



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

ESCUELA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN MECATRÓNICA

TEMA:

“IDENTIFICACIÓN DE UN CUADRICÓPTERO”
“*QUADCOPTER IDENTIFICATION*”

AUTOR: FRANCISCO XAVIER GRIJALVA MANTILLA

DIRECTOR: CARLOS XAVIER ROSERO

IBARRA-ECUADOR
SEPTIEMBRE 2017



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA
UNIVERSIDAD TÉCNICA DEL NORTE
IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto *Repositorio Digital Institucional*, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR			
CÉDULA DE IDENTIDAD	1003727870		
APELLIDOS Y NOMBRES	GRIJALVA MANTILLA FRANCISCO XAVIER		
DIRECCIÓN	CUIDADELA MUNICIPAL		
EMAIL	fxgrijalvam@utn.edu.ec - pancho20g93@gmail.com		
TELÉFONO FIJO	06 – 2510 – 565	TELÉFONO MÓVIL	0998684685
DATOS DE LA OBRA			
TÍTULO	“QUADCOPTER IDENTIFICATION”		
AUTOR	“IDENTIFICACIÓN DE UN CUADRICÓPTERO”		
FECHA	FRANCISCO XAVIER GRIJALVA MANTILLA		
PROGRAMA	SEPTIEMBRE DE 2017		
TÍTULO POR EL QUE OPTA	PREGRADO		
DIRECTOR	INGENIERO EN MECATRÓNICA		
	CARLOS XAVIER ROSERO C.		

AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Francisco Xavier Grijalva Mantilla con cédula de identidad Nro. 1003727870, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior, Artículo 144.

CONSTANCIA

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló sin violar derechos de autor de terceros, por lo tanto la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, Septiembre de 2017



Francisco Xavier Grijalva Mantilla
C.I.: 1003727870



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A
FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Francisco Xavier Grijalva Mantilla con cédula de identidad Nro. 1003727870, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado “QUADCOPTER IDENTIFICATION”, que ha sido desarrollado para optar por el título de Ingeniero en Mecatrónica, en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Ibarra, Septiembre de 2017


Francisco Xavier Grijalva Mantilla
C.I.: 1003727870



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CERTIFICACIÓN

En calidad de director del trabajo de grado “CUADCOPTER IDENTIFICATION”, presentado por el egresado FRANCISCO XAVIER GRIJALVA MANTILLA, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, Septiembre de 2017



Carlos Xavier Rosero
DIRECTOR DE TESIS



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
DECLARACIÓN

Yo, Francisco Xavier Grijalva Mantilla con cédula de identidad Nro. 1003727870, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte - Ibarra, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Ibarra, Septiembre de 2017

Francisco Xavier Grijalva Mantilla
C.I.: 1003727870

Agradecimiento

Gracias a mi madre por haberme dado la oportunidad de tener una excelente educación a lo largo de mi vida. A mi hermano por ser un ejemplo de vida a seguir y siempre estar a mi lado apoyándome en cada decisión tomada a lo largo de mi vida estudiantil. A mi familia por siempre estar pendientes de mi desempeño académico. De una manera especial a mi tutor de tesis Xavier Rosero, gracias por la confianza depositada en mi y por ser más que un profesor, ser un amigo que supo guiarme por el camino correcto en el desarrollo del proyecto. Al personal Docente de la carrera de ingeniería en mecatrónica por haber compartido su conocimiento y experiencias como ejemplo para nuestro óptimo desempeño en una vida profesional, inculcando valores propios de un ingeniero. A mi novia Karen Rodríguez por siempre estar a mi lado aconsejándome como compañera y pareja, demostrándonos que trabajando en equipo siempre podremos salir adelante y alcanzar todas las metas que nos proponemos.

Dedicatoria

El presente proyecto esta dedicado principalmente a mi amada madre María Mantilla por ser el pilar más importante en mi vida, y ser la persona que siempre estuvo conmigo en los momentos más difíciles y de la cual aprendí los valores más importantes como son el amor, respeto y responsabilidad. A mi abuelita Isabel Moya y mi querido padre Arturo Grijalva que siempre me brindaron su apoyo incondicional a la largo de la carrera. A mi hermano Henry que descansa en paz, por guiarme desde pequeño como una persona de bien. A mi hermano Danilo Grijalva quien siempre supo como animarme cuando mis ánimos estaban bajos y siempre confió ciegamente en mi.

Francisco Xavier Grijalva Mantilla

Resumen

En las últimas décadas el avance de la tecnología ha creado nuevos campos de investigación. Uno de los campos más interesantes es el de la robótica móvil y su relación con los llamados *drones* o vehículos aéreos no tripulados.

