 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-27

IDENTIFICACIÓN DEL SISTEMA DINÁMICO DE UN CUADRACOPTERO Y DESARROLLO DE UN MODELO DE SIMULACIÓN EN MATLAB

Duvier Alonso Sepúlveda Naranjo

Erica Johana Figueroa Durango

David Andrés Morales Olarte

Tecnología en Electrónica

Ricardo Andrés Velásquez Vélez

German David Goetz Sánchez

INSTITUTO TECNOLÓGICO METROPOLITANO

Octubre 14 del 2016

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RESUMEN

Los cuadracopteros son sistemas con modelos dinámicos no-lineales complejos con muchos parámetros que por lo general no son obvios y deben obtenerse experimentalmente. En este trabajo partimos de un modelo dinámico de vuelo completamente analítico del cuadracoptero, el cual es complementado con un modelo experimental obtenido de la caracterización de motores sin escobillas. Durante este trabajo se utilizaron varias técnicas de identificación y se encontró que el tipo de modelo más adecuado para los motores sin escobillas es el modelo no lineal Hammerstein-Wiener. El modelo completo del cuadracoptero se comparó contra vuelos reales de un cuadracoptero obteniéndose comportamientos muy similares.

Palabras clave: Identificación de sistemas, modelo dinámico de vuelo, cuadracopteros, modelos no lineales.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

RECONOCIMIENTOS

Agradecemos a Dios por hacer posible este proyecto. Al profesor Ricardo Andrés Velásquez Vélez por su constante ayuda, por el tiempo dedicado y los conocimientos que nos compartió durante el desarrollo de esta tesis. También reconocemos al profesor German David Goez Sánchez por su apoyo y acompañamiento. Además, agradecemos a nuestras familias por la motivación y confianza que siempre nos tuvieron.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACRÓNIMOS

AEyCC	Automática Electrónica y Ciencias Computacionales
ESC	<i>Electronic Speed Controller</i>
GPS	<i>Global Positioning System</i>
I2C	<i>Inter-Integrated Circuit</i>
IMU	<i>Inertial Measurement Unit</i>
OSD	<i>On-Screen Display</i>
PWM	<i>Pulse Width Modulation</i>
RPM	Revoluciones por Minuto
UAV	<i>Unmanned Aerial Vehicle</i>
USB	<i>Universal Serial Bus</i>

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

TABLA DE CONTENIDO

Justificación	7
Objetivo General.....	8
Objetivos Específicos	8
1.1 Modelamiento e Identificación de Sistemas	9
1.1.1 Modelos basados en la estimación de parámetros de modelos analíticos	9
1.1.2 Identificación de modelos lineales locales	10
1.1.3 Modelos RBF-ARX	11
1.1.4 Modelos de caja negra.....	11
1.1.4.1 Modelos de redes neuronales	11
1.1.4.2 Modelos de base de datos.....	11
1.2 Modelo dinámico de vuelo para un cuadricoptero en configuración X.....	12
1.2.1 Momento de Inercia de masa.....	12
1.2.1.1 Momento de inercial de los motores: cilindros solidos ($J_{x,M}$, $J_{y,M}$, y $J_{z,M}$)	13
1.2.1.2 Momento de inercia del HUB Central: cilindro solido ($J_{x,H}$, $J_{y,H}$, $J_{z,H}$)	14
1.2.1.3 Momento de inercia de los brazos o booms: cilindros largos ($J_{x,A}$, $J_{y,A}$, $J_{z,A}$)	16
1.2.2 Modelo matemático	17
1.2.2.1 Notación.....	17
1.2.2.2 Coeficiente de Empuje.....	19
1.2.2.3 Coeficiente de Torque	19
1.2.2.4 Planteamiento inicial del sistema matricial.....	20
1.2.2.5 Fuerzas Giroscópicas (Precesión)	20
1.2.2.6 Construcción matricial completa	20
1.2.2.7 Ecuaciones de estado	21
2.1 Metodología de Identificación	24
2.2 Materiales y Equipos	25
2.2.1 Componentes de un Cuadricoptero	25
2.2.1.1 Motores DC sin escobillas	25
2.2.1.2 Controlador de Velocidad Electrónico (ESC).....	26

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.1.1.3 Hélices.....	26
2.1.1.4 Batería.....	27
2.1.1.5 Controlador de vuelo	27
2.1.1.6 Frame	28
222.1.1.8 GPS.....	29
2.1.1.9 Radio Control	30
2.1.1.10 Radio Receptor	30
2.1.1.11 Radio telemetría	32
2.1.2 Sistema electrónico de un cuadrapterero.....	33
2.2 Caracterización del subsistema motor-hélice-ESC	34
2.3 Sistema de Visualización en Matlab	47
2.2.1 Orientación y matrices de rotación	47
3. RESULTADOS Y DISCUSIÓN.....	52
CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO.....	68
REFERENCIAS	69
APÉNDICE.....	70

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

INTRODUCCIÓN

Justificación

El desarrollo y popularización en la tecnología de vehículos aéreos no tripulados (UAV por sus siglas en inglés) ha sido bastante notable en los últimos años. Los UAVs son usados no solamente en aplicaciones militares sino también aplicaciones civiles y comerciales. En el campo civil los UAVs son la respuesta a muchas necesidades en áreas como la fotografía y exploración aérea, agricultura, seguridad, entre otras. Entre los diferentes tipos de UAVs, el cuadracoptero se destaca por su gran flexibilidad, gracias a la capacidad de realizar vuelo estacionario, lo que reduce el esfuerzo para llevar a cabo un aterrizaje con respecto a otros UAVs. Debido a esto los cuadracopteros son vistos como la plataforma UAV más prometedora para el desarrollo e implementación de aplicaciones civiles.

Los cuadracopteros son sistemas dinámicamente complejos en los que intervienen fuerzas externas como la gravedad, la velocidad del viento y diferentes perturbaciones que se puedan presentar en el transcurso de un vuelo. Durante la puesta a punto de aplicaciones con cuadracopteros es frecuente la necesidad de sintonizar el controlador de vuelo y estabilización. No obstante, la sintonización del controlador directamente en el cuadracopteros puede representar un peligro, no solo por la posibilidad de destrucción del vehículo, sino también por las posibles lesiones que pueda causar a las personas a su alrededor. En este contexto, este proyecto de grado aborda el problema de la obtención de un modelo de simulación para un cuadracoptero utilizando técnicas de modelamiento conocidas. El modelo debe ser lo suficientemente preciso como para permitir la sintonización segura de controladores para el cuadracoptero.

Este trabajo de grado está enmarcado en el proyecto P14103 “Plataforma de Simulación de Código Abierto para Análisis de Problemas Asociados a Vehículos Aéreos no Tripulados”. El

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

aporte de esta tesis es contribuir a nuevos modelos dinámicos de vuelo para el simulador, en particular los modelos dinámicos de vuelo para cuadracopteros.

Objetivo General

Obtener el modelo dinámico de un cuadracoptero e implementarlo en MATLAB utilizando un método de identificación clásico.

Objetivos Específicos

- Seleccionar un método de identificación para obtener el modelo dinámico de un cuadracoptero y aplicarlo al cuadracoptero real que tenemos a disposición en el laboratorio.
- Implementar el modelo dinámico en MATLAB de tal forma que se pueda simular el vuelo del cuadracoptero.
- Evaluar cualitativamente el desempeño del modelo dinámico identificado respecto a vuelos reales del cuadracoptero real.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

1. MARCO TEÓRICO

1.1 Modelamiento e Identificación de Sistemas

Tradicionalmente los modelos dinámicos de vehículos aéreos han sido determinados analíticamente de leyes de la física como las leyes de Newton para cuerpos rígidos. Los parámetros de dichos modelos son determinados a través de costosas y largas pruebas en túneles de viento (Zhang, Li, Wang, & Lu, 2014). Estos métodos, aunque útiles, tienen limitaciones cuando son aplicados en pequeñas aeronaves como los cuadracopteros, debido a diferencias claves como:

- Baja velocidad del aire y números de Reynolds bajos.
- Tasas dinámicas mayores debido a una menor masa.
- Dominancia de las fuerzas de propulsión versus las fuerzas aerodinámicas.
- Diseños asimétricos o atípicos.
- Imposible caracterizar todos los regímenes de vuelo.

Para un cuadracoptero en particular, obtener el modelo dinámico a partir de las leyes de la física es bastante directo. Sin embargo, la formulación matemática obtenida se caracteriza por su complejidad matemática y por su no linealidad, la cual, en términos prácticos se convierte en una pesadilla para el diseño de controladores (Zhang et al., 2014).

Una solución alternativa, es utilizar identificación de sistemas para derivar el modelo dinámico del cuadracoptero. A continuación, se exponen los principales esquemas de identificación dedicados al modelamiento de cuadracopteros.

1.1.1 Modelos basados en la estimación de parámetros de modelos analíticos

Este esquema utiliza un modelo dinámico obtenido a través de las leyes de la física (Burkamshaw, 2010; Pounds, Mahony, & Corke, 2006; Zawiski & Błachuta, 2012). La tarea de identificación se concentra en encontrar el conjunto de parámetros del modelo. Los

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

parámetros del modelo pueden obtenerse directamente midiendo, o pueden ser derivados experimentalmente (Zhang et al., 2014). A continuación, se listan los parámetros que pueden obtenerse por métodos regulares de identificación:

- Parámetros aerodinámicos: rotores, hélices y parámetros aerodinámicos se obtienen por medio de mediciones, cálculos, simulaciones o de referencias.
- Masas y desplazamientos: los componentes de masas y distancias pueden medirse directamente en el cuadracoptero, respecto al plano de rotor.
- La Inercia rotacional se puede obtener directamente midiendo o a través de cálculos.
- Las constantes del motor: el motor puede modelarse de forma simple a través un modelo de primer orden y las constantes extraídas de datos experimentales.

1.1.2 Identificación de modelos lineales locales

Es común en sistemas no-lineales hacer linealizaciones alrededor de un punto de trabajo para deshacerse de las complejidades de los modelos no lineales. La idea detrás de la identificación con varios modelos locales (Gremillion & Humbert, 2010; Ljung, 2010; Wei, Tischler, Schwartz, & Cohen, 2014) es manejar la no linealidad desarrollando modelos locales que representan una buena aproximación en diferentes zonas de trabajo, para luego componer un modelo global a partir de ellos. Frecuentemente, los modelos locales son lineales por lo tanto el método de identificación se denomina modelos locales lineales. Se debe tener en cuenta, que una linealización es una simplificación después de todo, lo cual resulta en un sesgo, o incluso en modelos incorrectos para puntos de trabajo no considerados por los modelos locales (Zhang et al., 2014). Un desafío adicional de este tipo de identificación son las transiciones entre los diferentes modelos de acuerdo al punto de trabajo en el que se encuentre el sistema. Diversas técnicas pueden utilizarse en este tipo de modelos:

- Identificación paramétrica (Ljung, 2010)
- Identificación en el dominio del tiempo (Gremillion & Humbert, 2010)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Identificación en el dominio de la frecuencia (Wei et al., 2014)

1.1.3 Modelos RBF-ARX

Un modelo RBF-ARX (Ma, Li, & Nie, 2016; Peng, 2007) es un modelo no-lineal variante en el tiempo cuya estructura se asemeja a un modelo auto-regresivo ARX. Sus variables independientes son grupos de señales indicando el estado no-lineal del sistema, y los parámetros del modelo pueden ser ajustados lo mejor posible aprovechando las ventajas de las redes neuronales RBF. Las redes de base radial son un tipo de redes de neuronas artificiales que calculan la salida de la función en función de la distancia a un punto denominado centro. Al igual que con los perceptrones multicapa, sirven como aproximadores universales. Debido a los parámetros auto-ajustables, los modelos RBF-ARX no solo tiene excelente desempeño en espacios locales lineales sino también un desempeño global superior (Zhang et al., 2014).

1.1.4 Modelos de caja negra

1.1.4.1 Modelos de redes neuronales

Un modelo de caja negra que utiliza redes neuronales (Sonntag, 2011) para aprender la dinámica del cuadricoptero. Por lo general, los modelos de redes neuronales utilizan 6 redes neuronales para las velocidades lineales en X , Y y Z y las velocidades angulares en *roll*, *pitch* and *yaw*. Las otras variables son integradas (posición/ángulos) y derivadas (aceleraciones) de las velocidades calculadas. Los modelos implementados con redes neuronales son buenos aprendiendo la dinámica del cuadricoptero, sin embargo, requieren de mucho trabajo para generar los datos de entrenamiento y prueba de la red (Zhang et al., 2014).

1.1.4.2 Modelos de base de datos

El principal propósito de las técnicas fundamentadas en bases de datos es aprovechar todas las ventajas de grandes cantidades información adquirida a través de procesos de medición. A través de un análisis exhaustivo de los datos obtenidos a través de procesos de medición, es posible extraer información como características del sistema y regularidades (Zhang et

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

al., 2014). Existen varias corrientes típicas de base de datos que dependen solo en procesos de medición:

- Análisis de componentes principales (PCA)
- Mínimos cuadrados parciales (PLS)
- Control por aprendizaje iterativo (ILC)

1.2 Modelo dinámico de vuelo para un cuadracoptero en configuración X

1.2.1 Momento de Inercia de masa

El momento de inercia de un Objeto en movimiento rotacional juega un papel similar al de la masa en el movimiento translacional. El momento de inercia de masa determina como la velocidad rotacional es afectada por el torque aplicado. Esto depende tanto de la masa como de su distribución al rededor del eje de rotación.

$$\sum \tau = J \dot{\omega} \quad (1)$$

El momento de inercia de masa es muy importante para modelar el vuelo de un cuadracoptero y por eso debemos encontrar la matriz del momento de inercia. Denotaremos la letra J para el momento de inercia de masa y $\dot{\omega}$ para el vector velocidad angular. Una suposición que haremos es acerca de la simetría del cuadracoptero en los tres ejes “x”, “y”, y “z”. También supondremos que el centro de masa del vehículo está en el centro geométrico de los brazos. Con estas suposiciones, podemos afirmar que la matriz J^b es una matriz diagonal como se muestra en la ecuación (2). Los términos J_x y J_y son identicos debido tambien a la simetria.

$$J^b = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix} \quad (2)$$

La estrategia que seguiremos para calcular los términos J_x , J_y y J_z es la siguiente:

- Dividir el cuadracoptero en componentes separados, modelando cada uno de ellos como una forma geométrica simplificada con densidad constante.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

- Medir y pesar cada componente.
- Utilizar el teorema del eje paralelo para determinar la contribución de cada componente al momento inercial de masa sobre cada eje de referencia del vehículo.
- Sumar las inercias de cada componente en cada eje.

El teorema del eje paralelo establece que el momento de inercia de masa de un componente dado en relación a un eje paralelo que pasa por el centro de masa de dicho componente está determinado por la ecuación 3.

$$J_{\text{eje_paralelo}} = J_{COM} + mr^2 \quad (3)$$

Donde J_{COM} es el momento de inercia de un componente individual sobre su propio eje (el cual debe pasar por el centro de masa) paralelo al eje sobre el cual queremos moverlo. En la ecuación 3, m es la masa del componente, y r es la distancia perpendicular entre los ejes paralelos.

1.2.1.1 Momento de inercia de los motores: cilindros solidos ($J_{x,M}$, $J_{y,M}$, y $J_{z,M}$)

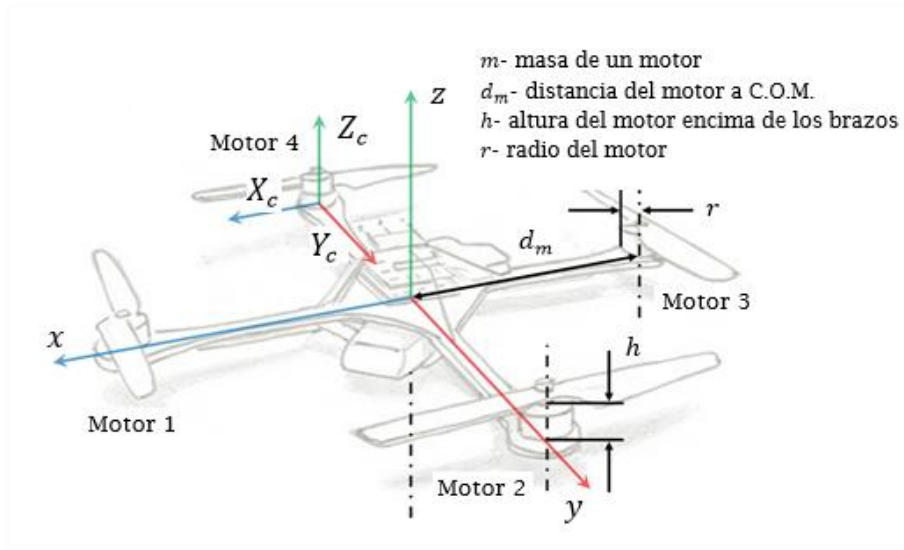


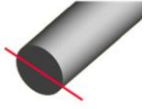
Figura 1 Momento de Inercia de los Motores (Hartman, K. Landis, M. Mehrer, S. Moreno, 2014)

Ecuaciones necesarias:

Cilindro rotando de un extremo

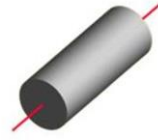
Cilindro rotando sobre su eje central

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22



$$J_{COM} = \frac{1}{4}mr^2 + \frac{1}{3}mh^2$$

Ejes X y Y



$$J_{COM} = \frac{1}{2}mr^2$$

Eje Z

$J_{x,M}$ y $J_{y,M}$ sera

$$J_{x,M} = J_{y,M} = 4 \left[\frac{1}{4}mr^2 + \frac{1}{3}mh^2 + \frac{1}{2}md_m^2 \right] \quad (4)$$

Con respecto al eje Y, usamos la ecuación de un cilindro rotando alrededor de un extremo.

Así, los 4 motores rotan a una distancia $\frac{\sqrt{2}}{2}d_m$. Debido a la simetría el momento de inercia de masa en el eje X será idéntico al del eje Y.

Calculemos ahora el momento de inercia respecto al eje z, $J_{z,M}$. Para esto usaremos la ecuación para un cilindro rotando alrededor de un eje central. Los 4 motores están rotando alrededor de un eje central paralelo al eje z a una distancia d_m . La ecuación 5 presenta el momento de inercia de masa de los 4 motores respecto al eje z.

$$J_{z,M} = 4 \left[\frac{1}{2}mr^2 + md_m^2 \right] \quad (5)$$

1.2.1.2 Momento de inercia del HUB Central: cilindro solido ($J_{x,H}$, $J_{y,H}$, $J_{z,H}$)

En el HUB central está concentrada toda la electrónica del cuadracoptero y su masa debe dar cuenta de ello. Es obvio que garantizar la distribución simétrica de todos los componentes electrónicos es difícil por lo que esta es solo una aproximación.

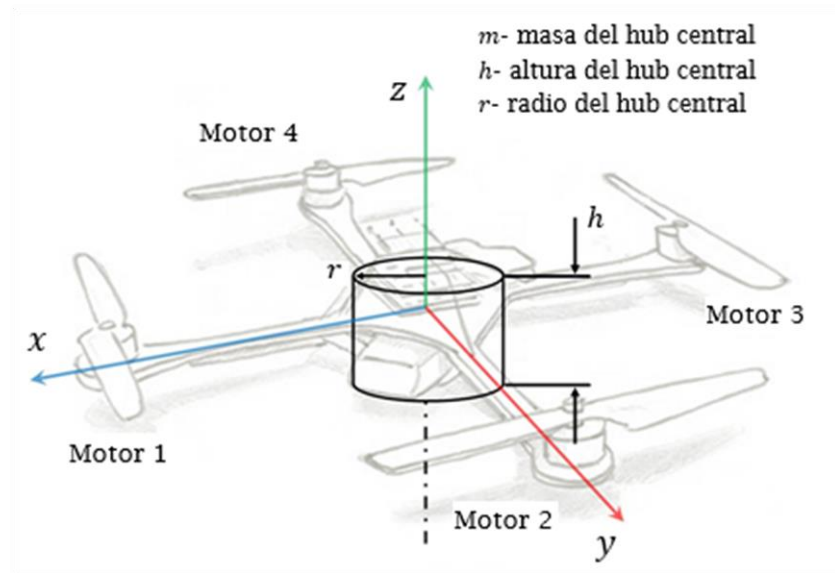
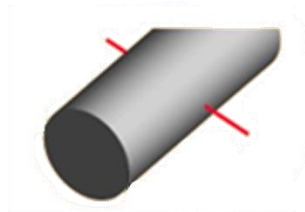


Figura 2 Momento de Inercia del HUB central (Hartman, K. Landis, M. Mehrer, S. Moreno, 2014)

Ecuaciones necesarias:

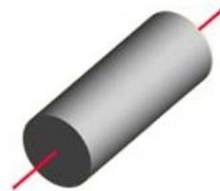
Cilindro rotando alrededor de un eje transversal



$$J_{COM} = \frac{1}{4}mr^2 + \frac{1}{12}mh^2$$

Ejes X y Y

Cilindro rotando alrededor de un eje central



$$J_{COM} = \frac{1}{2}mr^2$$

Eje Z

Para encontrar $J_{x,H}$ y $J_{y,H}$ usamos la ecuación para un cilindro rotando alrededor de un eje transversal. Tanto para $J_{x,H}$, como para $J_{y,H}$ la distancia al eje x o y respectivamente es cero y ambos tendrán la misma magnitud. La ecuación 6 presenta el momento de masa de inercial para HUB central en x y en y.

$$J_{x,H} = J_{y,H} = \left[\frac{1}{4}mr^2 + \frac{1}{12}mH^2 \right] \tag{6}$$

Calculemos ahora para $J_{z,H}$, debemos usar la ecuación de un cilindro alrededor de un eje central. En este caso el eje de rotación del HUB coincide con el eje z, por lo cual la distancia es cero. La ecuación 7 presenta la contribución del momento de masa de inercia para el HUB en el eje z.

