Estas pruebas permitieron encontrar errores en las categorías siguientes:
 - Funciones incorrectas o faltantes.
 - Errores de interfaz.
 - Errores en las estructuras de datos.
 - Errores de comportamiento o rendimiento.
 - Errores de inicialización y terminación [5].

4. Conclusiones

Debido al creciente desarrollo del software educativo durante los últimos años, gran parte del mismo, ha sido realizado en forma desorganizada y poco documentada, y considerando el aumento exponencial que sufrirá en los próximos años, surge la necesidad de lograr metodologías multidisciplinarias para su desarrollo, por esta

razón, en este trabajo, se utilizó la metodología EDIC, siendo cuidadosos en seguir las etapas que comprenden dicha metodología, permitiéndonos llevar un buen diseño estructural e instruccional del contenido de la herramienta incluido dentro del sistema, y de esta manera, tener las bases necesarias para desarrollar un sistema que cumpla con los requerimientos y necesidades del usuario al que va dirigido el software. Esto permite construir programas educativos de calidad, siguiendo las pautas de las teorías educativas y de la comunicación, ya que el software es un elemento clave en la evolución de sistemas y productos basados en computadoras, y una de las tecnologías más importantes en todo el mundo. Esto nos permite concluir que se generó el software de Calidad denominado SORHI, el cual puede ser adoptado para ser utilizado en instituciones USAER y CAM que puedan hacer uso de él. como apoyo didáctico para los niños Hipoacúsicos a los que atienden.

4.1. Perspectivas

Al implementar el software SORHI, se podrían efectuar casos de estudio con niños hipoacúsicos, apoyados por sus maestros del nivel básico, personal de las USAER y CAM que atienden a estos niños y sus respectivos familiares, que permitan validar el software con respecto a los objetivos planteados en el Diseño Instruccional, al efectuar una evaluación de conocimientos y aprendizajes esperados, con respecto a los aprendidos.

De acuerdo con las recomendaciones del estudio principal de la problemática y necesidad, los resultados como aplicación en las pruebas de caja blanca y negra fueron viables, no obstante aún existen necesidades que complementan en gran medida el desarrollo del software y se han considerado como perspectivas:

- Agregar módulos que permitan la visualización de animaciones y elementos multimedia.
- Insertar un módulo que evalúe el aprendizaje y aprovechamiento de los alumnos.

Referencias

- [1] SEP., *Centro Digital de Recursos de Educación Especial.*, Disponible: <http://educacionespecial.sepdf.gob.mx/institucional-que-es-educacion-especial.php>, mayodel2012.
- [2] José del Río, María., *Lenguaje y comunicación en personas con necesidades especiales.*, Barcelona: Martínez Roca, 1997.
- [3] DGPYEE., *Principales Cifras del Sistema Educativo Nacional 2014 - 2015.* Disponible: http://planeacion.sep.gob.mx/assets/images/principales_cifras/2014_2015_PRINCIPALES_CIFRAS_BOLSILLO.pdf.

- [4] Galvis, A. (2000)., *Ingeniería de software educativo.*, Colombia: Universidad de los Andes.
- [5] Presmman, R. S. , *Ingeniería del Software. Un enfoque práctico.*, USA: Mc Graw Hill.
- [6] Marcano, I. & Benigni, G., *Análisis de alternativas metodológicas para el desarrollo de software educativo.*, Saber, 26(3): 297-304.
- [7] Razquin, P. , *Los sistemas de autor multimedia.*, Revista General de Información y Documentación, 8 (2), 127-139.
- [8] Infogen, *Defectos al nacimiento.* Disponible: <http://infogen.org.mx/sordera-congenita/>.
- [9] Braude, E., *Ingeniería de Software: Una perspectiva orientada a objetos.*, México, D.F.: Alfaomega.
- [10] Benigni, G. (2004)., *Una metodología orientada a objetos para la producción de software multimedia.*, Saber, 16(1): 26-32.
- [11] García Á. J. L. (2011)., *Análisis de los aspectos y factores que se consideran en el diseño de software educativo de calidad.*, Journal La Sociología en sus escenarios, (24).
- [12] Marqués, P. (2005)., *Modelos de Desarrollo de MDCS. La metodología de Pere Marqués.*, Recuperado de <http://www.peremarques.net/disdesa.htm>.
- [13] Polo, M. (2003)., *Aproximación a un modelo de Diseño: ADITE.*, Docencia Universitaria, 1(4): 67-83. Recuperado de http://www.ucv.ve/fileadmin/user_upload/sadpro/Documentos/docencia_vol4_nl_2003/7_art._4Marina_Polo.pdf.
- [14] Sommerville, I., *Ingeniería de Software.*, España: Pearson y Addison Wesley. 2007.
- [15] Cruz Flores, René., *Modelado del proceso para desarrollar entornos didácticos interactivos.*, Guadalajara Jalisco: s.n., 2012. Congreso Internacional de Investigación e Innovación en Ingeniería de Software 2012.