En este trabajo se presenta el desarrollo y validación de un modelo matemático que describe la dinámica del desplazamiento en vuelo del cuadricóptero desde un punto inicial a un punto final deseado. Se toman en consideración varios aspectos tales como, al sistema se lo identifica como un sistema lineal y continuo en el tiempo, el grado de la planta obedece al comportamiento del sistema y obteniendo además la velocidad que generan los motores al producir este desplazamiento.

En la realización de este proyecto se utiliza el cuadricóptero ARDrone 2.0 de la empresa Parrot, el metasistema operativo ROS y el lenguaje python. En cuanto al ambiente de trabajo se ha escogido un entorno cerrado para evitar perturbaciones en la adquisición de datos desde el cuadricóptero. El software Matlab con su herramienta de identificación de sistemas llamada *ident* la cual es usada para la identificación de los parámetros del cuadricóptero, y la posterior aplicación de un modelo ARX.

Finalmente, el análisis de los modelos resultantes de la identificación que cuentan con un mínimo de error tanto en las pruebas realizadas FPE (Error de predicción final) y MSE (Función de pérdida), además estos modelos se ajustan a la dinámica real del sistema siendo estas ecuaciones validas a implementar en el sistema real.

Abstract

In the last decades the technological advances have influenced the creation of new research fields, one of the most interesting is the mobile robotics field and its relation with the so called drones or unmanned aerial vehicles.

In this work is shown the development and validation of a mathematic model which describes the mechanical displacement during the fly of the quadcopter from a starting point to a desired one. It is taken into consideration many aspects such as: the identification of the system which in this case is lineal and continuous regarding time, the plant degree which obeys to the behaviour of the system and also obtaining the generated speed in the motors when the displacement is produced.

In the making of this project the quadcopter used is the ARDrone 2.0 which is produced by the company Parrot, the ROS operative metasytem and the programming language Python. Regarding the work environment is an enclosed surrounding this was done in order to avoid perturbations in the data acquisition from the quadcopter. In this work is also used Matlab software and its system identification tool called Ident which is used for the identification of the quadcopter parameters and the following application of an ARX model.

Finally, the analysis of the resultant models of identification which have a minimal mistake range in the FPE (final prediction error) try outs and MSE (mean squared error), besides these models are adjusted to the real dynamic of the system being these valid equations to be implemented in the real system.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.2.1. Objetivo Principal	2
1.2.2. Objetivos Específicos	2
1.3. Antecedentes	2
1.4. Problema	2
1.5. Justificación	3
1.6. Alcance	3
2. Revisión Literaria	4
2.1. Identificación de sistemas	4
2.1.1. Obtención de modelos	4
2.1.2. Identificación de sistemas	4
2.1.2.1. Tipos de identificación	5
2.1.3. Tipos de modelos	5
2.1.4. Métodos de identificación	6
2.1.4.1. Dependiendo del tipo de modelo obtenido	6
2.1.4.2. Dependiendo de la aplicación	7
2.1.5. Estructura del modelo	7
2.1.6. Etapas a seguir para la identificación de un modelo	9
2.2. Cuadricóptero	11
2.2.1. Vehículo aéreo no tripulado, UAV	11
2.2.2. Selección de cuadricóptero	12
2.2.2.1. Grabación de video	13
2.2.2.2. Electrónica	13
2.2.2.3. Motorización	13
2.2.2.4. Estructura del cuadricóptero	14
3. Metodología	16
3.1. Introducción	16
3.2. Variables del cuadricóptero	17

3.2.1.	Variables manipuladas	17
3.2.2.	Variables controladas	17
3.2.3.	Acceso a los datos del cuadricóptero	17
3.3.	Planificación de experimentos de vuelo para la adquisición de datos	20
3.4.	Muestras obtenidas	20
3.5.	Estrategia de identificación	23
3.6.	Modelo y validación para el eje y	23
3.7.	Modelo y validación para el eje x	27
4.	Conclusiones y recomendaciones	32
4.1.	Conclusiones	32
4.2.	Recomendaciones	33
4.3.	Trabajo futuro	33
A.	Datos del cuadricóptero	35
B.	Listado de códigos	36
B.1.	Programas implementados	36
B.1.1.	Carga de datos a Matlab	36
B.1.2.	Gráficas de datos en Matlab	36
B.1.3.	Identificación	37