$$J_{z,H} = \left[\frac{1}{2} mr^2 \right] \quad (7)$$

1.2.1.3 Momento de inercia de los brazos o booms: cilindros largos ($J_{x,A}$, $J_{y,A}$, $J_{z,A}$)

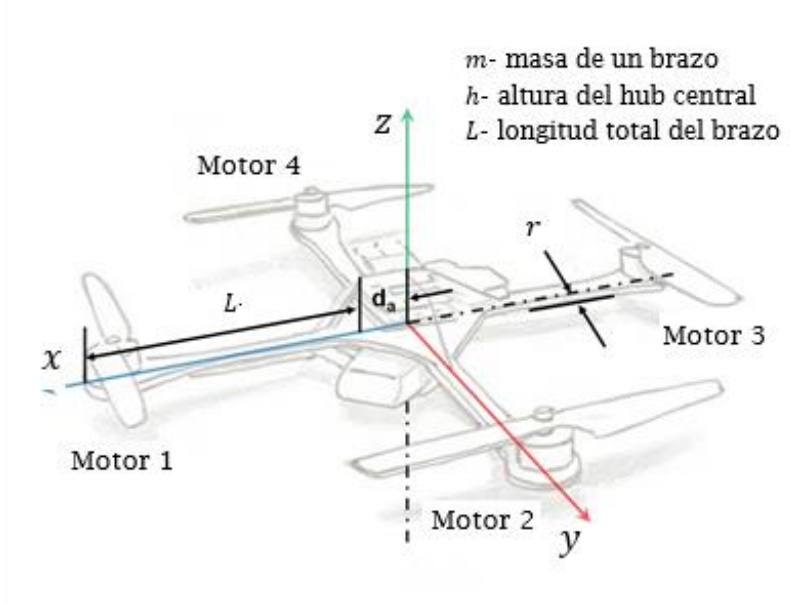
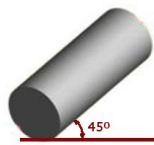


Figura 3 Momento de Inercia de los Brazos del Quadcoptero (Hartman, K. Landis, M. Mehrer, S. Moreno, 2014)

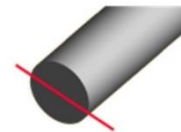
Ecuaciones necesarias:

Cilindro rotando alrededor del eje con un ángulo de 45°



$$J_{COM} = \frac{\sqrt{2}m}{12} (L^2 + 3Lr + 6r^2)$$

Cilindro rotando de un extremo



$$J_{COM} = \frac{1}{4} mr^2 + \frac{1}{3} mL^2$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Ejes X y Y

Eje Z

Para encontrar $J_{x,A}$ y $J_{y,A}$ usamos la ecuación para un cilindro rotando de un eje con ángulo de 45° . Supongamos que estamos encontrando $J_{x,A}$, gracias a la simetría, los 4 brazos contribuyen igualmente al momento de inercia y están a una distancia $\frac{\sqrt{2}}{2}d_A$. Por lo tanto el término asociado con el teorema del eje paralelo será $\frac{1}{2}md_A^2$. De nuevo gracias a la simetría $J_{x,A} = J_{y,A}$ La ecuación 8 presenta el momento inercial de masa para los 4 brazos del cuadracoptero para los ejes X y Y .

$$J_{x,A} = J_{y,A} = 4 \left[\frac{\sqrt{2}m}{12} (L^2 + 3Lr + 6r^2) + \frac{1}{2}md_A^2 \right] \quad (8)$$

Para encontrar $J_{z,A}$ utilizamos la ecuación para un cilindro rotando de un extremo. En este caso los 4 brazos están rotando de un extremo a una distancia d_A del eje z del cuadracoptero, por lo tanto el componente del teorema del eje paralelo es md_A^2 . De nuevo, gracias a la simetría y al hecho de que cada brazo tiene igual longitud el momento inercial de masa total en el eje z de los 4 brazos es 4 veces el momento inercial de uno de los brazos. La ecuación 9 presenta la expresión para $J_{z,A}$.

$$J_{z,A} = 4 \left[\frac{1}{4}mr^2 + \frac{1}{3}mL^2 + md_A^2 \right] \quad (9)$$

1.2.2 Modelo matemático

1.2.2.1 Notación

A continuación presentamos la notación que utilizaremos a lo largo de esta sección:

$${}^b\ddot{V}_{CM|i}$$

En este ejemplo particular, la variable base es la aceleración lineal, o \dot{v} . Como se puede apreciar, la variable tiene dos superíndices y un subíndice para indicar detalladamente que se está describiendo. El superíndice de la izquierda indica que la derivada fue tomada respecto al sistema de referencia local. Por su parte el superíndice de la derecha indica que la aceleración es dada en términos de los componentes vectoriales de sistema de referencia local. Por otro lado, el subíndice $CM|i$, indica que la variable está haciendo referencia al centro de masa con respecto al sistema de referencia inercial.

Otro aspecto importante acerca del modelo matemático es el sistema de coordenada utilizado. El modelo seleccionado y sus convenciones son muy importante, y varían de acuerdo la configuración del cuadraoptero utilizada (“+” o “x”). En esta sección presentamos el modelo matemático para la configuración X. En la Figura 4 se muestra el sistema de coordenadas respecto al bosquejo de un cuadraoptero. Tenga en cuenta que los brazos del cuadraoptero forman ángulos de 45° con los ejes coordenados “X” y “Y” y que la dirección positiva de “X” está entre los motores 1 y 2.

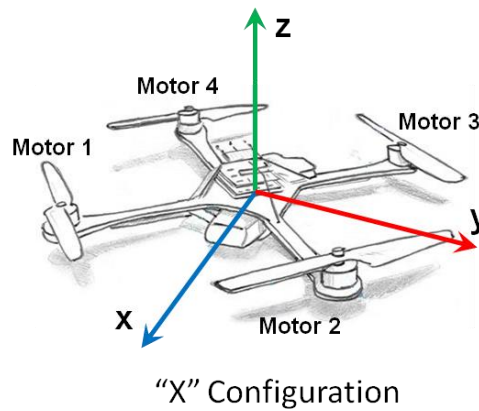


Figura 4 Sistema de coordenadas para cuadraoptero en configuración X

La Figura 5 muestra las etiquetas de los ejes y las direcciones positivas para las rotaciones Pitch, Roll y Yaw. Los ejes en el sistema de referencia inercial están representados por las

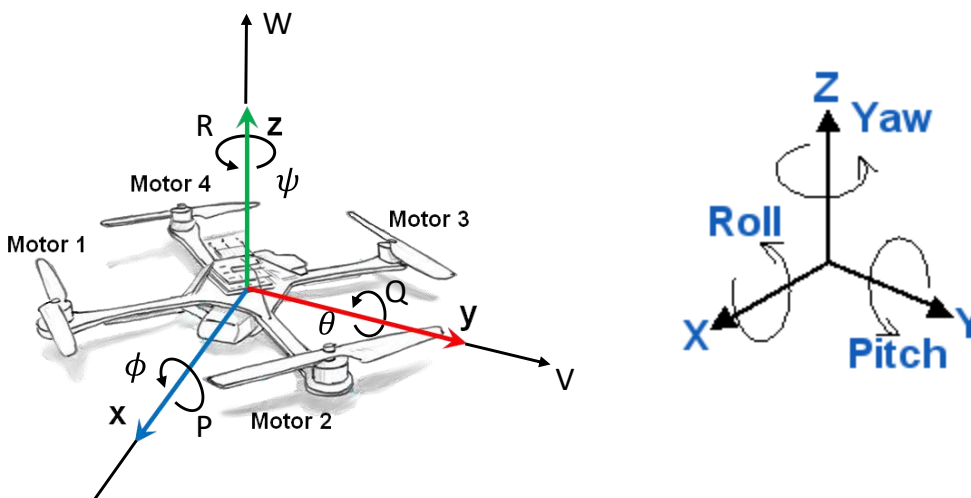


Figura 5 Etiquetas de los Ejes y convenciones positivas (Las flechas indican las direcciones positivas)

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

letras “X”, “Y” y “Z”; mientras que las rotaciones están representadas por ϕ , θ , ψ . En el sistema de referencia local los ejes están etiquetados “U”, “V” y “W”; y los ángulos de rotación se identifican con las letras “P”, “Q” y “R” para las rotaciones Roll, Pitch y Yaw respectivamente.

1.2.2.2 Coeficiente de Empuje

El empuje de los motores es la fuerza responsable por todas las maniobras del cuadracoptero, lo cual la hace fundamental en simulación y diseño de controladores. El empuje, T , proporcionado por un sistema motor-hélice puede ser calculado como lo indica la ecuación 10.

$$T = C_T \rho A_r r^2 \omega^2 \quad (10)$$

Donde C_T es el coeficiente de empuje para un motor específico, ρ es la densidad del aire, A_r es la sección de área efectiva a la rotación de la hélice, r es el radio del motor, y ω es la velocidad angular del motor. Para simplificar el modelo utilizaremos una sola constante c_T para expresar la relación entre el empuje y la velocidad angular, como lo muestra la ecuación (11).

$$T = c_T \omega^2 \quad (11)$$

El empuje proporcionado por el sistema motor-hélice proporciona una fuerza perpendicular al plano U-V en el sistema de referencia local en la dirección positiva de W. El coeficiente c_T debe obtenerse de forma experimental.

1.2.2.3 Coeficiente de Torque

La rotación del sistema motor-hélice genera un torque que tiene como consecuencia una rotación en *yaw*. Así el torque, Q , creado por el sistema motor-hélice puede expresarse de forma similar al empuje. La ecuación 12 define el torque como una función cuadrática de la velocidad de rotación ω .

$$Q = c_Q \omega^2 \quad (12)$$

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

En este caso Q es el torque creado por el motor y c_Q el coeficiente de torque para el sistema motor-hélice. Dicho torque proporciona una fuerza que hace que el cuadraoptero gire alrededor del eje W. El coeficiente c_Q debe obtenerse de forma experimental.

1.2.2.4 Planteamiento inicial del sistema matricial

La ecuación 13 describe el empuje y los torques en el sistema como una operación de matrices. Todas las etiquetas en 13 han sido discutidas anteriormente con excepción de d_x . La constante d_x es la distancia entre los motores y los ejes de rotación en el sistema de referencia local. Para una configuración X, d_x es $d * \sin(45^\circ)$, y d es la distancia entre el centro del cuadraoptero y el motor.

$$\begin{bmatrix} \Sigma T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ -d_x c_T & d_x c_T & d_x c_T & -d_x c_T \\ -d_x c_T & -d_x c_T & d_x c_T & d_x c_T \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix} \quad (13)$$

1.2.2.5 Fuerzas Giroscópicas (Precesión)

Hay un conjunto adicional de fuerzas que no hemos considerado anteriormente. La precesión giroscópica es un fenómeno que ocurre cuando el eje de rotación de un cuerpo rotante se cambia, los resultados son normalmente poco intuitivos. Las fuerzas giroscópicas resultantes en el *frame* son gobernadas por la inercia de cada componente del motor rotando (J_m), las tasas de cambio en P (*roll*) y Q (*pitch*), además de la velocidad de rotación del sistema motor-hélice (ϖ_i). La ecuaciones 14 y 15 definen los torques giroscópicos creados por los movimientos en *pitch* y *yaw* del cuadraoptero.

$$\tau_{\phi_{gyro}} = J_m Q \left(\frac{\pi}{30} \right) (\varpi_1 - \varpi_2 + \varpi_3 - \varpi_4) \quad (14)$$

$$\tau_{\theta_{gyro}} = J_m P \left(\frac{\pi}{30} \right) (-\varpi_1 + \varpi_2 - \varpi_3 + \varpi_4) \quad (15)$$

El término $\pi/30$ corresponde a la conversión de RPM a radianes.

1.2.2.6 Construcción matricial completa

Al incluir las fuerzas giroscópicas en 13, es posible plantear un vector que considere los todos los momentos actuando sobre el cuadraoptero. La ecuación 16 define el vector

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

momento $M_{A,T}^b$, el cual considera todos los momentos actuando en el cuadracoptero resultado de la aerodinámica, empuje y torques.

$$M_{A,T}^b = \begin{bmatrix} -d_x c_T \varpi_1^2 + d_x c_T \varpi_2^2 + d_x c_T \varpi_3^2 - d_x c_T \varpi_4^2 + J_m Q \left(\frac{\pi}{30}\right) (\varpi_1 - \varpi_2 + \varpi_3 - \varpi_4) \\ -d_x c_T \varpi_1^2 - d_x c_T \varpi_2^2 + d_x c_T \varpi_3^2 + d_x c_T \varpi_4^2 + J_m P \left(\frac{\pi}{30}\right) (-\varpi_1 + \varpi_2 - \varpi_3 + \varpi_4) \\ -c_Q \varpi_1^2 + c_Q \varpi_2^2 - c_Q \varpi_3^2 + c_Q \varpi_4^2 \end{bmatrix} \quad (16)$$

El cuadracoptero también experimenta fuerzas que actúan sobre él como la gravedad y el empuje de lo rotores. La ecuación 17 presenta el vector $F_{A,T}^b$, que representa todas las fuerzas actuando en el cuadracoptero debido a la aerodinámica y al empuje. Como es evidente de la ecuación 17, todas las fuerzas se asumen orientadas en la dirección W del sistema de referencia local.

$$F_{A,T}^b = \begin{bmatrix} 0 \\ 0 \\ c_T (\varpi_1^2 + \varpi_2^2 + \varpi_3^2 + \varpi_4^2) \end{bmatrix} \quad (17)$$

Es preciso aclarar que hay efectos aerodinámicos adicionales como la oscilación de las hélices y la fricción con el viento que no han sido considerados en este modelo.

1.2.2.7 Ecuaciones de estado

La ecuación 18 presenta la velocidad angular. Esta ecuación describe la tase de cambio en *roll* (P), *pitch* (Q), y *yaw* (R) del cuadracoptero teniendo en cuenta la inercia, la velocidad angular, y el momento aplicado por el sistema motor-hélice. ${}^b \dot{\omega}_{b|i}^b$ es la aceleración angular en cada eje del sistema de referencial local con respecto al sistema de referencia inercial. J^b es la matriz inercial discutida en la sección 1.2.

$${}^b \dot{\omega}_{b|i}^b = (J^b)^{-1} [M_{A,T}^b - \Omega_{b|i}^b J^b \omega_{b|i}^b] = \begin{bmatrix} \dot{P} \\ \dot{Q} \\ \dot{R} \end{bmatrix} \quad (18)$$

$\Omega_{b|i}^b$ es una matriz producto-cruzado para la velocidad rotacional. La forma de la matriz se muestra en la ecuación 19

$$\Omega_{b|i}^b = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix} \quad (19)$$

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Nuevamente aquí P, Q y R son respectivamente las tasas de rotación sobre los ejes X, Y y Z. Finalmente, el termino $\omega_{b|i}^b$ es la velocidad rotacional del cuadracoptero en el sistema de referencia local y es definido directamente por P, Q y R como lo muestra la ecuación 20.

$$\omega_{b|i}^b = \begin{bmatrix} P \\ Q \\ R \end{bmatrix} \quad (20)$$

La siguiente ecuación de estado corresponde a la Ecuación Cinemática de Euler (Ecuación 21), la cual nos permite determinar la rata de cambio de los ángulos de Euler en el sistema de referencia inercial.

$$\dot{\Phi} = H(\Phi)\omega_{b|i}^b = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (21)$$

En forma matricial la ecuación cinemática de Euler toma la forma mostrada en 22. Donde t(), s(), y c() corresponden al operador tan(), sin() y cos() respectivamente.

$$\dot{\Phi} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & t(\theta) s(\phi) & t(\theta) c(\phi) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)/c(\theta) & c(\phi)/c(\theta) \end{bmatrix} \begin{bmatrix} P \\ Q \\ R \end{bmatrix} = H(\Phi)\omega_{b|i}^b \quad (22)$$

La ecuación de estado de velocidad se presenta en 23. Esta ecuación describe la aceleración del centro de masa del modelo de cuerpo rígido del cuadracoptero con base a las fuerzas de aceleración actuando sobre él.

$${}^b\dot{v}_{CM|i}^b = \left(\frac{1}{m}\right) F_{A,T}^b + g^b - \Omega_{b|i}^b \omega_{CM|i}^b = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix} \quad (23)$$

Aquí, ${}^b\dot{v}_{CM|i}^b$ es la aceleración lineal del centro de masa en el sistema de referencia local con respecto al sistema de referencia inercial. La variable m es la masa total del cuadracoptero, mientras que g^b es la aceleración de gravedad en el sistema de referencia local. La ecuación

24

$$g^b = \begin{bmatrix} c(\theta) c(\psi) & c(\theta) s(\psi) & -s(\theta) \\ (-c(\phi) s(\psi) + s(\phi) s(\theta) c(\psi)) & (c(\phi) c(\psi) + s(\phi) s(\theta) s(\psi)) & s(\phi) c(\theta) \\ (s(\phi) s(\psi) + c(\phi) s(\theta) c(\psi)) & (-s(\phi) c(\psi) + c(\phi) s(\theta) s(\psi)) & c(\phi) c(\theta) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ g^i \end{bmatrix} \quad (24)$$

Usando la ecuación 23 es posible encontrar la aceleración lineal de cuadracoptero en X, Y y Z en el sistema de referencia local.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Finalmente, la última ecuación de estado en ser introducida es la ecuación de posición (Ver ecuación 25), la cual describe la velocidad del centro de masa del cuadricoptero en el sistema de referencia inercial.

$${}^i\dot{p}_{CM|i} = \begin{bmatrix} c(\theta) c(\psi) & c(\theta) s(\psi) & -s(\theta) \\ (-c(\phi) s(\psi) + s(\phi) s(\theta) c(\psi)) & (c(\phi) c(\psi) + s(\phi) s(\theta) s(\psi)) & s(\phi) c(\theta) \\ (s(\phi) s(\psi) + c(\phi) s(\theta) c(\psi)) & (-s(\phi) c(\psi) + c(\phi) s(\theta) s(\psi)) & c(\phi) c(\theta) \end{bmatrix} v_{CM|i}^b = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad (25)$$

La ecuación 25 nos permite determinar la velocidad del cuadricoptero en las direcciones X, Y y Z del sistema de referencia inercial.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2. METODOLOGÍA

En este capítulo se presentan los materiales y métodos utilizados para obtener el modelo dinámico del cuadrapterio. La Sección 2.1 introduce la metodología de trabajo aplicada durante el proyecto y argumenta sobre la metodología de identificación seleccionada. La Sección 2.2 presenta los equipos utilizados, en particular se hace una descripción detallada del cuadrapterio y sus componentes principales. En la sección 2.3 presenta el modelo de simulación en Matlab y el sistema de visualización.

2.1 Metodología de Identificación

En la sección 1.1 se expusieron diversas técnicas para la identificación del modelo dinámico de un cuadrapterio. Cada una de ellas presenta sus ventajas y desventajas desde el punto de vista técnico y matemático. Después de evaluar las fortalezas del equipo de trabajo y los recursos a nuestra disposición, se decidió por una identificación de parámetros de modelos analíticos.

Como se expuso en la sección 1.1.1, los modelos basados en la estimación de parámetros de modelos analíticos, parten de un modelo matemático obtenido a partir de las leyes y principios de la física (Mahony, Kumar, & Corke, 2012). Los parámetros de modelo pueden obtenerse directamente midiendo, o pueden ser identificados experimentalmente. Nuestra primera tarea fue entonces encontrar un modelo analítico de referencia a partir del cual empezar a hacer la identificación de parámetros. En este contexto, producto de investigación bibliográfica, encontramos un modelo analítico de referencia en código abierto llamado *Quad-Sim-master* (Hartman, K. Landis, M. Mehrer, S. Moreno, 2014). Este proyecto fue desarrollado en la Universidad de Drexel, en los Estados Unidos. *Quad-Sim-master* proporciona el código en Matlab para un cuadrapterio configurable en relación a su geometría y distribución de masa. Además el proyecto cuenta con utilidades para identificar las constantes del sistema motor-hélice.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Después de estudiar el modelo y cómo funcionaba nos dimos cuenta que el modelo utilizado para modelar el subsistema **motor-hélice** era bastante simple, además dejaba por fuera la dinámica de los ESCs. Por esta razón, y teniendo en cuenta que las fuerzas de empuje son componentes principales en la dinámica de los cuadracopteros, se decide caracterizar el subsistema **motor-hélice-ESC**, por medio de un modelo dinámico lineal o no-lineal de mayor precisión.

2.2 Materiales y Equipos

2.2.1 Componentes de un Cuadracoptero

Durante este trabajo de grado contamos con un cuadracoptero de la línea de automatización y robótica del grupo de investigación AEyCC. A continuación, se describen los componentes principales del cuadracoptero utilizado para la obtención del modelo.

2.2.1.1 Motores DC sin escobillas

El cuadracoptero cuenta con 4 motores DC sin escobillas de referencia NTM Prop Drive Series 28-30 1100KV. Los motores son alimentados con corriente alterna trifásica, son de alta eficiencia. En particular estos motores alcanzan 1100 RPM por cada voltio de alimentación, y están diseñados para alimentarse entre 12V y 15V. La Figura 6 Características del Motor y Geometría **Error! No se encuentra el origen de la referencia.** muestra los parámetros suministrados por el fabricante y la geometría del motor.

Kv(rpm/v)	1100
Weight (g)	68
Max Current(A)	26
Resistance(mh)	0
Max Voltage(V)	15
Power(W)	385
Shaft A (mm)	3
Length B (mm)	30
Diameter C (mm)	28
Can Length (mm)	19
Total Length E (mm)	45

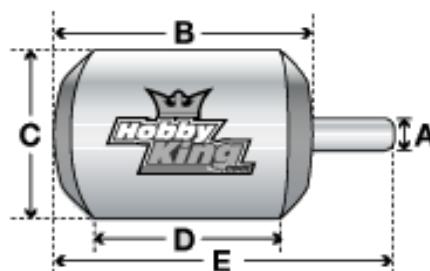


Figura 6 Características del Motor y Geometría

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.2.1.2 Controlador de Velocidad Electrónico (ESC)

Los ESCs son dispositivos electrónicos que realizan el control de velocidad sobre los motores sin escobillas. Para esto los ESC reciben una señal de PWM que establece la velocidad deseada el motor. El ESC produce la señal trifásica DC que alimenta el motor. Las características principales de un ESC comprenden: la corriente constante máxima suministrada, la corriente transitoria máxima, la compatibilidad de voltaje en términos del número de celdas de batería soportadas (Celdas *Lipo* o *NiMH*), la frecuencia y ciclo de trabajo de la señal PWM de entrada, peso y tamaño. Algunos ESC tienen además un circuito de eliminación de batería (BEC) que proporciona una fuente de alimentación para los dispositivos electrónicos. El cuadricoptero que utilizamos en este trabajo utiliza ESC de referencia *Turnigy Plush 30A* con BEC. La Figura 7 **Error! No se encuentra el origen de la referencia.** presenta las características del ESC y su geometría.

Const Current (A)	30
Burst Current (A)	40
BEC Mode	Linear
BEC	5V @ 2A
Lipo Cells	2 @ 4
NiMH Cells	5 @ 12
Weight (g)	25
Length A (mm)	45
Width B (mm)	24
Thick C (mm)	11

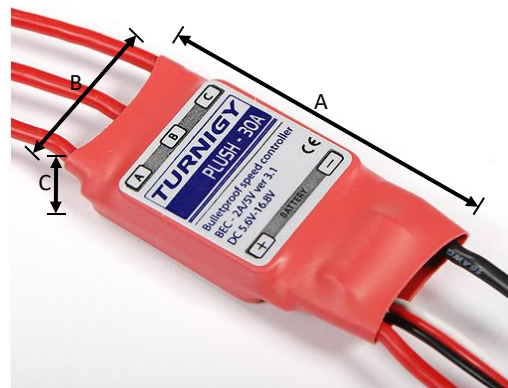


Figura 7 Características del ESC y geometría

2.1.1.3 Hélices

Las hélices son las encargadas de convertir la potencia suministrada por el motor en el empuje necesario para el vuelo. Las hélices varían levemente en su geometría y los fabricantes por lo general suministran solo tres parámetros característicos: el peso, el diámetro y el pitch. Estos dos últimos parámetros se proporcionan generalmente en pulgadas. El pitch, o paso de la hélice, se define como la distancia recorrida por la hélice en una revolución. El cuadricoptero utiliza 4 hélices del tipo *Slowfly*, fabricadas en Nylon, con un diámetro de 11 pulgadas y un pitch de 4.5. La Figura 8 **Error! No se encuentra el origen**

de la referencia. muestra las características de la hélice usada en el cuadricoptero y su geometría.

Length (Inch [X])	11
Pitch (Inch [Y])	4.5
Weight (g)	16

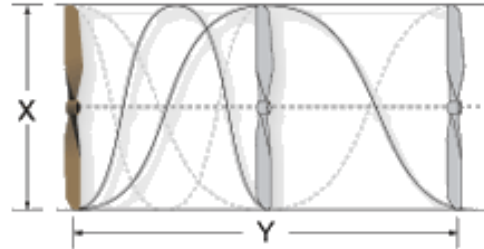


Figura 8 Características de la Hélice y Geometría

2.1.1.4 Batería

La fuente de alimentación de motores, ESCs, y demás dispositivos electrónicos del cuadricoptero es una batería. La tecnología de batería más común utilizada en cuadricopteros es la de litio-ion polímero, gracias a su peso, tamaño y capacidad comparado a otras tecnologías. El cuadricoptero usado en este trabajo utiliza una batería con referencia ZIPPY Compact 4500mAh. Esta referencia se caracteriza por su tamaño compacto y alta capacidad de descarga. La Figura 9 Características y Geometría de la Batería ZIPPY Compact 4500mAh **Error! No se encuentra el origen de la referencia.** presenta las características de la batería Delegadas del suministro de potencia del sistema, de las cuales se deben conocer la cantidad de amperios-hora, voltaje, configuración interna, descarga, peso y dimensiones.

Capacity(mAh)	4500
Config (s)	4
Discharge (c)	35
Weight (g)	496
Max Charge Rate (C)	5
Length-A(mm)	159
Height-B(mm)	46
Width-C(mm)	38

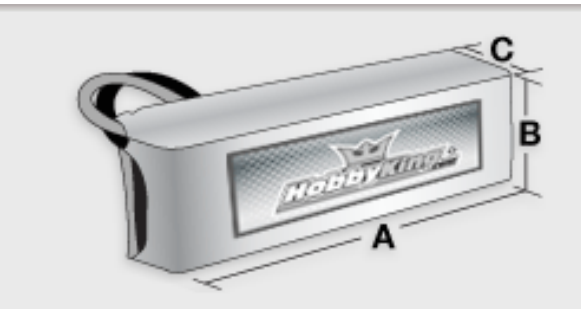


Figura 9 Características y Geometría de la Batería ZIPPY Compact 4500mAh

2.1.1.5 Controlador de vuelo

Es el cerebro del cuadricoptero, gobierna todo el sistema, toma decisiones autónomas de acuerdo a unas características programadas, posee una unidad central de proceso (CPU),

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

memoria, sensores y periféricos de entrada y salida. El controlador de vuelo utilizado por el cuadraoptero es un *HKPilot Mega 2.7* corriendo el software de *ardupilot APM v2.6*.

Algunas características del *HKPilot Mega 2.7* son:

- Brújula digital marca *Honeywell HMC5883L-TR*
- Acelerómetro y giroscopio marca *Invensense MPU-6000*
- Sensor barométrico de presión marca *Measurement Specialties MS5611-01BA03*.
- Microcontroladores de 8 bits *Atmel ATMEGA2560* para control de vuelo y microcontrolador *ATMEGA32U-2* para comunicación USB.
- Puertos de expansión para medición de potencia, GPS, brújula externa, radio telemetría, OSD y sensores de velocidad del aire.
- Peso 32g
- Dimensiones 44mm x 70mm x 15mm.

La Figura 10 **Error! No se encuentra el origen de la referencia.** presenta el controlador de vuelo *HKPilot 2.7* y las diferentes interfaces I/O de las que dispone.



Figura 10 Interfaces I/O HKPilot MEGA 2.7

2.1.1.6 Frame

Estructura rígida que sirve de soporte a los motores y demás elementos electrónicos que componen el cuadraoptero. Los *frames* vienen en diversos tamaños y son fabricados en

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

diversos materiales entre los que se destacan: aluminio, fibra de vidrio, fibra de carbono, plástico, e incluso madera. Los *frames* de fibra de carbono son los más populares gracias a su resistencia y bajo peso. El *frame* utilizado es el *Turnigy Talon V2*. Es un *frame* de fibra de carbono con algunos elementos en aluminio. La Figura 11 **¡Error! No se encuentra el origen de la referencia.** presenta el *frame* y sus principales dimensiones.

222.1.1.8 GPS

Es un sistema que permite determinar en todo el mundo la posición de un objeto con una precisión de hasta centímetros, funciona mediante una red de 24 satélites con trayectorias sincronizadas que orbitan sobre el planeta tierra, cada satélite transmite su posición y tiempo, el receptor mide la diferencia de los tiempos de estas señales y se calcula la posición mediante la intersección de distancias (Sturm, n.d.).

Para nuestros experimentos utilizamos un módulo GPS *Ublox Neo-7M GPS*. Este módulo también incorpora una brújula digital *HMC5883I*, el cual se muestra en la Figura 12 **¡Error! No se encuentra el origen de la referencia..**

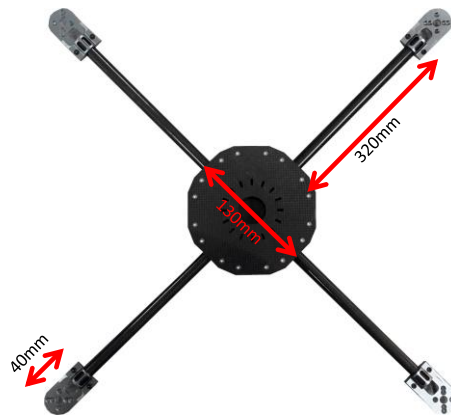


Figura 11 Frame Turnigy Talon V2

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

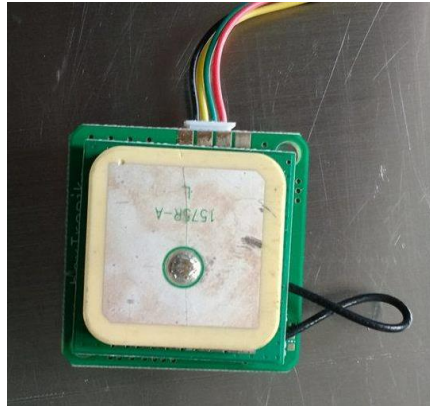


Figura 12 GPS Ublox Neo-7M GPS

2.1.1.9 Radio Control

El radio control permite al usuario volar el cuadracoptero desde una distancia considerable. Es normal que un radio control tenga alcances de un poco más de un kilómetro, dependiendo de la frecuencia y la potencia de transmisión. Los comandos que el usuario realiza a través del radio control se envían al cuadracoptero como el porcentaje de dureza de una señal de PWM. Las señales controladas por el usuario comprenden el nivel de empuje deseado, las desviaciones de los ángulos *pitch*, *roll* y *yaw*, además del estado de algunos interruptores. Cada señal enviada ocupa un canal y los radios tiene un número máximo de canales que pueden transmitir al cuadracoptero.

Para el control del cuadracoptero durante los experimentos se usó un radio control *FrSky Taranis X9D Plus*. El *Taranis* es un radio control 100% configurable y *open-source*, lo cual implica, que el firmware puede ser personalizado a las necesidades del usuario. Otra ventaja de este control es su capacidad para visualizar información de telemetría directamente en la pantalla gráfica. Soporta hasta 16 canales de información. La Figura 13 muestra el *Taranis* y sus diferentes elementos de control.

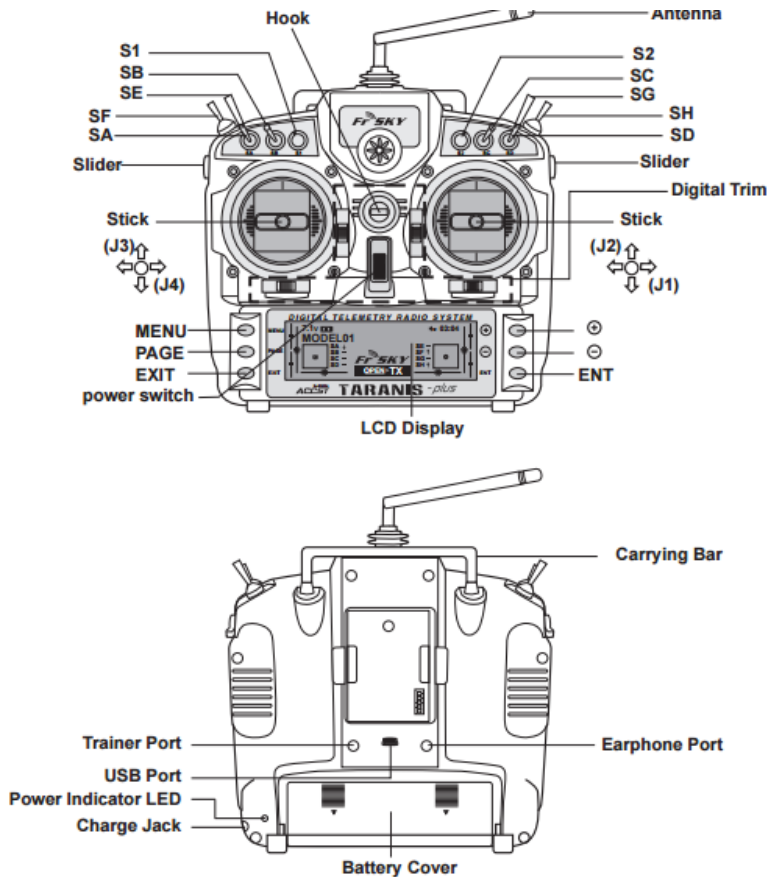
2.1.1.10 Radio Receptor

El radio receptor es la contraparte del radio control en el cuadracoptero. Este dispositivo recibe las órdenes del radio control y se las comunica al controlador de vuelo. Algunos receptores permiten además sensor variables durante el vuelo y transmitir las al radio

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

control para mantener informado al piloto sobre su estado. Es importante anotar que debe existir una compatibilidad entre el radio control y el radio receptor para que la comunicación pueda funcionar. Las modulaciones utilizadas en la comunicación pueden variar de una fabricante (modulaciones propietarias en algunos casos) a otro haciendo que solo dispositivos de un mismo fabricante sean compatibles entre sí. También es importante anotar que para que el radio control y el receptor hablen por primera vez, es necesario seguir un procedimiento de acoplamiento.

El radio receptor Frsky X8R, fue el receptor usado durante este trabajo. Es un dispositivo bastante versátil de 8 canales de salida convencionales WPM, 16 canales de salida a través de una interfaz SBUS, un puerto bidireccional SmartPort para conexión con sensores asociados a telemetría. La Figura 14 **Error! No se encuentra el origen de la referencia.** muestra los puertos y estado de los LEDs del receptor X8R.



(Switch Default Settings)

- SA: 3 positions, Short Lever
- SB: 3 positions, Long Lever
- SC: 3 positions, Long Lever
- SD: 3 positions, Short Lever
- SE: 3 positions, Short Lever
- SF: 2 positions, Long Lever
- SG: 3 positions; Short lever
- SH: 2 positions, Momentary; Long Lever

You can choose the Switch and define its positions in the Mixer menu.

Figura 13 Radio Control FrSky Taranis

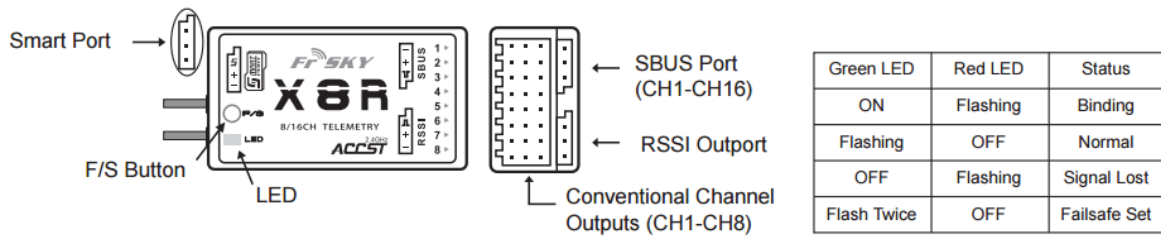


Figura 14 Radio Receptor FrSky X8R

2.1.1.11 Radio telemetría

La telemetría permite al cuadricoptero enviar información de los sensores a bordo a una estación de control en tierra. También es posible controlar el cuadricoptero a través de la radio telemetría. El sistema de radio telemetría jugó un papel muy importante en el desarrollo de este trabajo ya que permitió capturar trazas del estado de los sensores y las señales de control sobre los motores del cuadricoptero durante vuelos reales.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La telemetría usada son transceptores *HKPilot* de 915 MHz con 100mW de potencia. La Figura 15 **Error! No se encuentra el origen de la referencia.** muestra uno de los dos radios de telemetría.



Figura 15 Radio telemetría HKPilot V2

2.1.2 Sistema electrónico de un cuadricoptero

El siguiente diagrama de bloques representa la idea general de las conexiones de los elementos anteriormente descritos.

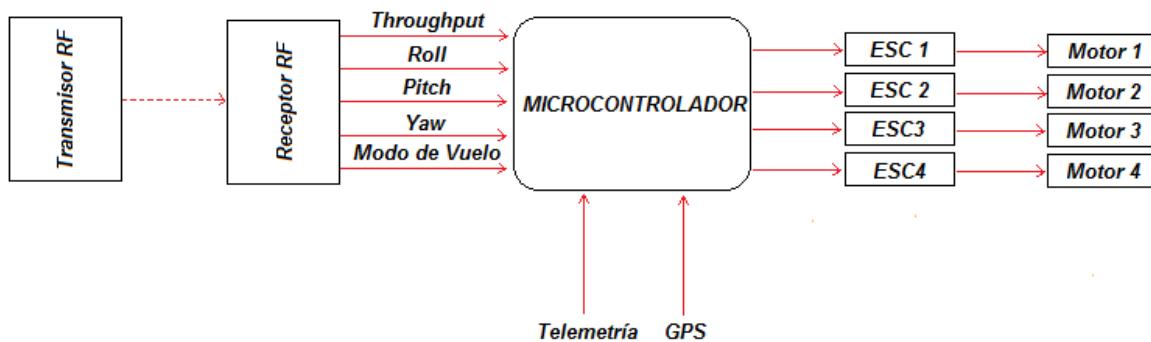


Figura 16 Plano de conexiones del cuadricoptero

Se observa que el radio control envía de manera inalámbrica los comandos al cuadricoptero, en el cuadricoptero el radio receptor recibe los comandos del piloto. El receptor se conecta con el controlador de vuelo y proporciona al microcontrolador la información de *thrust*, *roll*, *pitch*, *yaw* y modo de vuelo necesaria para la correcta navegación. Además de ello, el micro recibe también datos de la telemetría y el GPS. Con esta información, dicho dispositivo genera unas señales de control para cada ESC con el fin de revolucionar los motores según las órdenes recibidas.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

2.2 Caracterización del subsistema motor-hélice-ESC

Para obtener el modelo dinámico del subsistema motor-hélice-ESC del cuadricoptero, se midieron las variaciones de las magnitudes RPM y el empuje del motor con respecto a variaciones a la señal PWM aplicada al motor (Modulación por ancho de pulso) conocida y también cambiante, para ello se desarrolló un programa en el microcontrolador *Arduino* (usando la placa *Arduino-Due*), el cual se dividió en tres partes fundamentales: envío de señal PWM, lectura de empuje y lectura de RPM. A continuación se describe cada fragmento de código con sus respectivos diagramas de flujo.

Señal de salida PWM:

Este programa se dividió básicamente en dos partes: calibración del ESC y variación de velocidad del motor. Posteriormente se detalla cada una.

Calibración ESC: Para el correcto funcionamiento del controlador electrónico de velocidad, se tuvo se cuenta que la primera señal que debía recibir era el tiempo en alto máximo (2000us) con el cual el motor alcanzaría su máxima velocidad, y luego de 2 segundos (tiempo muerto mientras el ESC se configura con dicha señal) se enviaría el tiempo en alto mínimo (700us, pues el controlador no reconoce valores menores a éste) equivalente a la velocidad más baja (0 RPM).

Para el envío de las señales se hizo uso de la librería de *Arduino* para servomotores, en la que se manejan números entre 0 y 180, lo cual se traduce en posiciones angulares en un servo, pero que para este caso dichos valores fueron convertidos en tiempos en alto (siendo 0 equivalente a 700us y 180 a 2000us) con el fin de escribirlos sobre el puerto correspondiente provocando así la variación de la velocidad en el motor.

Cambio de velocidad del motor: Por razones de seguridad, el motor trabajaría sólo hasta el 50% de su velocidad máxima, debido a esto, el valor más alto a procesar con la librería del servo sería igual a 90 (para luego ser transformado en tiempo); además se planearon unos ciclos de incremento y decremento de dicha variable en pasos de a 8, 16, 32, 64 y 90; con

un intervalo de tiempo definido (15 segundos); en las siguientes gráficas se puede apreciar con mayor claridad lo anteriormente mencionado.

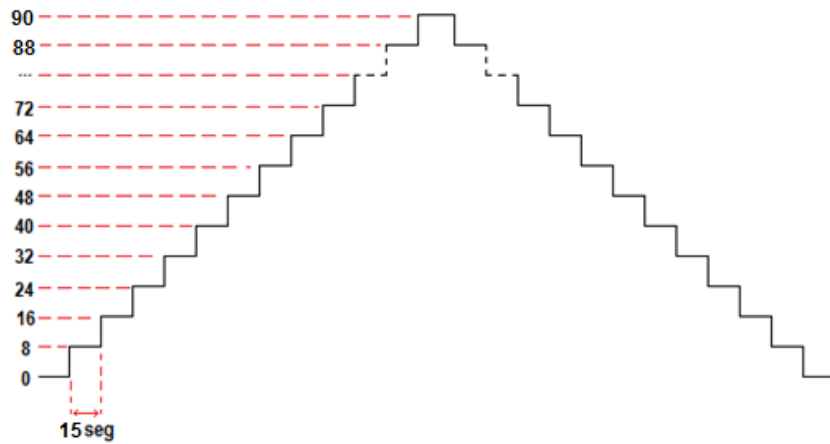


Figura 17 Curva de excitación del Motor con incrementos de 3% en la señal de PWM

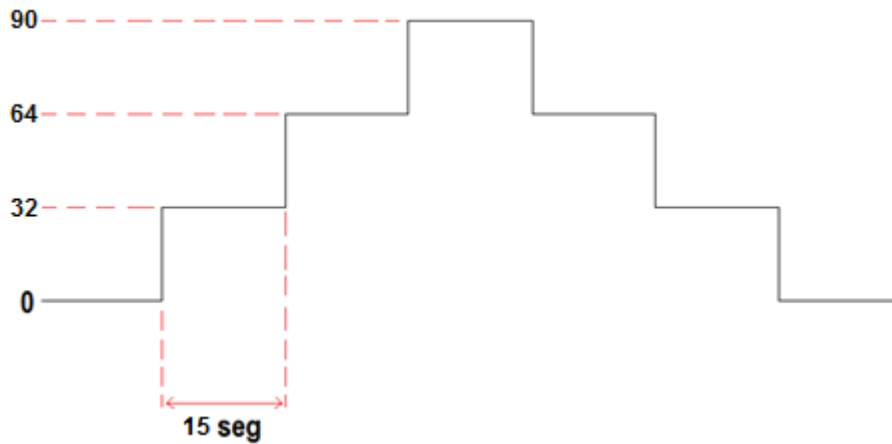


Figura 18 Curva de excitación del Motor con incrementos de 12% en la señal de PWM

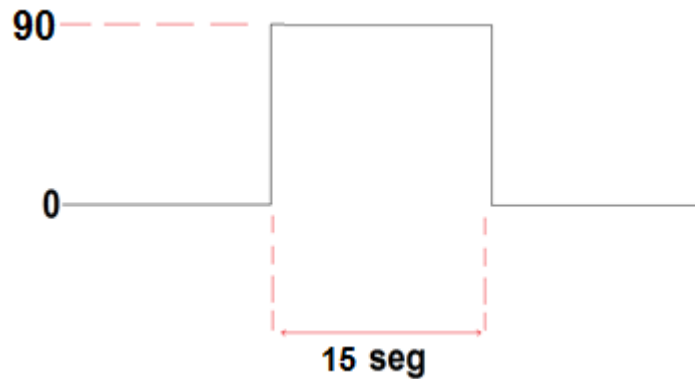


Figura 19 Curva de excitación del Motor con escalon de 35% en la señal de PWM

Asimismo, se realizó el programa para que la señal PWM empezara a cambiar con pasos de a 8, y luego de terminarlos, continuar con los de 16 y así sucesivamente; por otra parte, también se creó un control de repeticiones, el cual ejecutaría un ciclo completo de un mismo valor en varias ocasiones, 2 para este caso. Las Figuras Figura 17, Figura 18 y Figura 19 **¡Error! No se encuentra el origen de la referencia.¡Error! No se encuentra el origen de la referencia.¡Error! No se encuentra el origen de la referencia.** presentan ejemplos de las señales de excitación del motor. La Figura 20 presenta el diagrama de flujo correspondiente al programa de generación de la señal de PWM.