Índice de figuras

2.1. Proceso de identificación de un sistema	5
2.2. Estructura BJ	8
2.3. Estructura OE	8
2.4. Modelo ARX	8
2.5. Modelo ARMAX	9
2.6. Flujograma del proceso de identificación	10
2.7. OQ-2 radioplane	11
2.8. Estación UAV	12
2.9. Ángulos principales	14
2.10. Movimiento de hélices	15
2.11. Ángulo Roll	15
2.12. Ángulo Pitch	15
2.13. Ángulo Yaw	15
3.1. Variables que interactúan en el sistema	17
3.2. Flujo de información entre el computador y el AR.Drone 2.0	18
3.3. Pseudocódigo de comunicación	19
3.4. Envío y recepción de datos	19
3.5. Entradas y salidas	20
3.6. Plano de vuelo	21
3.7. Datos del sistema	22
3.8. Datos para indentificación y validación del sistema	24
3.9. Estructura de selección modelo ARX	24
3.10. Modelos posibles para el eje y	25
3.11. Análisis residual	26
3.12. Mejores ajustes	26
3.13. Datos de identificación y validación del eje x	28
3.14. Estructura de selección modelo ARX	28
3.15. Posibles modelos para el eje x	29
3.16. Mejores ajustes de modelos	30
3.17. Análisis residual	30

Índice de cuadros

2.1. Diferentes estructuras de modelos	9
3.1. Parámetros de los modelos en el eje y	25
3.2. Parámetros de los modelos en el eje x	29

Listings

B.1. Ingreso de archivo .txt a Matlab	36
B.2. Gráfica de datos en Matlab	36
B.3. Proceso de identificación en Matlab	37

Capítulo 1

Introducción

Este trabajo de grado ha sido realizado con el *Grupo de Investigación en Sistemas Inteligentes de la Universidad Técnica del Norte (GISI-UTN)*.

1.1. Motivación

El funcionamiento de los procesos industriales ha cambiado drásticamente en las últimas décadas debido principalmente a la evolución de la tecnología del computador y a la automatización de los procesos. L. Ljung nos dice: “En la actualidad, cada vez más, el trabajo de un ingeniero consiste en la realización de modelos matemáticos”. Los modelos son utilizados en áreas muy diversas como: bioingeniería, construcción, economía, meteorología, procesos químicos, etc. y en su campo de utilización destacan aplicaciones como: control, supervisión, predicción, simulación y optimización [1].

La mejora continua para aumentar la competitividad, y las innovaciones en procesos, hacen necesario el desarrollo de herramientas que permitan maximizar la eficiencia de los procesos, además que ayuden al operador humano en las tareas de supervisión, tareas destinadas a la detección y diagnóstico de fallos, comparando el proceso con un modelo de simulación, ya que es imprescindible disponer de un modelo matemático preciso del proceso a controlar que nos permita conocer su comportamiento, además que nos ayude al diseño de los algoritmos de control [2].

La identificación de sistemas se ocupa del problema de construir modelos matemáticos de sistemas dinámicos basados en datos observados del sistema, con lo cual en este documento se identificará el cuadricóptero ARDrone 2.0 mediante la recopilación de información de la odometría del robot y el uso de software libre.

1.2. Objetivos

1.2.1. Objetivo Principal

- Obtener la función de transferencia de un cuadricóptero.

1.2.2. Objetivos Específicos

- Estudiar los métodos de identificación de sistemas.
- Seleccionar un cuadricóptero con las características idóneas para su identificación con herramientas libres.
- Realizar pruebas para la adquisición de datos de vuelo..
- Desarrollar la identificación del sistema.
- Validar la función de transferencia del cuadricóptero.

1.3. Antecedentes

El estudio de la cinemática y la dinámica ayuda a comprender la física del quadrotor y su comportamiento. La identificación, el modelado y la determinación de la estructura del algoritmo de control son muy importantes para lograr una mejor estabilización y robustez del sistema pudiendo este ser probado mediante simuladores que proporcionan una salida gráfica 3D, así como datos de estado, para la depuración del rendimiento del sistema en la planta real [3].

Los experimentos realizados en un AR.Drone se centran en el modelado y la identificación, el control y el diseño del filtro, la simulación y la animación del quadrotor. Las ecuaciones de movimiento del cuadricóptero se derivaron de la cinemática y de la dinámica de los quadrotores comunes. La identificación se llevó a cabo con totalidad, incluida su baja resolución de sensores a bordo, como el giro de velocidad y el altímetro, además de la aplicación de diversos tipos de filtros, tales como filtro de paso bajo / alto, filtro complementario y filtro de Kalman para su óptimo desempeño [4].

1.4. Problema

Una de las tecnologías que más destacan en estos últimos años es la de los denominados *drones*, pequeños aparatos voladores no tripulados, que pueden ser controlados en forma remota, convirtiéndose en una herramienta indispensable para diversos propósitos [5]. El origen del desarrollo de estos vehículos pertenece a fines militares, ya que dan la posibilidad de realizar operaciones de alto riesgo, e incluso la de sobrevolar zonas en conflicto para la vigilancia o

recogida de información [6].

Empresas, instituciones y hogares ha optado por la utilización e implementación de drones principalmente como medios de video vigilancia, monitoreo, traslado de productos, rastreo de personas o simplemente por diversión, por lo que su uso se enfoca en tareas que el humano no puede o no quiere realizar [7],[8].