Lectura de empuje del motor:

Para este experimento, se utilizó una pesa *Lexus Prisma SN: YS146121*, la cual se comunicaba haciendo uso del protocolo serial RS232 a través de un DB9, enviando el peso codificado en paquetes de 11 bytes en formato hexadecimal, precedidos por FF 42 (los cuales siempre se enviaban para reconocer el comienzo de un nuevo dato) mandando un total de 13 bytes. La Figura 21 muestra una captura de pantalla del programa X-CTU donde se ilustran los datos de salida de la báscula para un peso de 1037.0g.

Así, el orden de los números para la interpretación del peso del objeto se presenta en la Figura 22. Note que entre los datos 21 y 22 existe una coma, siendo el último número la parte decimal del peso.

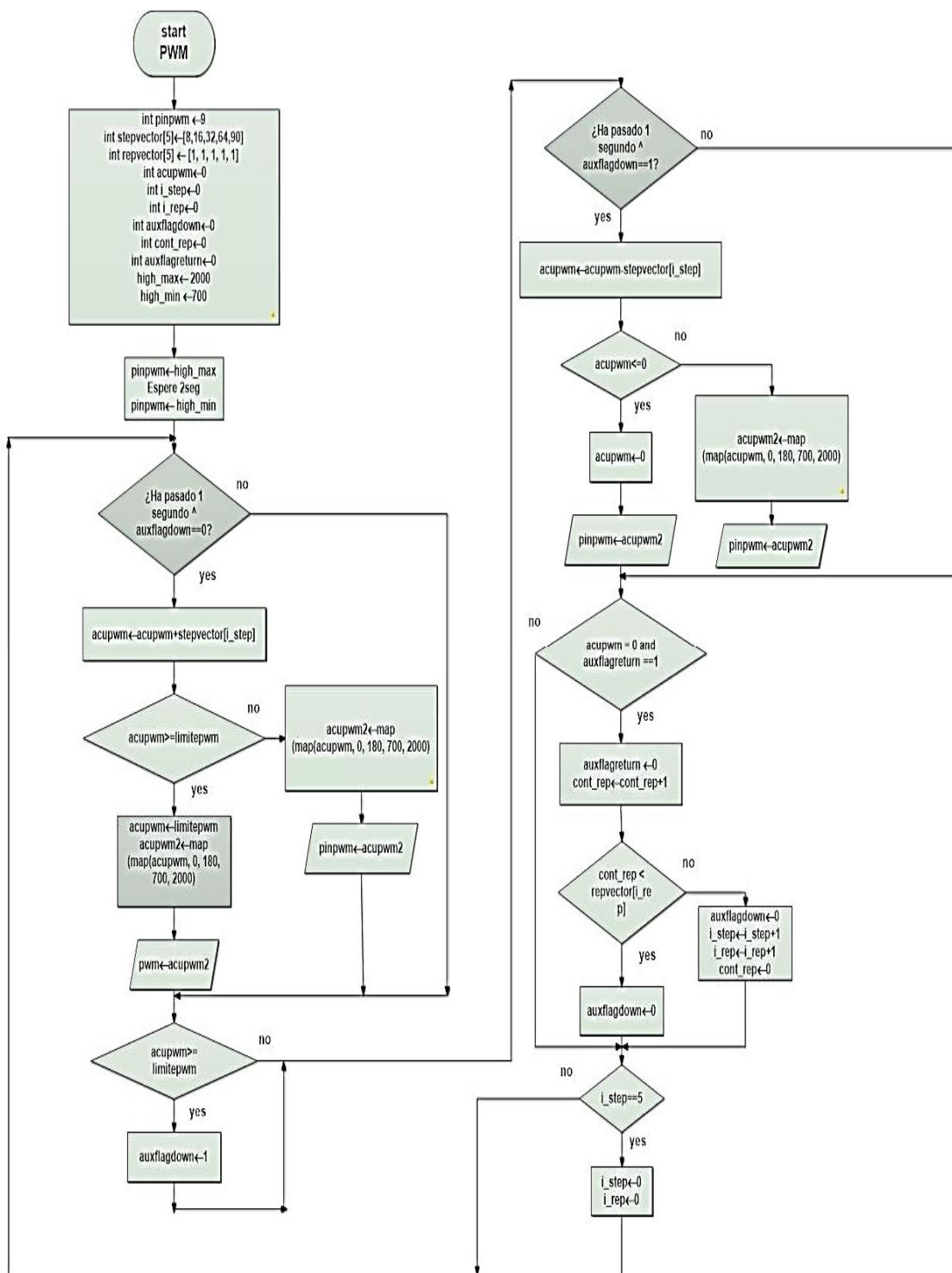


Figura 20 Diagrama de flujo del programa de generación de la señal de PWM

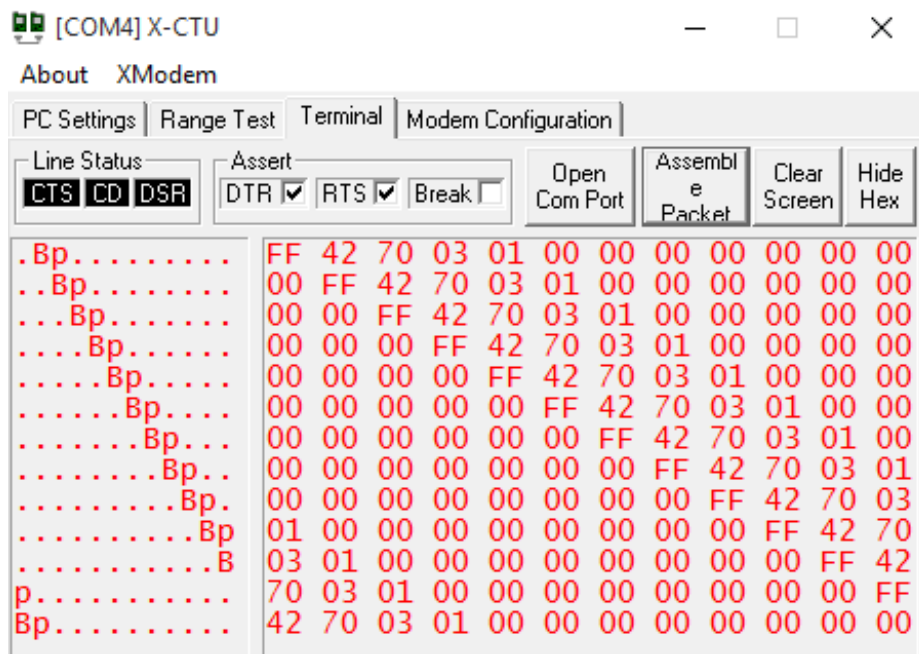


Figura 21 Trama de datos entregada por la báscula para un peso de 1037.0g

Encabezado

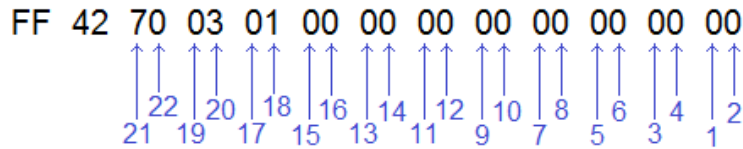


Figura 22 Interpretación de la trama de datos a un valor de peso en gramos

Para el programa desarrollado en *Arduino* con el fin de capturar estos datos, se utilizó el concepto *First In, First Out* (FIFO), teniendo en cuenta que el micro iba a atender varias tareas al tiempo. Además, se recurrió al uso de interrupciones, ya que de esta forma no se perdería información mientras se lee y almacena cada dato de peso entrante, caso contrario a lo que pasaría si se utilizara un *loop*.

La idea básica del programa era leer y guardar cada byte que arrojaba la pesa (empezando con FF), hasta que se tuvieran un total de 13, para luego extraer cada número por medio de corrimientos y máscaras e imprimirlos por el monitor serial siguiendo el orden

anteriormente descrito. Cabe anotar que no todos los datos fueron seleccionados para mostrarlos por pantalla, sólo los números desde el 15 hasta el 22, pues se consideró que los demás nunca serían diferentes de cero, ya que el motor no llegaría a tener un peso tan alto.

También es importante mencionar que para comunicar la pesa con el *Arduino* se usó un conversor RS-232 a TTL, puesto que el micro trabaja con los niveles de voltaje de este último.

El respectivo diagrama de flujo para la lectura del peso del motor se muestra como sigue:

- Función *fifoPushData*: Esta función inserta un dato en la FIFO.

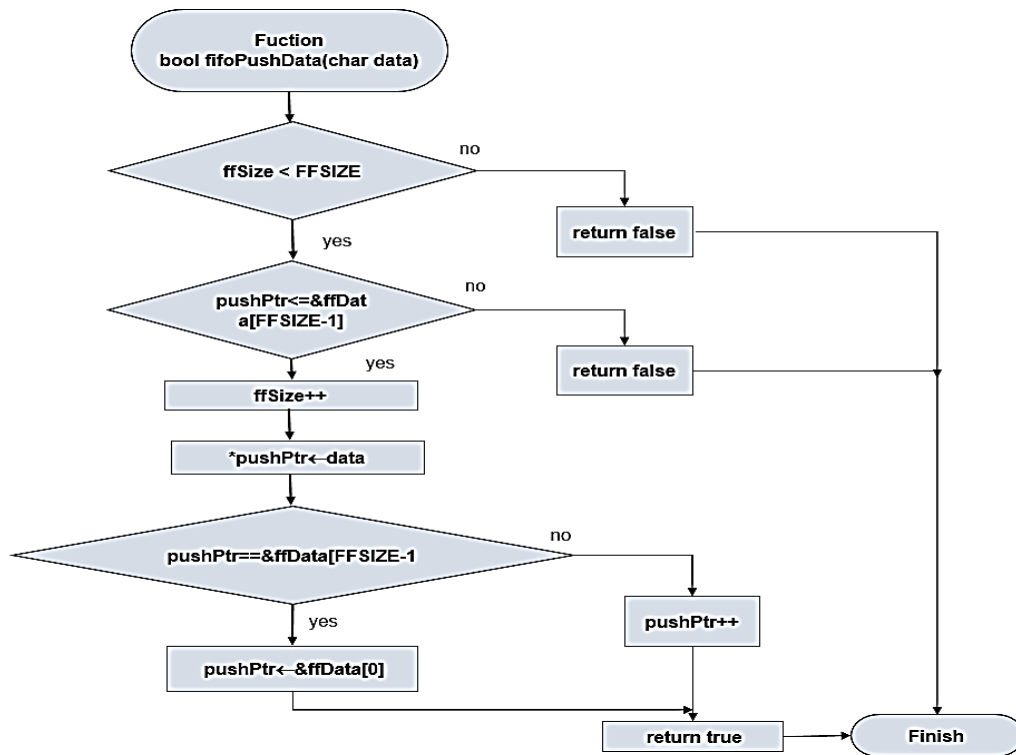


Figura 23 Diagrama de Flujo función *fifoPushData*

- Función *ffifoPullData*: Esta función saca el dato más antiguo de la FIFO.

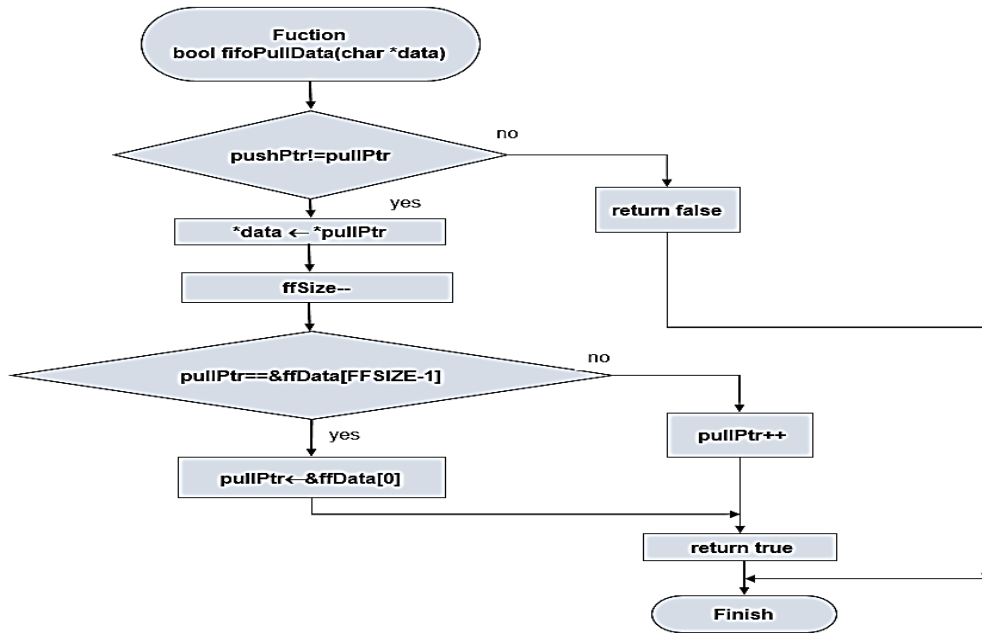


Figura 24 Diagrama de Flujo función fifoPullData

- Programa principal:

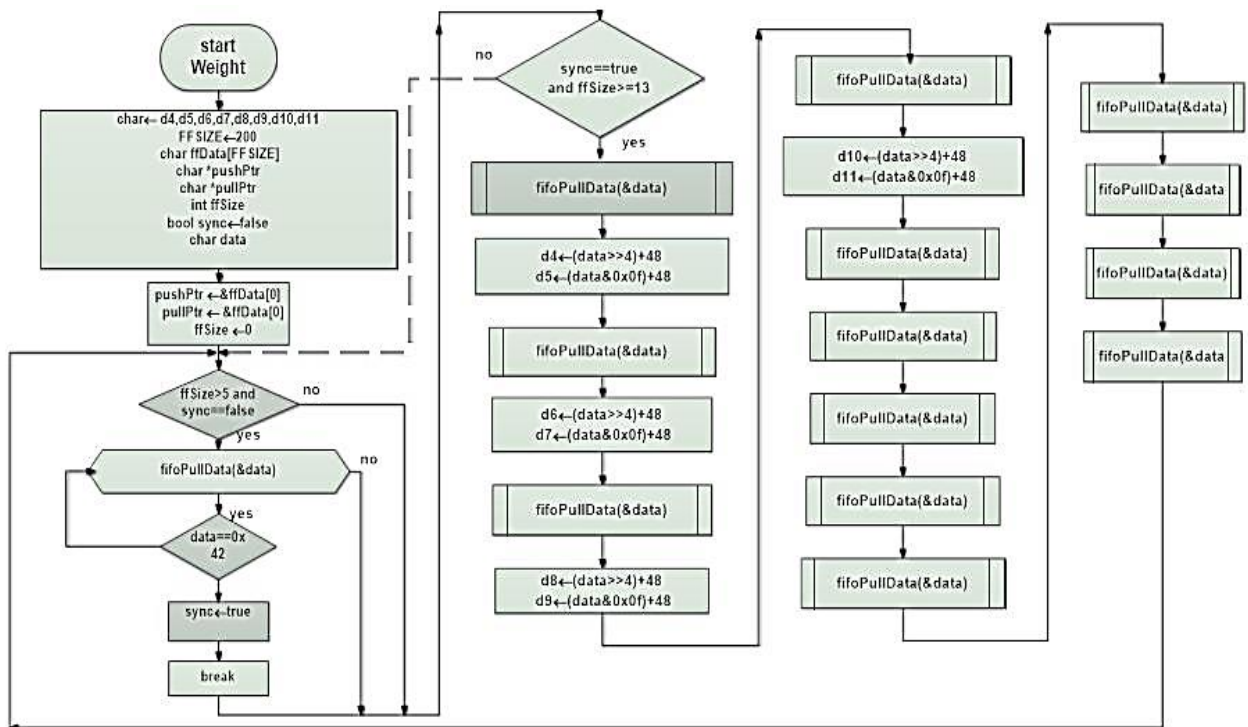


Figura 25 Diagrama de Flujo programa Comunicación con Báscula

- Subrutina SerialEvent2

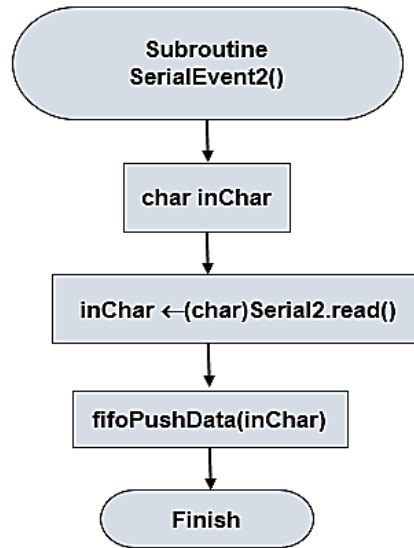


Figura 26 Digrama de Flujo Interrupción de Serial

Lectura de RPM del motor:

Para la toma de las revoluciones por minuto del motor se utilizó como sensor un diodo infrarrojo y un fototransistor, además de un amplificador operacional, buscando así obtener una señal perfectamente digital a la salida. El esquemático empleado se muestra en la siguiente imagen:

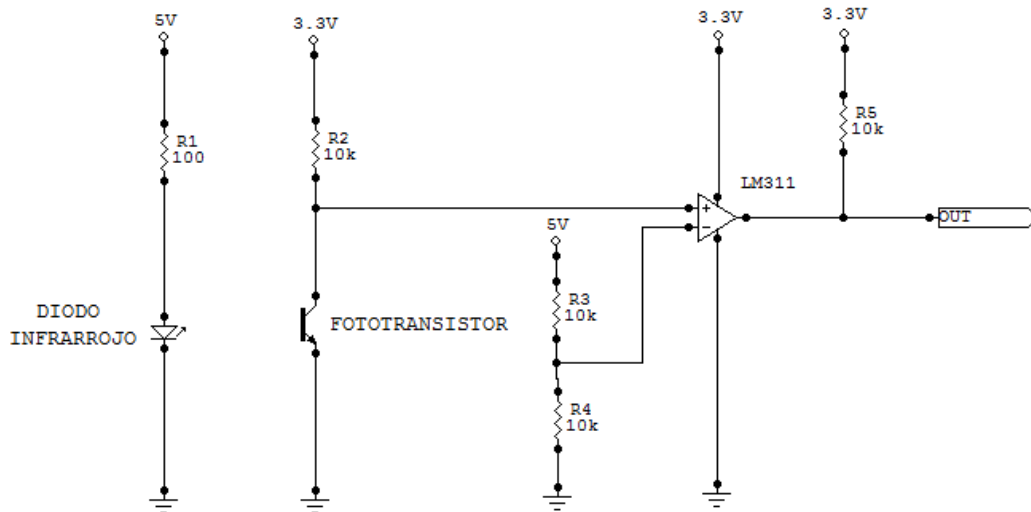


Figura 27 Circuito para sensar la velocidad del motor en RPM

El funcionamiento del circuito se puede analizar en dos momentos, a continuación se explica cada uno de ellos:

- Luz no interrumpida:** En este caso, el fototransistor recibe el haz de luz infrarrojo emitido por el diodo, lo cual hace que este primero trabaje en la región de saturación, poniendo en la entrada no inversora del amplificador operacional un voltaje de aproximadamente 1.8V (su cierre no es completo), y ya que esta tensión es inferior al nivel referencia de la entrada inversora (2.5V), a la salida del operacional aparecerá un potencial bajo (0V).
- Luz interrumpida:** Corresponde al momento en que la hélice se encuentra en el medio del diodo y el fototransistor impidiendo el paso de luz, obligando al transistor a trabajar en la región de corte; de esta manera llegará a la entrada no inversora del operacional un voltaje igual a 3.3V, que al ser restado con la referencia de 2.5V, resultará un nivel positivo de tensión, lo que hace que la salida del amplificador sea alta (3.3V).

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Esto puede resumirse de la siguiente manera: en el instante en que la hélice del motor pase a través del sensor, habrá un pulso alto a la salida.

Para facilitar la medición de esta magnitud, se construyó una estructura en forma de “C”, en donde se colocaron el diodo infrarrojo y el fototransistor separados una distancia de 3.5cm, espacio suficiente para el paso de la hélice. La Figura 28 ilustra la estructura del sensor en forma de “C”.

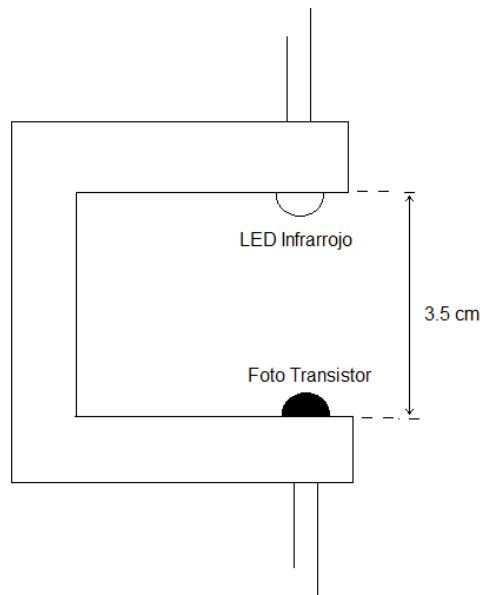


Figura 28 Estructura de sensor en C para medir la velocidad del motor

Al igual que para la lectura del peso, el programa para las RPM también se realizó utilizando interrupciones, las cuales se generarían cada vez que hubiera un flanco ascendente en el pin por el cual se leerían. Para calcular dichas revoluciones se optó por el método del periodo, pues conociendo el tiempo en el que la hélice da una vuelta completa, al calcular la frecuencia se puede obtener el número de vueltas que daría en 1 segundo, y al multiplicarlas por 60, ya se tendrían las RPM.

El diagrama de flujo del programa para la lectura de las revoluciones por minuto del motor se puede observar en la siguiente imagen:

- Programa principal:

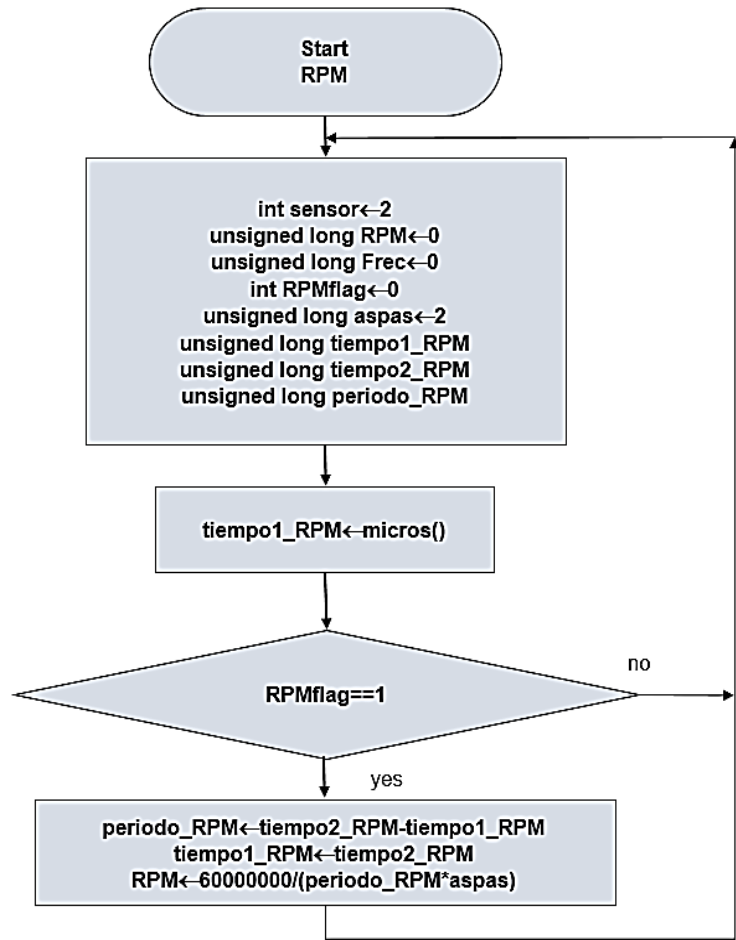


Figura 29 Diagrama de Flujo del Sensado de Velocidad

- Interrupción:

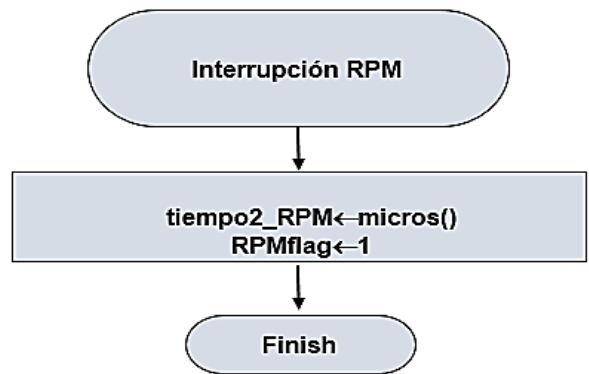


Figura 30 Diagrama de Flujo Interrupción de Input-Capture

De esta manera, el diagrama de flujo correspondiente al programa para la toma completa de los datos es el siguiente:

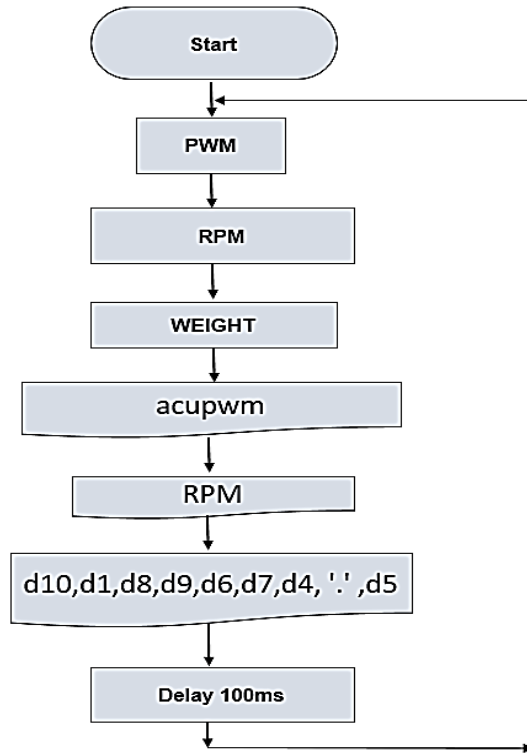


Figura 31 Diagrama de flujo del programa completo

Note que la impresión por pantalla de las tres magnitudes de interés se realizó cada 100ms, obteniendo así 10 paquetes de datos en 1 segundo.

Para los diagramas de flujo se tuvo en cuenta la siguiente simbología:

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

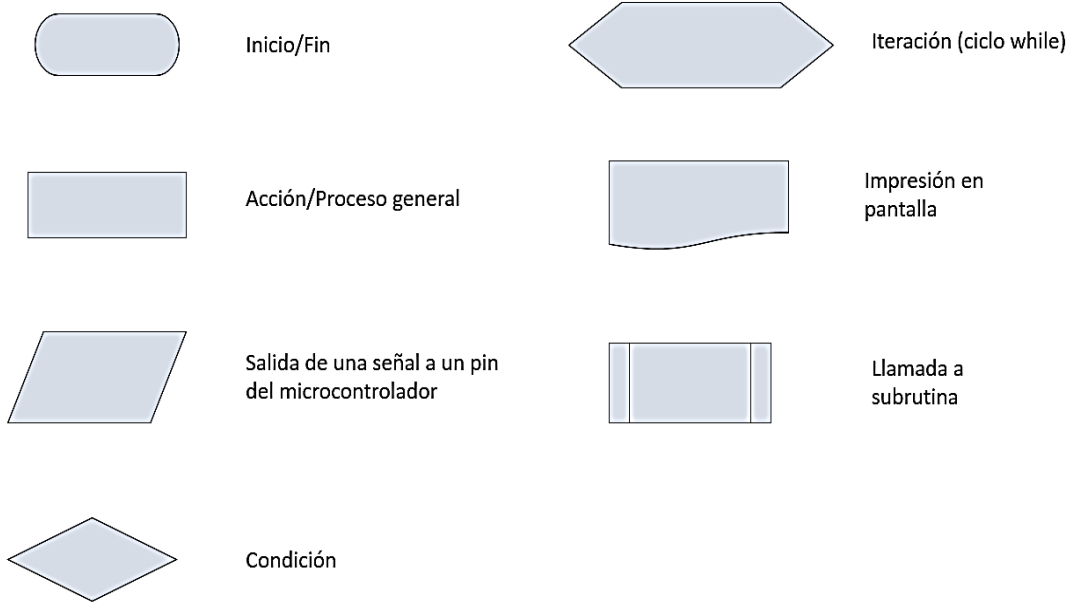


Figura 32 Simbología de los diagramas de flujo

El panorama completo de esta toma de datos puede apreciarse en el siguiente diagrama de bloques.

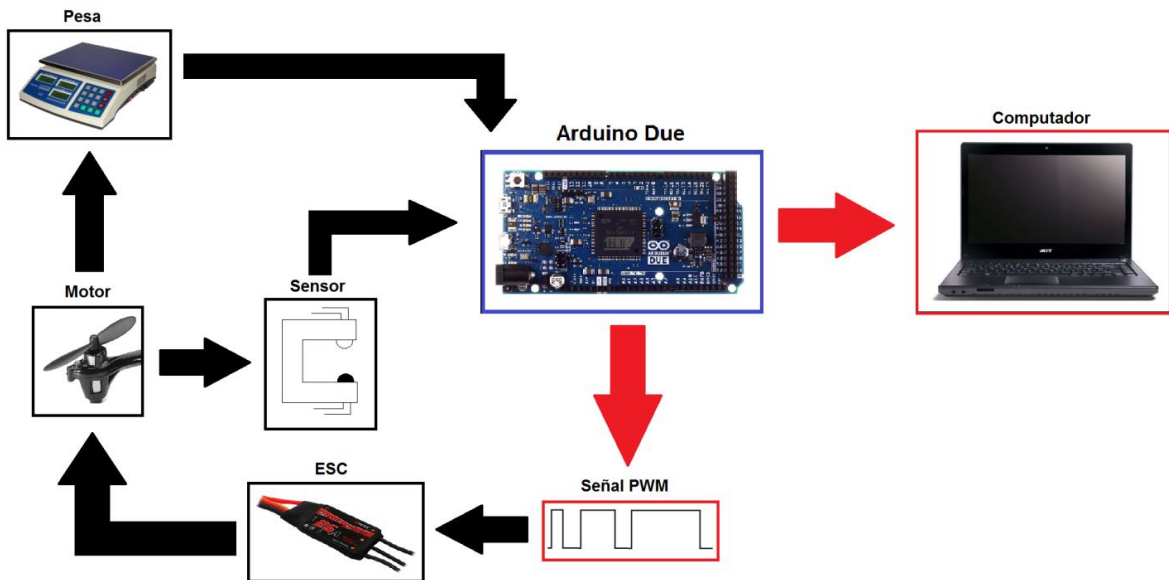


Figura 33 Diagrama de bloques del experimento

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Se observa que el *Arduino* envía la señal de salida PWM al ESC, el cual gobierna el motor, que a su vez es chequeado por la pesa y el sensor de RPM, obteniendo de éstos últimos la información de interés que es transmitida hasta la placa *Arduino-Due*, dicho microcontrolador procesa estos datos y se comunica con el computador para finalmente imprimir por pantalla los valores de PWM, empuje y RPM en cada instante de tiempo.

2.3 Sistema de Visualización en Matlab

Se utilizó un modo de evaluación cualitativa, en donde se hace una comparación entre el vuelo real del cuadracoptero y el vuelo dinámico. Para esto se requirió un modelo visual gráfico en MATLAB que nos permita expresar y visualizar la orientación del cuadracoptero con respecto a un marco de referencia fijo.

A continuación, se muestra una representación matemática por medio de los ángulos *pitch*, *roll* y *yaw*, los cuales representan las tres rotaciones sobre los ejes X, Y y Z respectivamente. Por facilidad a la hora de elaborar el código en MATLAB, la translación en los ejes x, y z es independiente a las rotaciones del modelo en los ángulos *pitch*, *roll* y *yaw*.

2.3.1 Orientación y matrices de rotación

Las rotaciones alrededor de unos ejes coordenados que simbolizarán el sistema de referencia del cuadracoptero, provocarán el cambio en la orientación del mismo. Se definió el siguiente sistema de coordenadas y las rotaciones en sentido positivo como se muestra en la Figura 34 **Error! No se encuentra el origen de la referencia..**

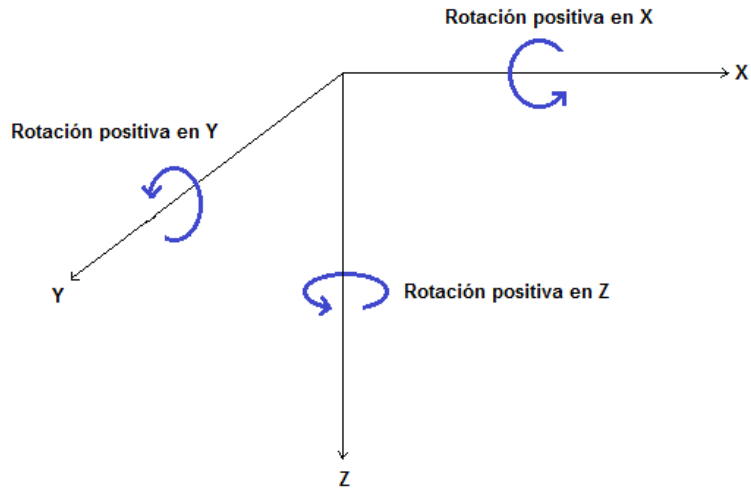


Figura 34 Sistema de coordenadas

Se eligió la configuración en +, de manera que la estructura del cuadraoptero y sus cuatro rotores quedan ubicados sobre el eje de coordenadas como se muestra en la Figura 35 **¡Error! No se encuentra el origen de la referencia.**)

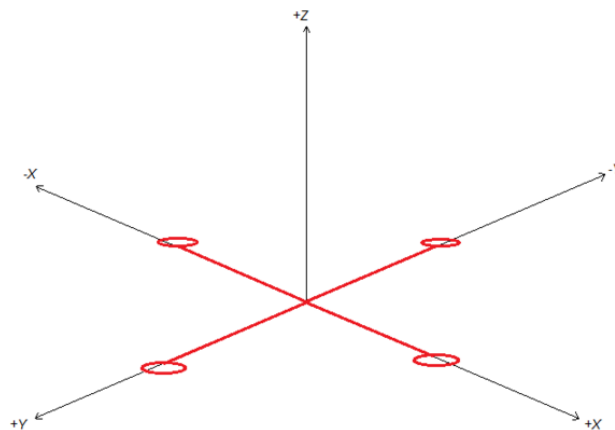


Figura 35 Cuadroptero en Configuración +

Para describir la orientación del cuadroptero respecto al el sistema de coordenadas dado (Fig. 2), se parte de que los vectores \overline{UX} , \overline{UY} , \overline{UZ} , son coincidentes con el eje de coordenadas X, Y, Z.

Si la estructura del cuadroptero rota alrededor del eje X, un ángulo ρ (*Pitch*) entonces se obtiene los vectores $\overline{UX'}$, $\overline{UY'}$, $\overline{UZ'}$ como se muestra en la Figura 36.

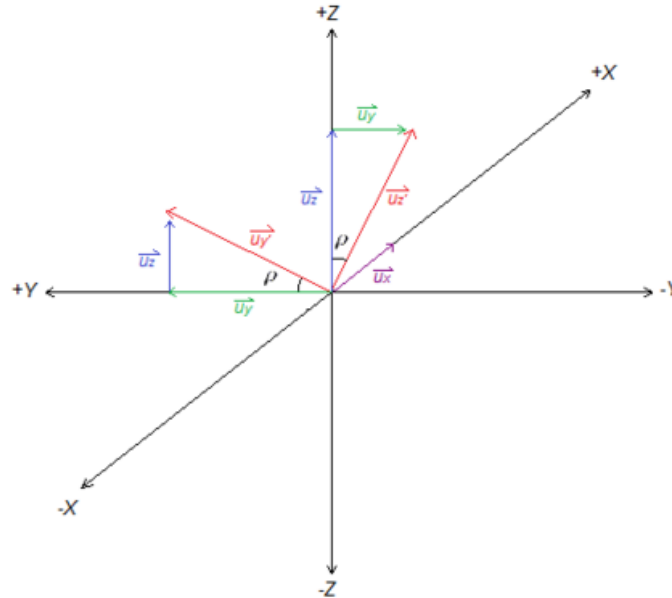


Figura 36 Rotación tipo PITCH

Así las coordenadas del vector resultante son:

$$\overline{UZ'} = \cos \rho \overline{UZ} - \sin \rho \overline{UY} \quad (26)$$

$$\overline{UY'} = \cos \rho \overline{UY} + \sin \rho \overline{UZ} \quad (27)$$

$$\overline{UX'} = \overline{UX} \quad (28)$$

Se tiene la siguiente matriz de rotación:

$$R_X(\rho) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \rho & -\sin \rho \\ 0 & \sin \rho & \cos \rho \end{bmatrix}$$

Si la estructura gira alrededor del eje Y, un ángulo γ (Roll) entonces se obtiene los vectores $\overline{UX''}$, $\overline{UY''}$, $\overline{UZ''}$ como se muestra en la Figura 37. **Error! No se encuentra el origen de la referencia..**

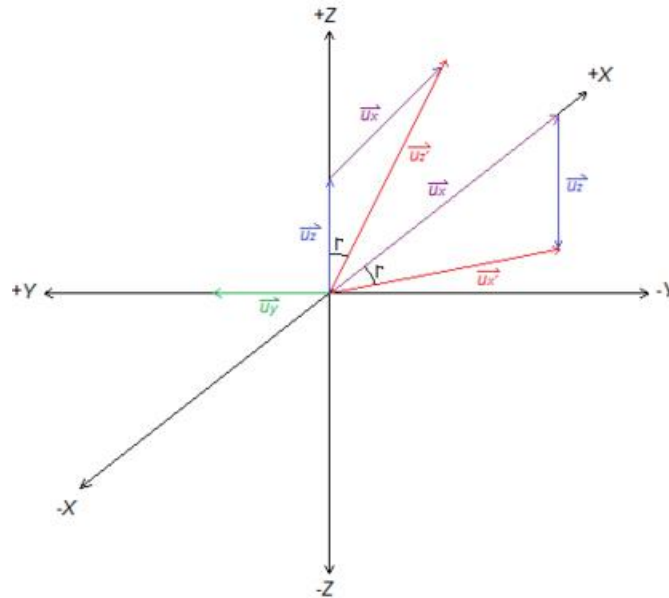


Figura 37 Rotación tipo ROLL

Las coordenadas del vector resultante son:

$$\overline{UX''} = \cos \gamma \overline{UX} - \sin \gamma \overline{UZ} \quad (29)$$

$$\overline{UZ''} = \cos \gamma \overline{UZ} + \sin \gamma \overline{UX} \quad (30)$$

$$\overline{UY''} = \overline{UY} \quad (31)$$

Se tiene la siguiente matriz de rotación:

$$R_y(\gamma) \begin{bmatrix} \cos \gamma & 0 & \sin \gamma \\ 0 & 1 & 0 \\ -\sin \gamma & 0 & \cos \gamma \end{bmatrix}$$

Si la estructura gira alrededor del eje Y, un ángulo θ (Yaw) entonces se obtiene los vectores $\overline{UX''''}$, $\overline{UY''''}$, $\overline{UZ''''}$ como se muestra en la Figura 38 **Error! No se encuentra el origen de la referencia..**

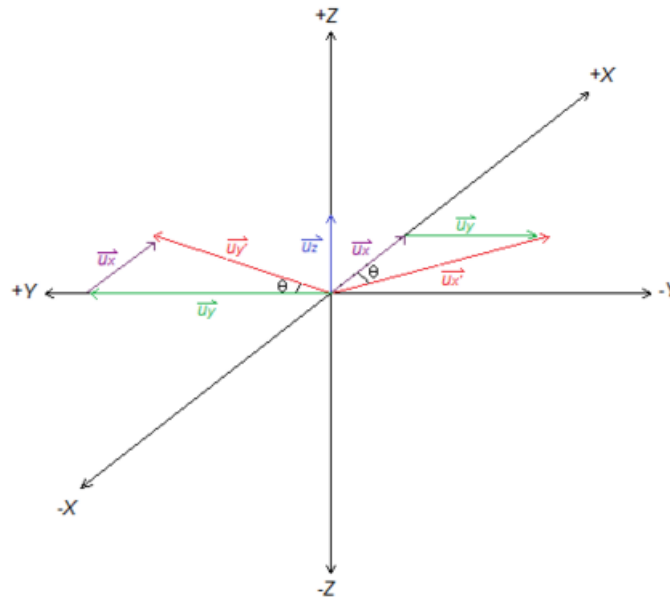


Figura 38 Rotación tipo YAW

Las coordenadas del vector resultante son:

$$\overline{UX}''' = \cos \theta \overline{UX} - \sin \theta \overline{UY} \quad (32)$$

$$\overline{UY}''' = \cos \theta \overline{UY} + \sin \theta \overline{UX} \quad (33)$$

$$\overline{UZ}' = \overline{UZ} \quad (34)$$

Se tiene la siguiente matriz de rotación:

$$R_z(\theta) \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Luego de definir las matrices de rotación se creó la representación gráfica del *drone* en Matlab, por medio de la matriz [3,8] *DroneData* en donde cada columna corresponde a uno de los puntos de cada parte de la estructura. Las coordenadas de cada modelo de la estructura se representan por pares coordenados, por ejemplo, un par de puntos son las coordenadas de uno del boom, otro par de puntos son las coordenadas de las flechas y así sucesivamente (Ver código MyGraficx.m).

3. RESULTADOS Y DISCUSIÓN

Para encontrar el modelo dinámico del subsistema motor-hélice-ESC utilizamos la interfaz gráfica del *toolbox* de identificación de sistemas, denominada “*ident*”. Esta herramienta ofrece diversos métodos de identificación de sistemas a partir de datos experimentales tomados de la planta de interés. A continuación, se describe cada atributo del entorno de trabajo de la interfaz gráfica del *toolbox* de identificación de sistemas “*ident*”.

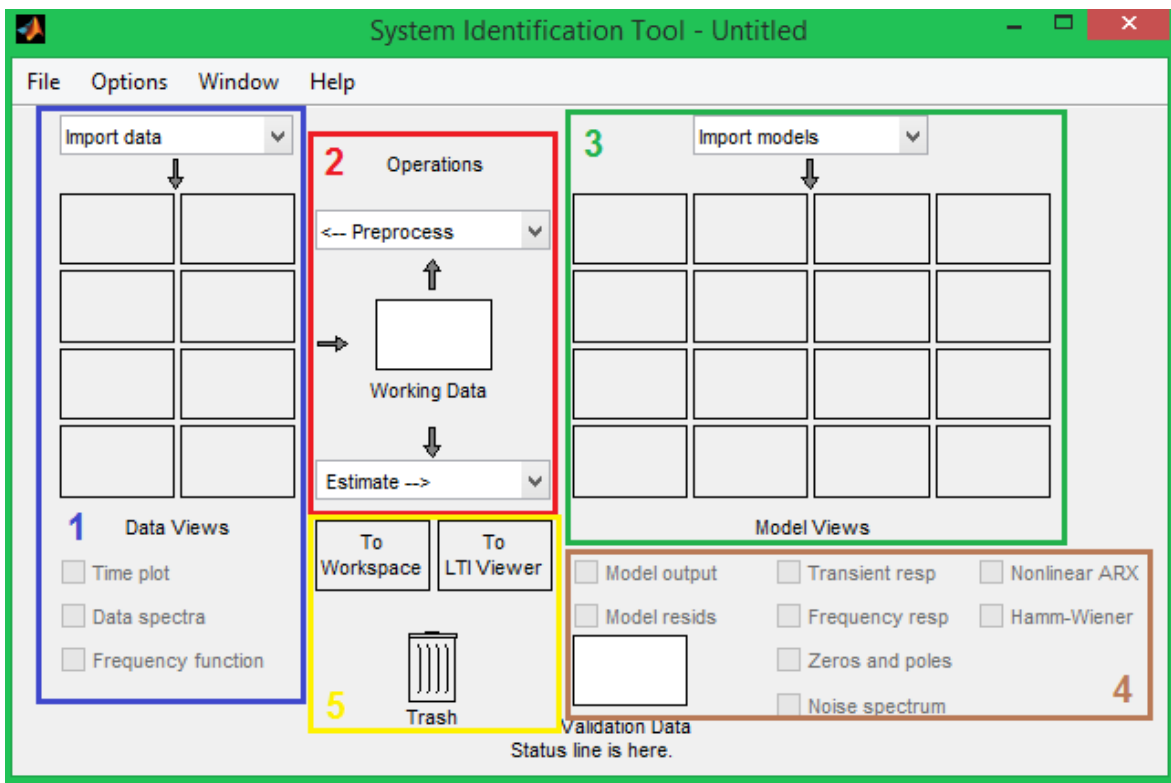


Figura 39 Ventana principal de la herramienta *Ident*

La Figura 39 presenta la ventana principal de *Ident*. Se ha encerrado en cuadros de colores numerados diferentes sectores de la interfaz gráfica y se explican a continuación:

Sector 1. Este sector es utilizado para importar los datos correspondientes a las entradas y salidas del sistema (previamente deben haber sido cargados en variables de Matlab). Cada recuadro representa un conjunto de datos de entrada del sistema dinámico que se desea identificar. También es posible visualizar las gráficas de entrada y salida en el dominio del

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

tiempo y de la frecuencia. Para esto utilice los *checkbox* “**time plot**”, “**data spectra**” y “**frequency function**”.

Sector 2: Este sector permite seleccionar el conjunto de datos que se quiere usar en la identificación (En caso). Aquí se seleccionan los datos de trabajo (en caso de haber varias importaciones), con el comando “*Preprocess*” pueden crearse nuevos conjuntos de datos a partir de los de trabajo (por ejemplo, para eliminar tendencias, seleccionar rango, entre otros); por último, en “*Estimate*” se elige el método con el cual se hallará el modelo matemático de los datos importados.

Sector 3: Simboliza la salida (modelo aproximado) predicha de los sistemas seleccionados, y al dar clic derecho sobre cada uno, se despliega una ventana con los parámetros de la ecuación que lo constituye.

Sector 4: Corresponde a los diferentes puntos de vista desde donde se puede analizar el modelo calculado, ya que es posible graficar dicho modelo, su respuesta en frecuencia, el lugar geométrico de las raíces, entre otras.

Sector 5: Esta zona permite exportar la expresión hallada al espacio de trabajo de Matlab, a la interfaz gráfica *LTI Viewer* o eliminar dicho elemento, simplemente arrastrando su símbolo al recuadro correspondiente.

A continuación se muestra la importación de los datos tomados del motor para su identificación, teniendo en cuenta que la entrada fue una señal PWM y como salida el Empuje.

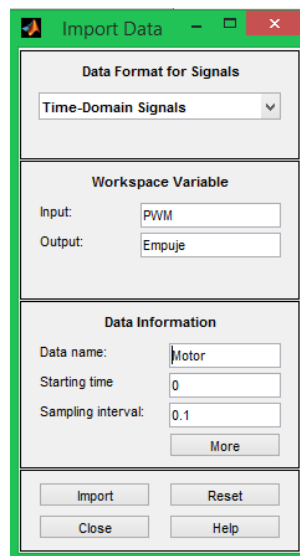


Figura 40 Ventana de importación de datos de entrada y salida

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Cabe anotar que el tiempo de muestreo se definió como 0.1 segundos, ya que en el experimento se adquiriría 1 dato cada 100ms.

Para este caso, se encontró la ecuación del motor de tres formas distintas, donde se observó una gran diferencia entre ellas en cuanto a aproximación y exactitud respecto a la función real; en cada método se presenta el primer intento y el mejor modelo hallado.

Método 1: Modelos Lineales Paramétricos

Dentro de esta técnica aparecen diferentes estructuras de salida para el sistema estimado: ARX, ARXMAX, OE y BJ, variando en cada una de ellas la complejidad de la función. En la siguiente imagen se muestra la ventana de configuración para el cálculo del modelo, posteriormente se detalla cada sector.



Figura 41 Ventana de trabajo para el método de Identificación con modelos lineales paramétricos

Sector 1: En este sector se elige el tipo de estructura que se utilizará para la estimación del sistema y se introducen los valores de los parámetros exigidos por el formato.

Sector 2: Este sector es utilizado para establecer el estado inicial, la covarianza y el enfoque del método: predicción, simulación, estabilidad, entre otros.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Sector 3: En este sector se configuran atributos más específicos dependiendo de la estructura seleccionada, como lo es la robustez del sistema, la tolerancia, el límite de iteraciones, etc.

Primer Intento: Inicialmente se utilizó la estructura de modelo ARX, ya que representaba de una forma polinómica simple la ecuación del sistema dinámico; para ello fue necesario ingresar los valores de tres variables: na (posible número de polos), nb (posible número de ceros+1) y nk (retardo de entrada o tiempo muerto del modelo); las cuales se pusieron a cambiar en números de 1 a 10 con el fin de probar múltiples combinaciones entre dichos parámetros, como se aprecia en la imagen.

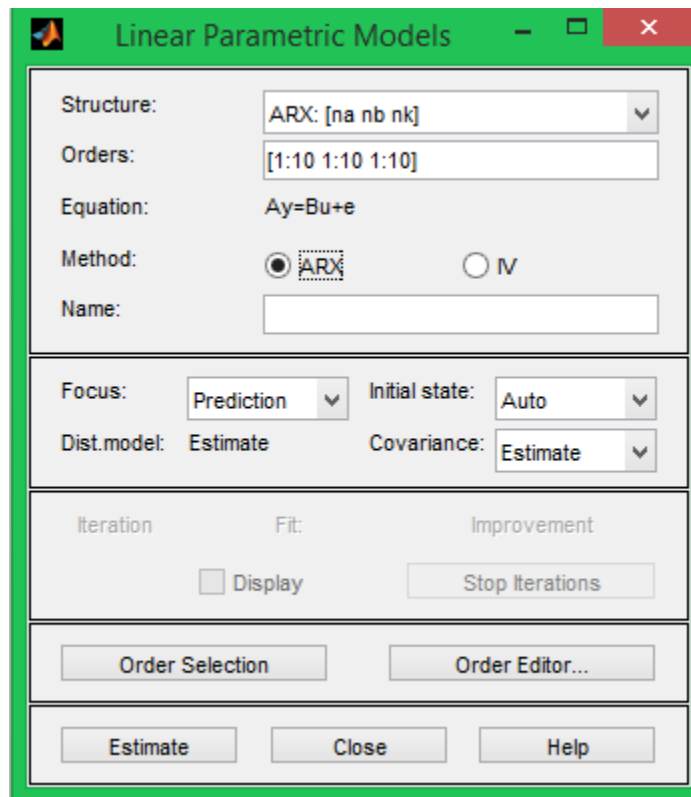


Figura 42 Ventana de configuración de parámetros para un modelo ARX

Al estimar la función con estos datos, se obtuvo la gráfica de la Figura 43, donde el eje horizontal corresponde al número de parámetros $na + nb$ (en total 20), y el eje vertical corresponde al porcentaje de error de cada modelo. Cada barra ilustra los modelos hallados (donde el más grande, es el modelo menos aproximado); por otra parte, la barra de color rojo significa que dicho modelo es el más cercano al sistema dinámico real, y fue encontrado con $na = 10$, $nb = 10$ y $nk = 7$.

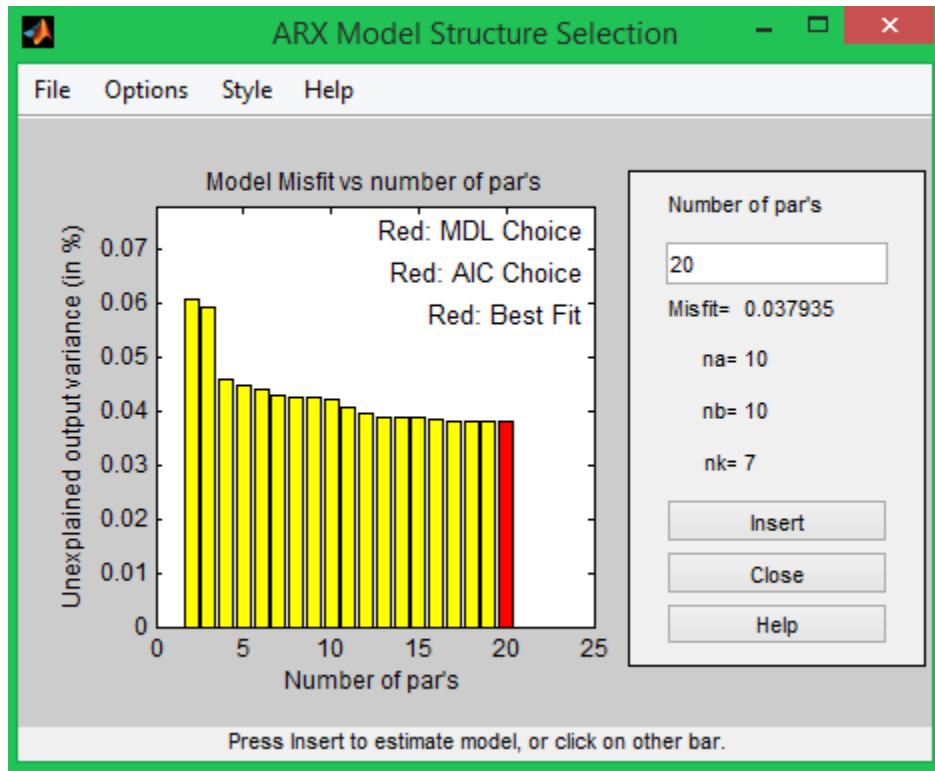


Figura 43 Porcentaje de incongruencia entre modelos con estructura ARX y los datos experimentales

La grafica de la mejor identificación del sistema se muestra en la siguiente imagen, donde se observa que su porcentaje de exactitud fue de 75.68%.

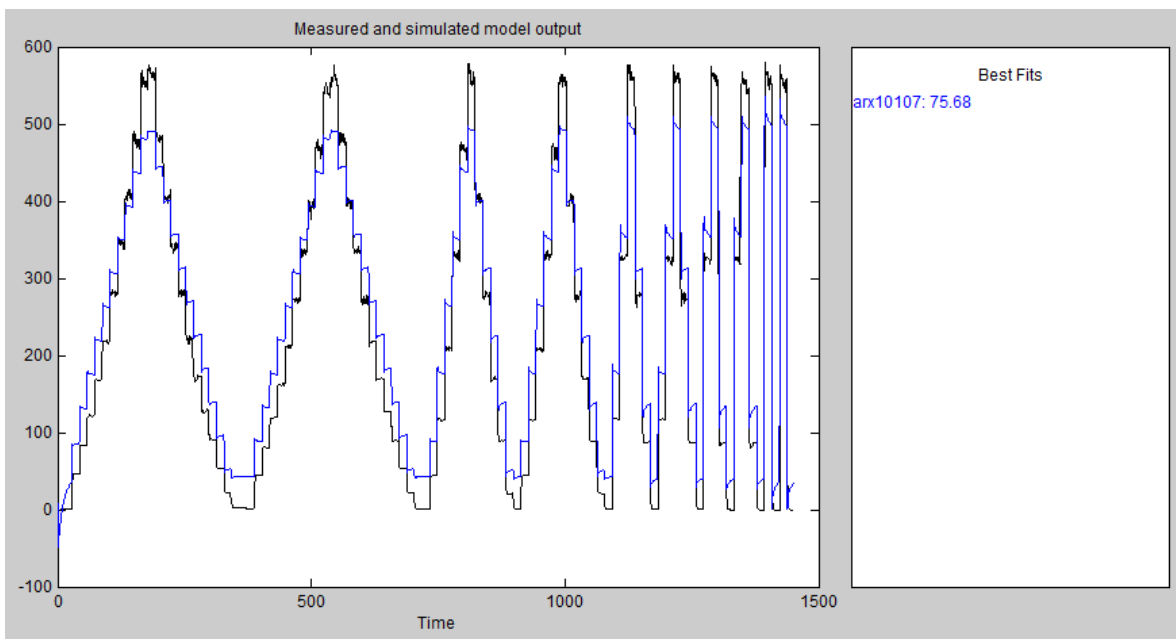


Figura 44 Comparación entre datos experimentales (negro) y modelo estimado ARX (azul)

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

La ecuación generada para este modelo fue la siguiente:

$$\text{Discrete-time IDPOLY model: } A(q)y(t) = B(q)u(t) + e(t)$$

$$\begin{aligned} \text{Empuje}(t) = & (PWM(t) * (0.5049 q^{-7} + 0.2934 q^{-8} - 0.09456 q^{-9} - 0.2465 q^{-10} - \\ & 0.01578 q^{-11} + 0.03128 q^{-12} + 0.0548 q^{-13} + 0.421 q^{-14} - 0.4186 q^{-15} - \\ & 0.4862 q^{-16}) + e(t)) / (1 - 0.889 q^{-1} - 0.465 q^{-2} + 0.351 q^{-3} + 0.03211 q^{-4} + \\ & 0.04991 q^{-5} + 0.03076 q^{-6} - 0.06328 q^{-7} - 0.07948 q^{-8} + 0.006371 q^{-9} + \\ & 0.03467 q^{-10}) \end{aligned}$$

Mejor modelo hallado: Al hacer uso de la estructura OE (Output Error) la cual incluye en su estimación posibles ruidos, se encontró el modelo más aproximado de este método, alcanzando una semejanza del 84.43%. La parametrización de esta estructura es idéntica a la del modelo ARX con la diferencia de que la variable na es llamada nf . Dichas variables se fueron cambiando de valor, llegando a ser los mejores $nb = 5, nf = 5$ y $nk = 1$.

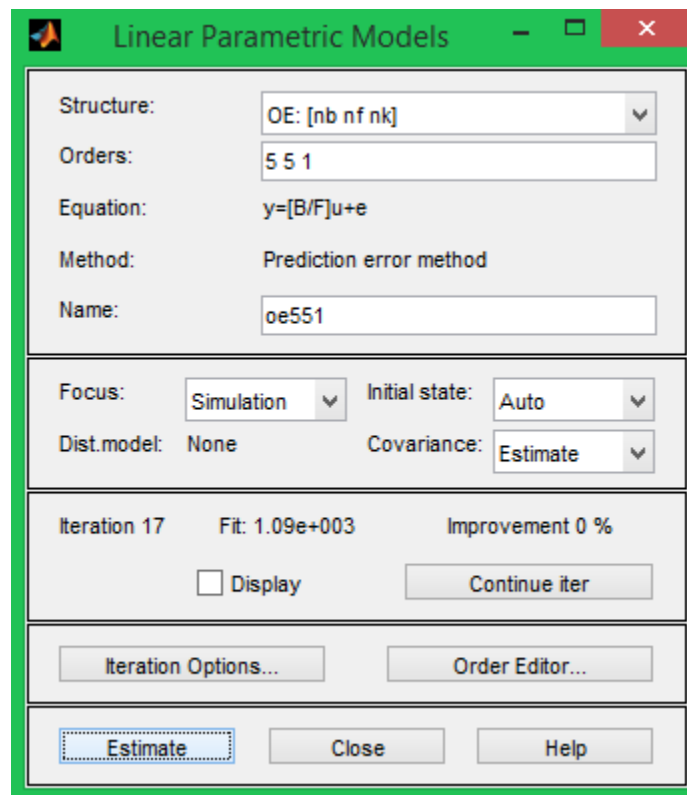


Figura 45 Ventana de configuración de parámetros para modelos tipo OE

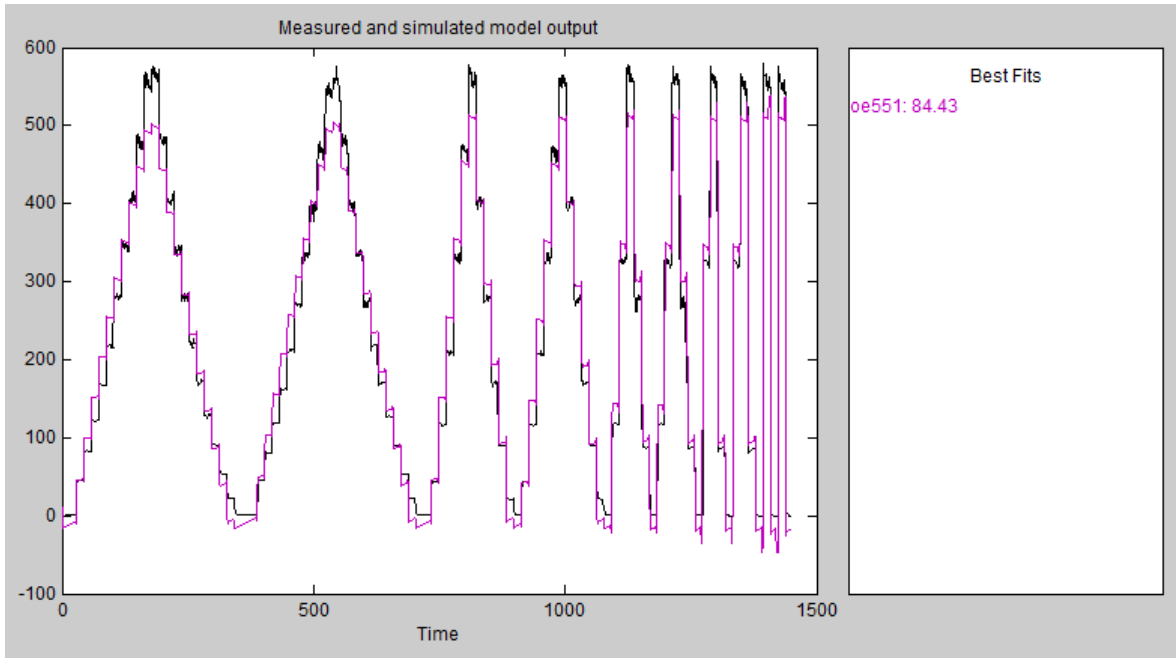


Figura 46 Comparación entre datos experimentales (negro) y modelo estimado OE (púrpura)

La ecuación correspondiente a esta curva se muestra a continuación:

$$\text{Discrete-time IDPOLY model: } y(t) = [B(q)/F(q)]u(t) + e(t)$$

$$\text{Empuje}(t) = \text{PWM}(t) * [(-0.3482 q^{-1} + 1.291 q^{-2} - 1.928 q^{-3} + 1.471 q^{-4} - 0.4853 q^{-5}) / (1 - 3.694 q^{-1} + 5.416 q^{-2} - 3.928 q^{-3} + 1.401 q^{-4} - 0.1942 q^{-5})] + e(t)$$

Método 2: Identificación de Modelo de Procesos

En este método puede seleccionar si el modelo contiene 1 cero, hasta 3 polos, retardo o integradores, cambiando el formato de la función de transferencia generada de acuerdo a ello; además, las variables de dicha función pueden ser editadas por el usuario agregando un número fijo o un intervalo en el cual se cree que puede estar su valor, o simplemente dejar que en la identificación se estime automáticamente. A continuación se describe cada sección de la interfaz de dicho método.

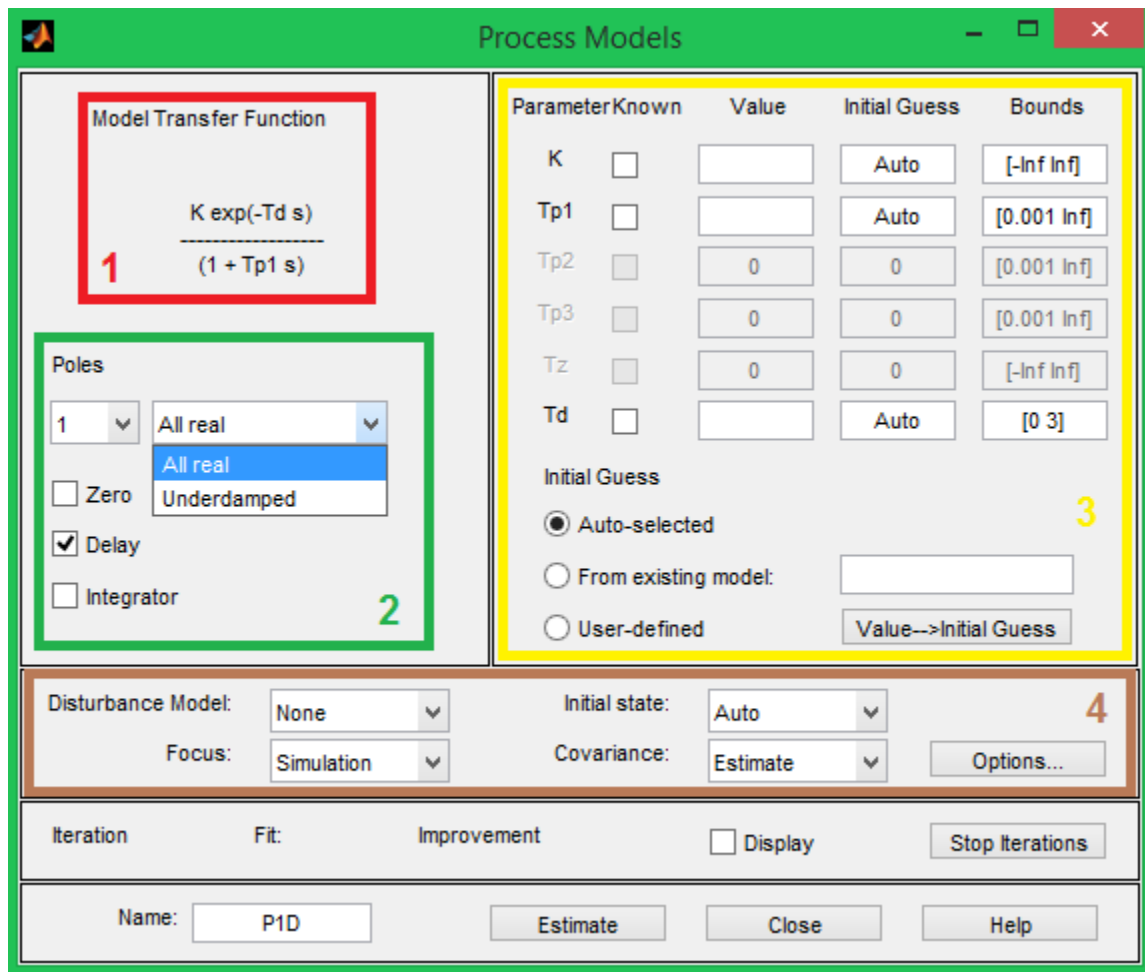


Figura 47 Ventana de trabajo para el método de Identificación con modelos de procesos

Sector 1: En este sector se aprecia la ecuación genérica de la función de transferencia del sistema.

Sector 2: Este sector es usado para elegir el número de polos, especificar si el sistema es subamortiguado e incluir 1 cero, retardo o 1 integrador a través de los *checkbox*.

Sector 3: En este sector se puede dar un valor fijo a cada una de las variables, restringirlas a un intervalo definido, inicializarlas con un número deseado o dejar que Matlab las estime. Además es posible adjuntar un modelo existente, para que el programa empiece a calcular el nuevo modelo a partir de él.

Sector 4: En este último sector, se configura el estado inicial, la covarianza y las posibles perturbaciones que podrían tenerse en cuenta para la estimación del modelo.

Primer intento: Se incluyó todo lo posible: 3 polos, 1 cero el retardo y el integrador, y se dejó en Auto las variables de la función como se ve en la Figura 48.

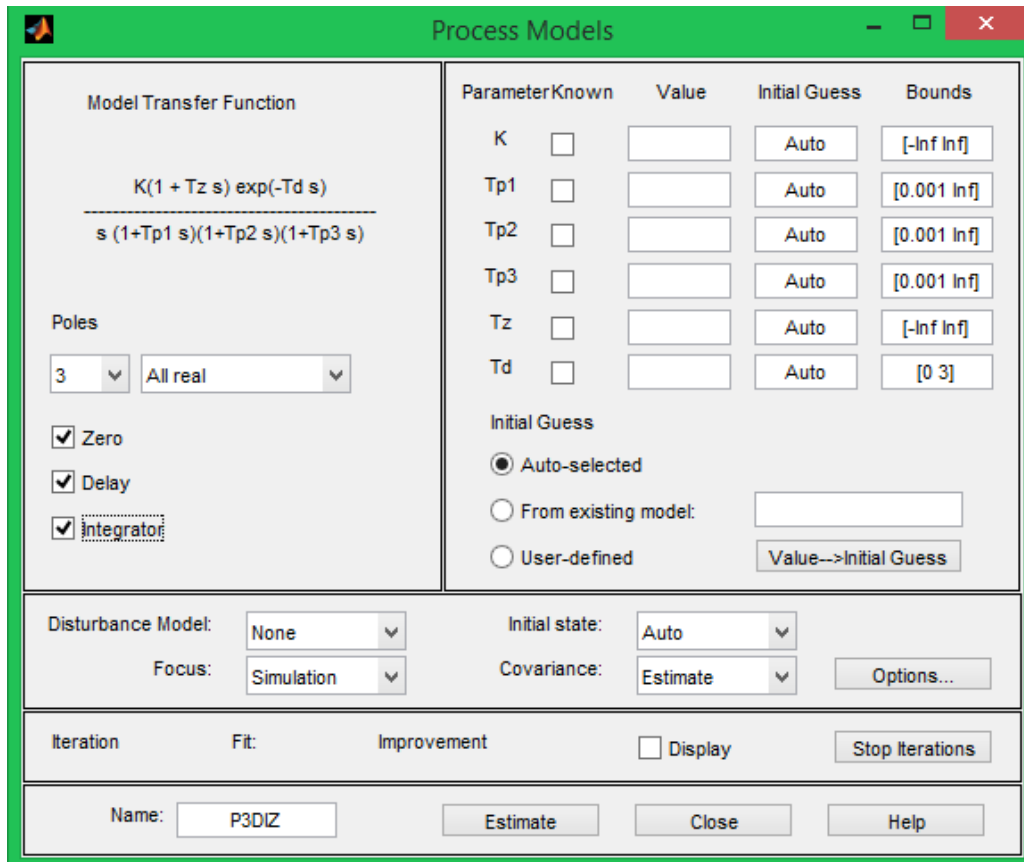


Figura 48 Ventana de configuración de parámetros para Modelos de Procesos intento 1

Dando origen a un sistema aproximado un 76.23%.

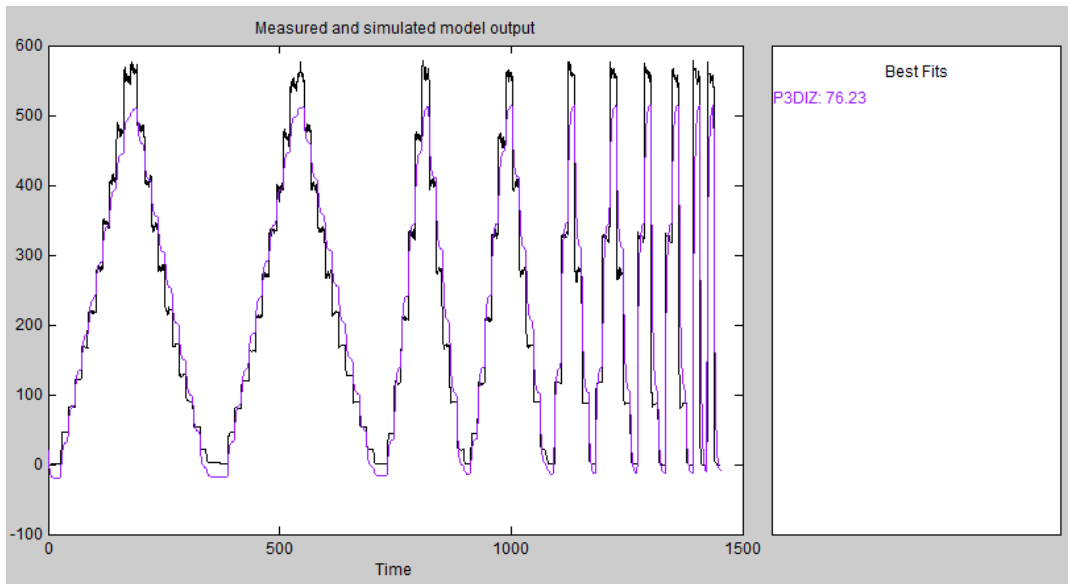


Figura 49 Comparación entre datos experimentales (negro) y el modelo de proceso (morado) Intento 1

La función de transferencia resultante fue la siguiente:

$$G(s) = Kp * \frac{1 + Tz * s}{s(1 + Tp1 * s)(1 + Tp2 * s)(1 + Tp3 * s)} * \exp(-Td * s)$$

$$G(s) = 0.00011052 * \frac{1 + 58252S}{S(1 + 0.001S)(1 + 2.7497S)(1 + 0.001S)} * e^{-0.17649S}$$

Mejor modelo hallado: En este caso, se eligieron 2 polos, se colocó 1 cero, se incluyó un retardo, y se dejó de lado el integrador como se ve en la imagen.

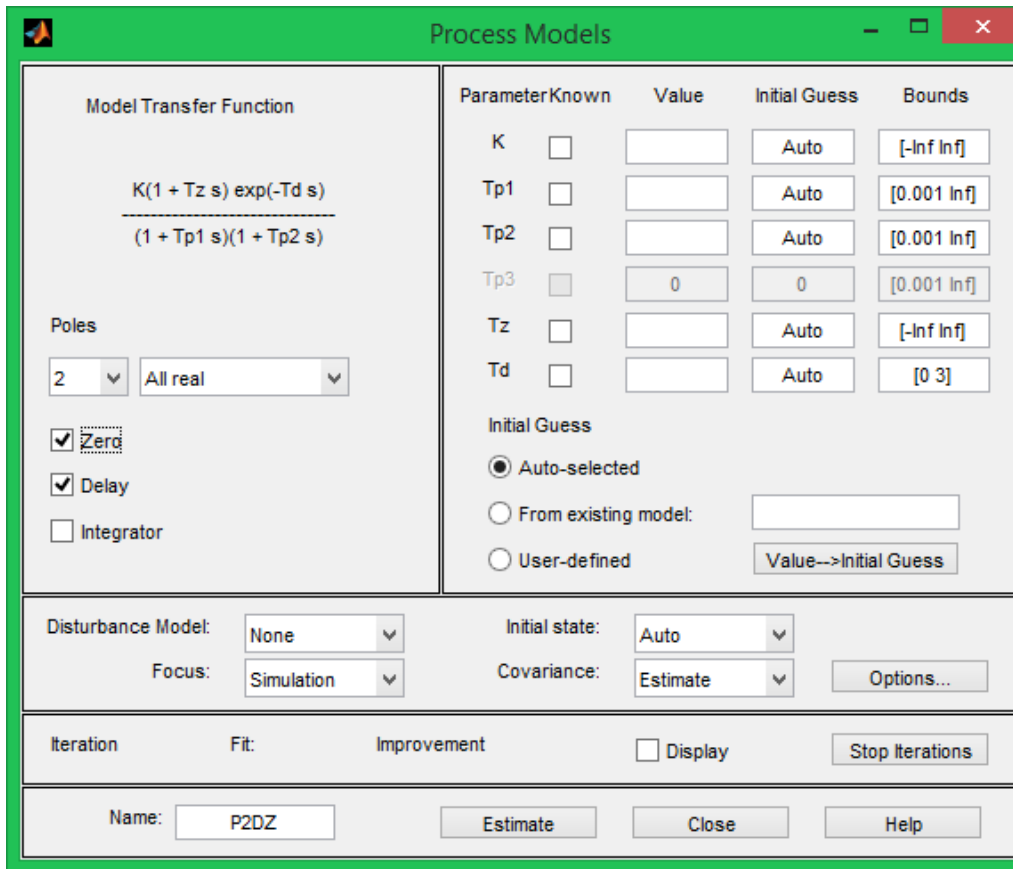


Figura 50 Ventana de configuración de parámetros para Modelos de Procesos intento 2

Esto dio como resultado un modelo con una aproximación del 85.39%.

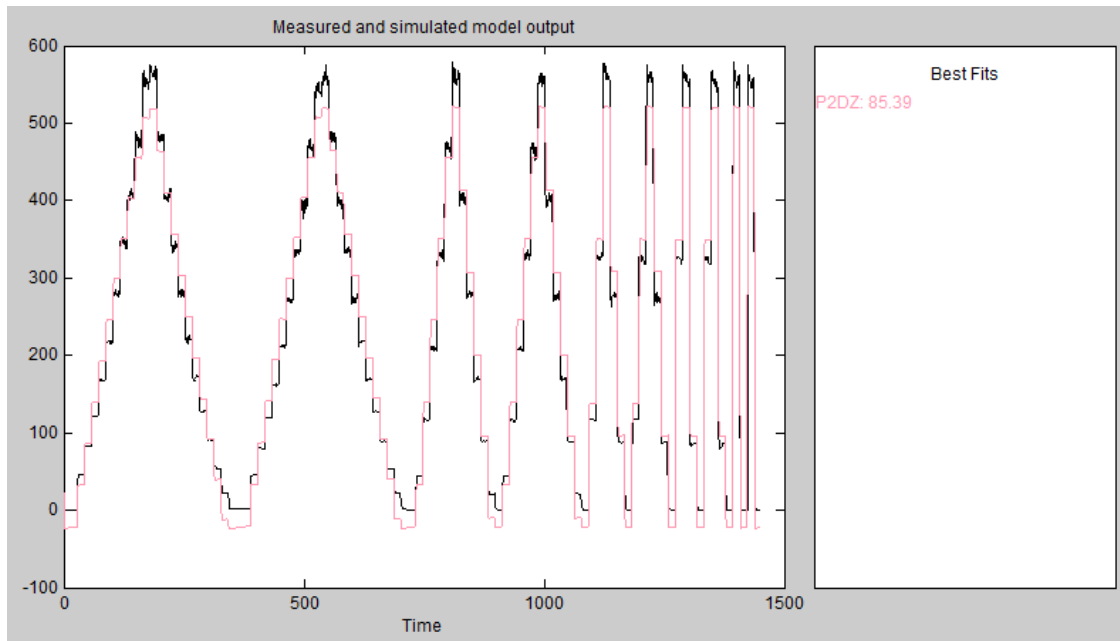


Figura 51 Comparación entre datos experimentales (negro) y el modelo de proceso (rosado) Intento 2

La función de transferencia de este sistema se ilustra en la próxima ecuación:

$$G(s) = Kp * \frac{1 + Tz * s}{(1 + Tp1 * s)(1 + Tp2 * s)} * \exp(-Td * s)$$

$$G(s) = 4.9849 * \frac{1 + 848.81S}{(1 + 638.47S)(1 + 0.38905S)} * e^{-0.73581S}$$

Método 3: Modelos no lineales

Este método da la opción de calcular las funciones de dos formas diferentes: ARX no lineal y Hammerstein-Wiener; este primero utiliza regresiones para hallar el modelo utilizando bloques de manera lineal y paralela para describirlo, mientras que el segundo es conformado por un elemento lineal estático intercalado entre dos sistemas dinámicos. La ventana de configuración de este método se muestra y explica como sigue.

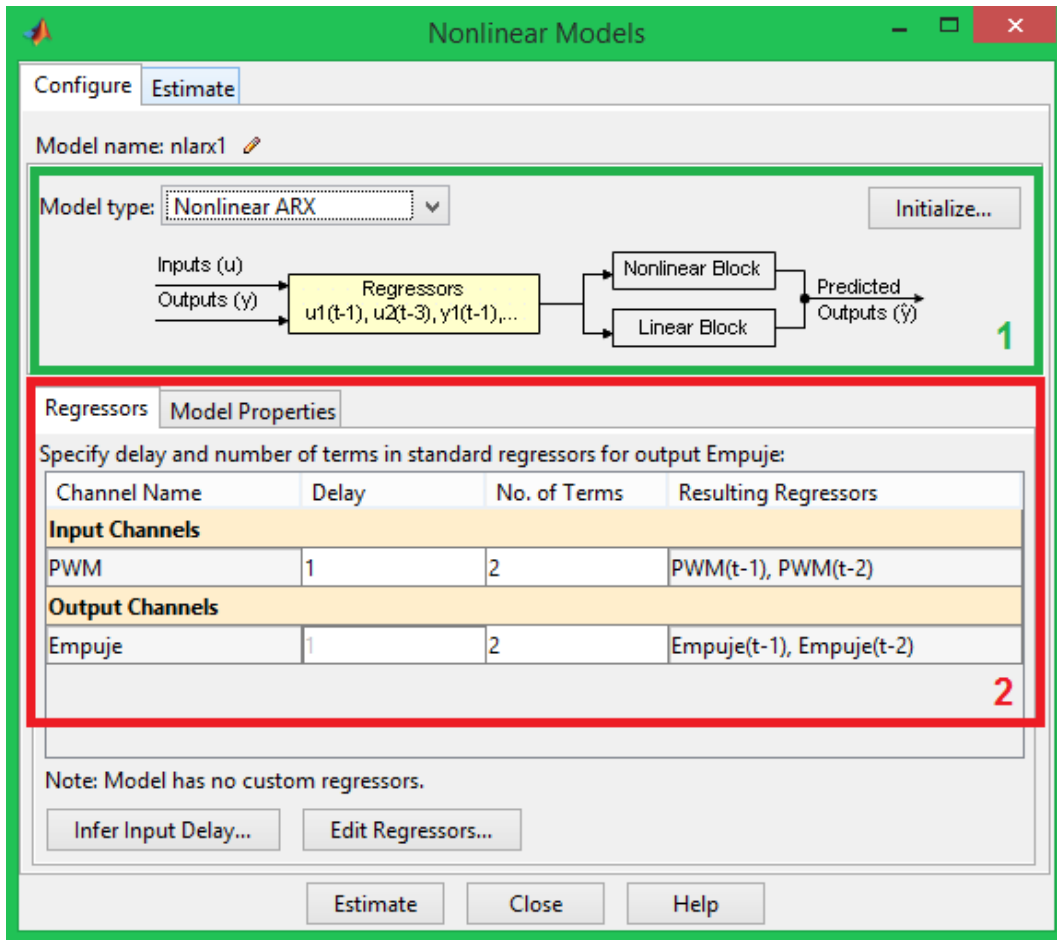


Figura 52 Ventana de trabajo para el método de Identificación con modelos no-lineales

Sector 1: En este sector se escoge el tipo de modelo no lineal a utilizar en la estimación. Además se ilustra el diagrama de bloques del método elegido.

Sector 2: Este sector es usado para configurar y parametrizar cada uno de los bloques anteriormente mencionados.

Primer intento: Al utilizar la forma ARX no lineal, y especificar el retardo y el número de términos que podría tener la expresión se obtuvo un modelo aproximado un 89.18% como se observa a continuación.

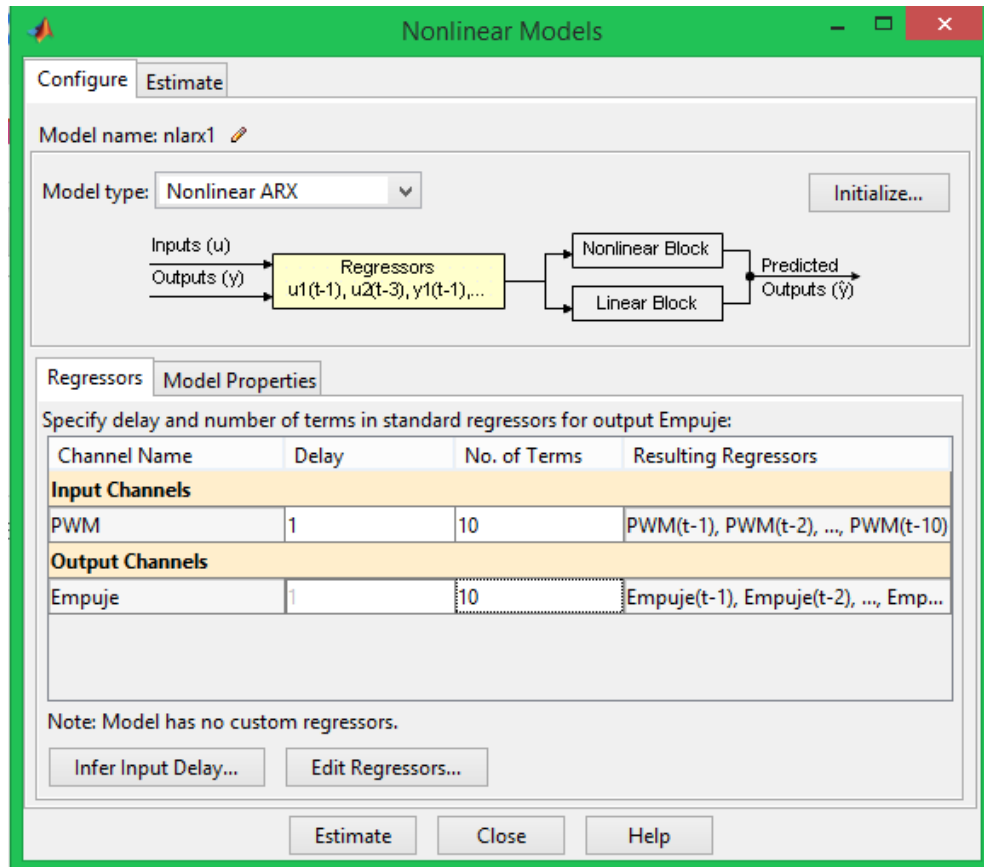


Figura 53 Ventana de configuración de parámetros para modelos ARX no-lineales

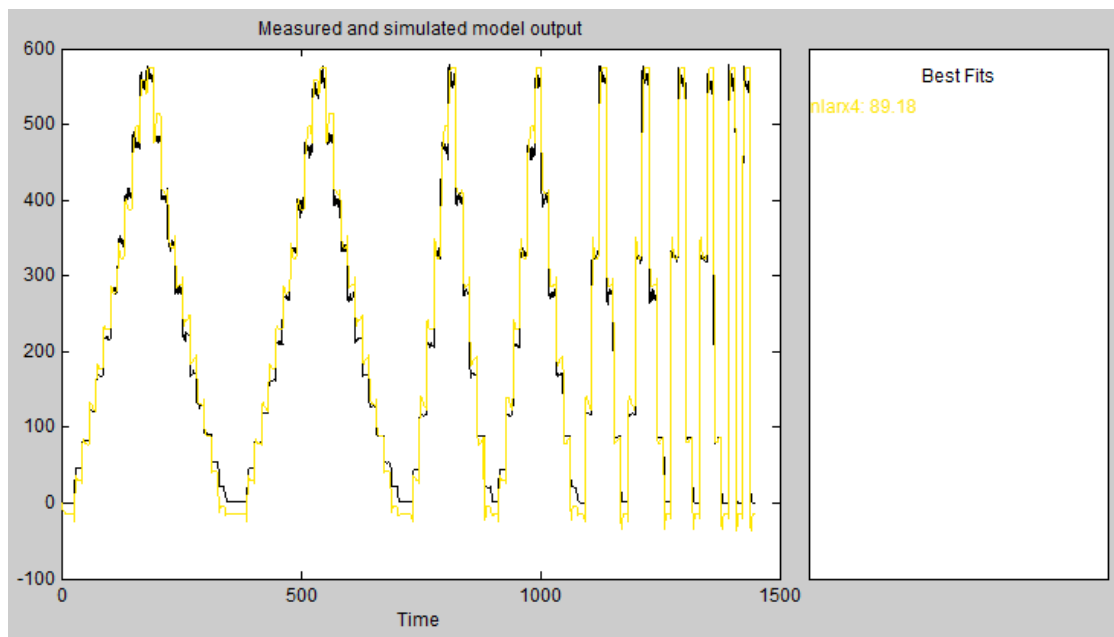


Figura 54 Comparación entre datos experimentales (negro) y el modelo ARX no-lineal (Amarillo)

Mejor modelo hallado: Para esta identificación se utilizó el modelo Hammerstein-Wiener, por ello se debieron configurar dos componentes: el bloque no lineal y el lineal; referente a este primero, el tipo de no linealidad seleccionado fue el de “lineal por tramos”, y se eligió 12 términos para la expresión; por otro lado, en la parte lineal se configuraron 2 ceros, 3 polos y un retardo igual a 1. Las siguientes figuras ilustran esta parametrización.

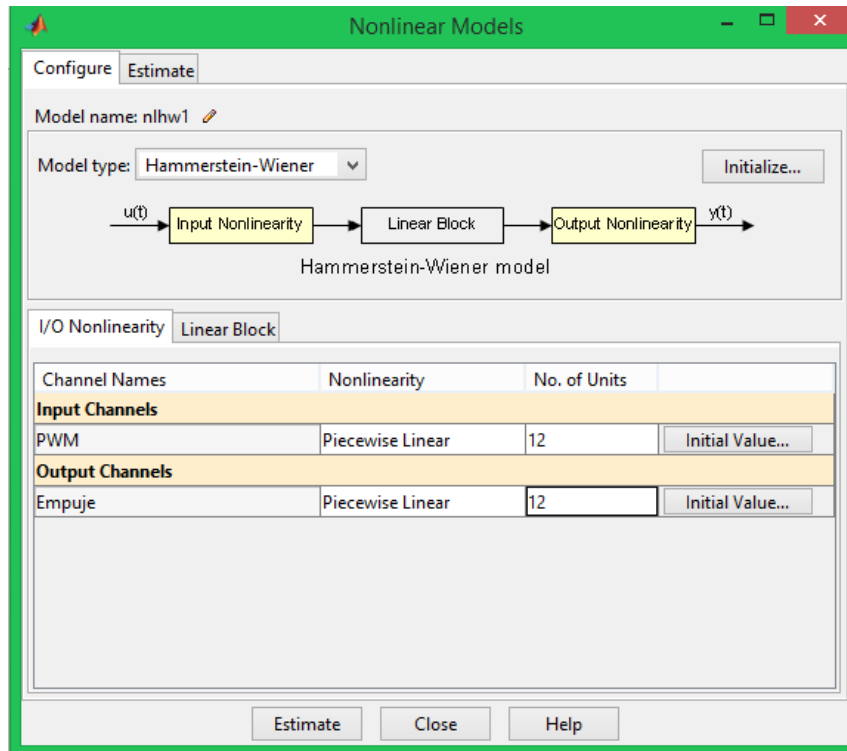


Figura 55 Ventana de configuración de parámetros para bloques no-lineales del modelo Hammerstein-Wiener

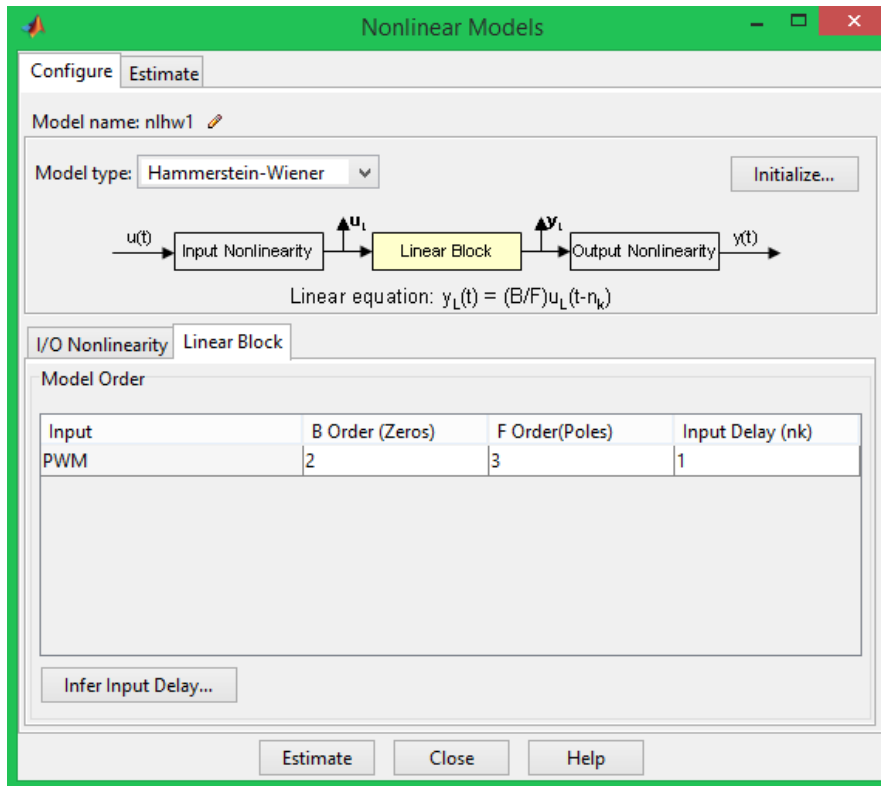


Figura 56 Ventana de configuración de parámetros para el bloque lineal del modelo Hammerstein-Wiener

La estimación con dicha parametrización dio como resultado un modelo con un 93.99% de aproximación al sistema dinámico real, como lo muestra la gráfica.

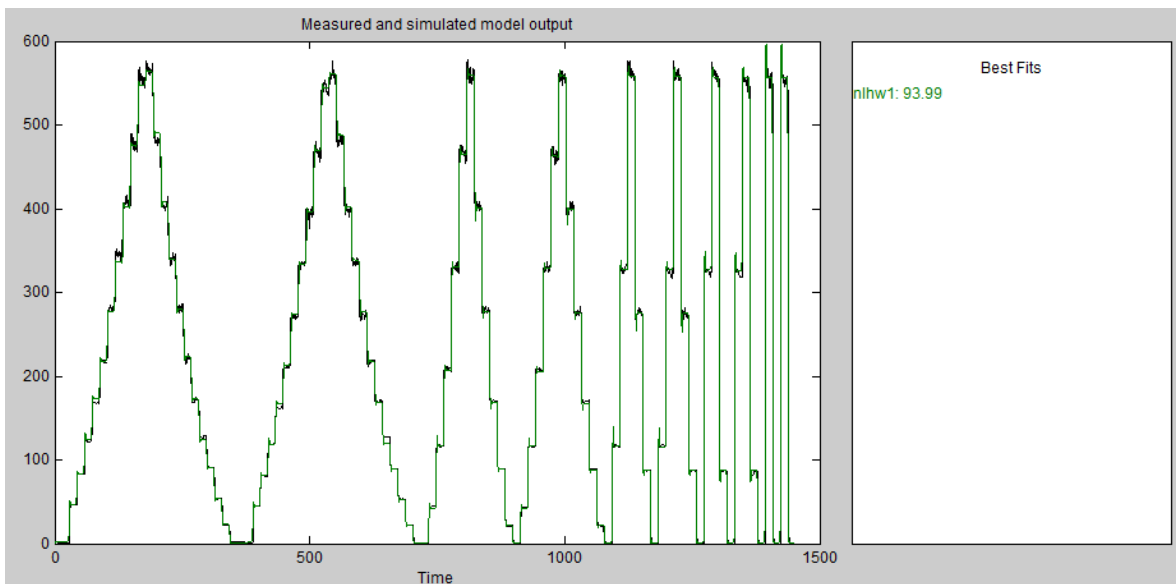


Figura 57 Comparación entre datos experimentales (negro) y el modelo Hammerstein-Wiener (verde)

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

- Durante el desarrollo de este trabajo tuvimos la oportunidad de estudiar y comprender con cierto grado de profundidad la complejidad del modelo dinámico de un cuadricoptero. Es claro que todo modelo dinámico tiene sus limitaciones y el usado en este trabajo no es la excepción. Una de las suposiciones esenciales del modelo es la simetría del cuadricoptero, la cual, en sistemas reales es casi imposible de alcanzar. Puede ser interesante tratar de cuantificar el impacto de dichas asimetrías en el desempeño general del modelo.
- El subsistema formado por el motor, la hélice y el ESC tiene un gran impacto en el modelo dinámico del cuadricoptero. En este trabajo hemos encontrado un modelo no lineal con un 94% de coincidencia con los datos experimentales y de esa forma hemos incrementado la precisión del modelo dinámico de vuelo. Sin embargo nuestro trabajo solo abarca la componente de empuje del motor e ignora el torque debido a la falta de instrumentos para medirlo. En trabajos futuros sería necesario incluir la caracterización del torque para mejorar aún más el modelo del subsistema motor-hélice-ESC.
- Creemos que la inclusión del ESC en la caracterización del motor es fundamental ya que estos dispositivos son de naturaleza compleja, e incluso tiene modos de trabajo configurables que es necesario estudiar y medir su impacto en la dinámica del subsistema motor-hélice-ESC.

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

REFERENCIAS

- Burkamshaw, L. K. (2010). Towards a low-cost quadrotor research platform.
- Gremillion, G., & Humbert, J. S. (2010). System identification of a quadrotor micro air vehicle. In *AIAA Conference on Atmospheric Flight Mechanics*.
- Hartman, K. Landis, M. Mehrer, S. Moreno, J. K. (2014). Quadcopter Dynamic Modeling and Simulation (Quad-Sim) v1.00.
- Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, 34(1), 1–12.
- Ma, X., Li, Z., & Nie, C. (2016). Multi-domain modeling and Simulation of Quad-rotor aircraft based on Modelica. In *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods (MSV)* (p. 120).
- Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, 19(3), 20–32.
- Peng, H. (2007). Multivariable RBF-ARX model-based predictive control for nonlinear systems. *IFAC Proceedings Volumes*, 40(12), 882–887.
- Pounds, P., Mahony, R., & Corke, P. (2006). Modelling and control of a quad-rotor robot. In *Proceedings Australasian Conference on Robotics and Automation 2006*.
- Sonntag, D. (2011). A Study of Quadrotor Modelling.
- Sturm, J. (n.d.). Autonomous navigation for flying robots. Lecture 3.2: Sensors.
- Wei, W., Tischler, M. B., Schwartz, N., & Cohen, K. (2014). Frequency-domain system identification and simulation of a quadrotor controller. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference* (pp. 1834–1839).
- Zawiski, R., & Błachuta, M. (2012). Dynamics and optimal control of quadrotor platform. In *AIAA Guidance, Navigation and Control Conf. and Exhibit* (Vol. 1).
- Zhang, X., Li, X., Wang, K., & Lu, Y. (2014). A Survey of Modelling and Identification of Quadrotor Robot. *Abstract and Applied Analysis*, 2014, 1–16. <https://doi.org/10.1155/2014/320526>
- Jaramillo, F., & Gómez, A., (2013). Sistema de control para la estabilidad y orientación de un helicóptero quadrotor, tesis de ingeniería, Escuela de ingeniería de Antioquia, Antioquia, Colombia.
- Phang, S., Cai, C., Chen, B.M., & Lee, T. (2012). Design and mathematical modeling of a 4-standard-propeller (4SP) quadrotor, *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 3270-3275, Beijing, China.
- Bourke, J. (1999). Understading electric power systems (Part 1-5), Control Tower.

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

APÉNDICE

El siguiente código corresponde al programa desarrollado en el microcontrolador Arduino para la medición de magnitudes RPM y empuje del motor con respecto a variaciones de PWM.

```
#include <Servo.h>

#define high_max 2000 // Tiempo máximo en alto (2000us).
#define high_min 700 // Tiempo mínimo en alto(700us).
#define pinpwm 9 // pin de salida pwm.

Servo motor;
int stepvector[5] = {8, 16, 32, 64, 90};
int repvector[5] = {2, 2, 2, 2, 2 };
int steptime=15000;
unsigned long tiempo1_PWM;
unsigned long tiempo2_PWM;
int acupwm=0;
int acupwm2=0;
int i_step=0;
int i_rep=0;
int auxflagdown=0;
int cont_rep=0;
int auxflagreturn=0;
int sensor=3;
unsigned long RPM=0;
unsigned long Frec=0;
int RPMflag=0;
unsigned long aspasm=2;
unsigned long tiempo1_RPM;
unsigned long tiempo2_RPM;
unsigned long periodo_RPM;

char d4;
char d5;
char d6;
char d7;
char d8;
char d9;
char d10;
char d11;

#define FFSIZE 200 // tamaño maximo de la fifo
char ffData[FFSIZE]; // vector de datos de la FIFO
char *pushPtr; // Apuntador a la siguiente posición libre en la fifo
char *pullPtr; // Apuntador al dato mas antiguo en la fifo
int ffSize; // Tamaño actual de la fifo
bool sync=false; // permite sincronizar la primera trama que llega de la peza con el caracter 0x42

unsigned long tiempo1_PRINT;
unsigned long tiempo2_PRINT;
int printtime=100; // Cada 100ms se imprime un dato.
int limitepwm=90; // Se establece un limite de velocidad equivalente al 50% de la velocidad máxima.
```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

void setup() {
  Serial.begin(9600);
  Serial2.begin(9600);
  // En las tres líneas siguientes inicializan la fifo.
  pushPtr = &ffData[0];
  pullPtr = &ffData[0];
  ffSize = 0;

  pinMode(sensor, INPUT);
  attachInterrupt(3, InterrupcionRPM, RISING);

  Serial.println("Iniciando programa...");
  Serial.println("Este programa calibrará el ESC.");

  motor.attach(pinpwm);
  // Escribe sobre el pin 9, el tiempo maximo (La velocidad más alta)
  Serial.println("Escribiendo el ancho de pulso máximo=2000.");
  Serial.println("Enciende la fuente, y espera 2 segundos, entonces presiona cualquier tecla+ENTER.");
  motor.writeMicroseconds(high_max);

  // Espera una entrada por teclado.
  while (!Serial.available());
  Serial.read();

  // Escribe sobre el pin 9, el tiempo mínimo (La velocidad más baja)
  Serial.println("Escribiendo el ancho de pulso mínimo");
  motor.writeMicroseconds(high_min);

  Serial.println("ESC calibrado.");

  tiempo1_RPM=micros();
  tiempo1_PWM=millis();
  tiempo1_PRINT=millis();
}

// Esta función inserta un dato en la fifo.
bool fifoPushData(char data){
  if (ffSize < FFSIZE){
    if(pushPtr<=&ffData[FFSIZE-1]){
      ffSize++;
      *pushPtr=data;
      if((pushPtr==&ffData[FFSIZE-1]))
        pushPtr=&ffData[0];
      else
        pushPtr++;
      return true;
    }
  }
  return false;
}

// Esta función saca el data más antiguo de la fifo.
bool fifoPullData(char *data){
  if (pushPtr!=pullPtr){
    *data = *pullPtr;
    ffSize--;
    if(pullPtr==&ffData[FFSIZE-1]){
      pullPtr=&ffData[0];
    }
    else
    {
      pullPtr++;
    }
    return true;
  }
  return false;
}

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

void loop() {
  tiempo2_PWM=millis();
  tiempo2_PRINT=millis();

  if((steptime<=tiempo2_PWM-tiempo1_PWM)&& auxflagdown==0){
    tiempo1_PWM=tiempo2_PWM;
    acupwm=acupwm+stepvector[i_step];
    if (acupwm>=limitepwm){
      acupwm=limitepwm;
    }
    acupwm2 = map(acupwm, 0, 180, 700, 2000);
    motor.writeMicroseconds(acupwm2);
    //Serial.println(acupwm);
  }

  if(acupwm>=limitepwm){
    auxflagdown=1;
  }

  if((steptime<=tiempo2_PWM-tiempo1_PWM)&&auxflagdown==1){

    tiempo1_PWM=tiempo2_PWM;
    acupwm=acupwm-stepvector[i_step];
    auxflagreturn=1;
    if(acupwm<=0){
      acupwm=0;
    }
    acupwm2 = map(acupwm, 0, 180, 700, 2000);
    motor.writeMicroseconds(acupwm2);
    //Serial.println(acupwm);
  }

  if(acupwm==0 && auxflagreturn==1){
    auxflagreturn=0;
    cont_rep++;
    if(cont_rep < repvector[i_rep]){
      auxflagdown=0;
    }
    else{
      auxflagdown=0;
      i_step++;
      i_rep++;
      cont_rep=0;
    }
  }

  if (i_step==5){
    i_step=0;
    i_rep=0;
  }

  if (RPMflag==1){
    RPMflag=0;
    periodo_RPM=tiempo2_RPM-tiempo1_RPM;
    //Serial.println(periodo_RPM);
    tiempo1_RPM=tiempo2_RPM;
    RPM=60000000/(periodo_RPM*aspas);

    //Serial.print("RPM: ");
    //Serial.println(RPM);
  }
}

```


 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

char data;

// Sincroniza arduino con las tramas que envía la pesa(solo se hace una vez)
if((ffSize>5)and (sync==false)){
  while(fifoPullData(&data)){
    if(data==0x42){
      sync=true;
      break;
    }
  }
}

// A continuación se Lee los datos en la fifo después de la sincronización y cuando la fifo tenga los 13 caracteres de la trama.
if((sync==true)and(ffSize>=13)){

  fifoPullData(&data);
  d4= (data>>4)+48;
  d5=(data&0x0f)+48;

  fifoPullData(&data);
  d6= (data>>4)+48;
  d7=(data&0x0f)+48;

  fifoPullData(&data);
  d8= (data>>4)+48;
  d9=(data&0x0f)+48;

  fifoPullData(&data);
  d10= (data>>4)+48;
  d11=(data&0x0f)+48;

// En las siguientes líneas se saca de la fifo los ultimos 9 caracteres de la trama que no se usan
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);
  fifoPullData(&data);

}

if(printtime<=tiempo2_PRINT-tiempo1_PRINT){
  tiempo1_PRINT=tiempo2_PRINT;
  //Serial.print("PWM: ");
  Serial.print(acupwm);
  //Serial.print("RPM: ");
  Serial.print(" ");
  Serial.print(RPM);
  RPM=0;
  //Serial.print("PESO: ");
  Serial.print(" ");
  Serial.print(d10);
  Serial.print(d11);

  Serial.print(d8);
  Serial.print(d9);

  Serial.print(d6);
  Serial.print(d7);

  Serial.print(d4);
  Serial.print('.');
  Serial.println(d5);
}
}

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

```

void InterrupcionRPM() {
    tiempo2_RPM=micros();
    RPMflag=1;
}

void serialEvent2() {
    char inChar;
    inChar = (char)Serial2.read();
    fifoPushData(inChar);
}

```

Los siguientes tres códigos corresponden el programa creado para hacer la representación gráfica del *drone* en MATLAB.

Drawdrone:

```

function [] = DrawDrone( DroneData, x, y, z, N)
%DrawDrone Draws a basic drone model
% DroneData is a Matriz where each column correspond to one of the
% models points. Beside each models point have a pair: point 1 and 2 are
% pairs, 3 and 4, 5 and 6, and so on.

persistent hVec;

l = length(hVec);
[m,n]=size(DroneData);
if m ~= 3
    return;
end
% si hvec no esta inicializada entonces inicializar estructura grafica
if l==0
    for i=1:n
        hVec(i) = line();
    end
end

% El drone no se mueve respecto a las coordenadas 'x', 'y' y 'z'
% aquí sumamos la posición actual del drone
DroneData1(1,:) = DroneData(1,:)+x;
DroneData1(2,:) = DroneData(2,:)+y;
DroneData1(3,:) = DroneData(3,:)+z;

% Actualiza coordenadas de los booms del Drone
for i=1:N
    set(hVec(i),'XData',[x DroneData1(1,i)],'YData',[y DroneData1(2,i)],'ZData',[z DroneData1(3,i)],'Color','k','Marker','o');
end

% Actualiza coordenadas de las flechas
]for i=N+1:2:n
    set(hVec(i),'XData',[DroneData1(1,i) DroneData1(1,i+1)],'YData',[DroneData1(2,i) DroneData1(2,i+1)],'ZData',[DroneData1(3,i) DroneData1(3,i+1)],'Color','r');
    set(hVec(i+1),'XData',[DroneData1(1,i) DroneData1(1,i)],'YData',[DroneData1(2,i) DroneData1(2,i)],'ZData',[DroneData1(3,i) DroneData1(3,i)],'Color','b');
end

```

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

Transform:

```
function [ DronDataNew ] = Transform( ro, teta, gamma, DroneData )
% Transform This función applies rotation in pitch, roll and yaw axes to
% the basic graphic model of a drone in a cartesian reference system.
%   ro is pitch axe
%   teta is yaw axe
%   gamma is roll axe
%   DroneData is a Matriz where each column correspond to one of the
%   models points.
Tyz=[1 0 0; 0 cos(ro) -sin(ro); 0 sin(ro) cos(ro)]; % pith
Txy=[cos(teta) sin(teta) 0; -sin(teta) cos(teta) 0; 0 0 1]; % yaw
Txz=[cos(gamma) 0 sin(gamma); 0 1 0; -sin(gamma) 0 cos(gamma)]; % roll

T=Tyz*Txy*Txz;

DronDataNew = T*DroneData;

end
```

Mygraficx:

```
cleae all
close all
l=20;
ro=0;
gamma=0;
teta=0;
x=0;
y=0;
z=0;
% Vertices del drone
% b1=[0 1 0];
% b2=[1 0 0];
% b3=[0 -1 0];
% b4=[-1 0 0];

omega=pi/4; %Ángulo de separación del los boom respecto al eje central.
% b1 y b2 son las coordenadas del boom1
b1=[l*sin(omega) l*cos(omega) 0];
b2=[-l*cos(omega) l*sin(omega) 0];
% b3 y b4 son las coordenadas del boom2
b3=[-l*sin(omega) -l*cos(omega) 0];
b4=[l*cos(omega) -l*sin(omega) 0];

% Puntos flecha adelante
b5=[0 l*sin(pi/4) 0];
b6=[0 l*sin(pi/4)/2 0];
% Puntos flecha arriba
b7=[0 0 l*sin(pi/4)];
b8=[0 0 l*sin(pi/4)/2];
```

	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

FIRMA ESTUDIANTES _____ 

_____ *David Alonso Sepúlveda.*

_____ *David Morales*

FIRMA ASESOR _____ 

FIRMA ASESOR _____ *Guillermo*

FECHA ENTREGA: _____

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO ___ ACEPTADO ___ ACEPTADO CON MODIFICACIONES _____

ACTA NO. _____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD _____

 Institución Universitaria	INFORME FINAL DE TRABAJO DE GRADO	Código	FDE 089
		Versión	03
		Fecha	2015-01-22

ACTA NO. _____
FECHA ENTREGA: _____