Con la implementación de métodos de identificación de sistemas se ha logrado potencializar el uso de los drones [3], como vehículos aéreos no tripulado (UAS, unmanned aircraft system) capaces de realizar tareas autónomas [9].

En lo concerniente a la realidad de la Universidad Técnica del Norte y particularmente a la Facultad de Ingeniería en Ciencias Aplicadas, no existen robots aéreos que se utilicen como herramientas de investigación. Esto repercute negativamente en la aplicación de distintos algoritmos de control sobre este tipo de sistema aéreo en las asignaturas de robótica e inteligencia artificial.

1.5. Justificación

Un drone es un vehículo aéreo no tripulado que tiene un gran potencial en áreas muy diversas, ya que pueden desplazarse rápidamente sobre un terreno irregular o accidentado y superar cualquier tipo de obstáculo [10]. Incluso permitiría a los investigadores responder a preguntas en tiempos pequeños con recursos mínimos y también hallar nuevas aristas en las investigaciones, haciendo uso de una herramienta de análisis e interpretación más cercana, más económica y práctica.

La relevancia del proyecto está en obtener la función de transferencia del cuadricóptero como un proyecto académico la cual será usada como una herramienta de simulación rápida, además de la aplicación de algoritmos de control, robótica e inteligencia artificial. Esto permitirá observar el desempeño de los mismos y obtener una base para el posterior diseño y desarrollo de sistemas robóticos móviles autónomos.

1.6. Alcance

El proyecto deberá constar de una interfaz gráfica de usuario para la visualización, envío y recepción de datos de vuelo. Lo expuesto anteriormente y la interfaz gráfica deberán ser desarrollados con software y hardware libres.

Capítulo 2

Revisión Literaria

Este capítulo detalla la teoría base utilizada para la ejecución de este proyecto.

2.1. Identificación de sistemas

2.1.1. Obtención de modelos

Existen dos métodos principales para obtener el modelo de un sistema, los cuales son descritos a continuación, esta información fue recuperada de [11].

Modelo Teórico: Se trata de un método analítico, en el que se recurre a leyes básicas de la física para describir el comportamiento dinámico de un fenómeno o proceso. El modelado teórico tiene un campo de aplicación restringido a procesos muy sencillos de modelar, o a aplicaciones en que no se requiera gran exactitud en el modelo obtenido.

Identificación del sistema: Se trata de un método experimental que permite obtener el modelo de un sistema a partir de datos reales recogidos de la planta bajo estudio. La estructura del modelo obtenido a partir del conocimiento físico de la planta posee un conjunto de parámetros desconocidos y que sólo se pueden determinar experimentando sobre el sistema real. De ahí la necesidad de recurrir a los métodos de identificación de sistemas.

2.1.2. Identificación de sistemas

En este artículo se presentan los conceptos de la teoría de identificación de sistemas, siendo esta el arte y la ciencia que se encarga del problema de la realización de modelos matemáticos de sistemas, partiendo del análisis de datos experimentales, mediciones y observaciones del sistema entre su entrada y salida [1]. La Figura 2.1 ilustra el diagrama de bloques de la identificación de sistemas, esta imagen se tomó de [14].

Las técnicas de identificación de sistemas han cobrado gran relevancia debido a que los sistemas dinámicos abundan y las técnicas de control actuales requieren de modelos matemáticos cada vez más exactos para el análisis y el diseño. Además de su aplicación en diversas áreas

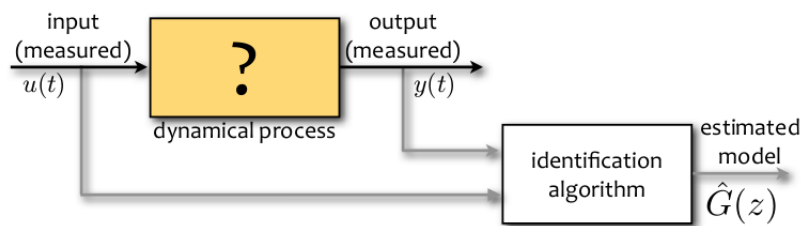


Figura 2.1: Proceso de identificación de un sistema

del conocimiento (ingenierías, economía, biotecnología, etc.), donde se requiere de un modelo preciso para fines de análisis, predicción, simulación, diseño y control [12],[13].

2.1.2.1. Tipos de identificación

El enfoque de la identificación se puede realizar en función de la estructura del modelo, y del comportamiento físico o no del mismo. Podemos distinguir como nos indica [14]:

Black-box: Los parámetros del modelo no tienen una interpretación física. Un modelo basado en leyes fundamentales es muy complicado o se desconoce. El modelo de la estructura y sus parámetros son completamente desconocidos, estos se estiman de datos de entrada y salida del sistema.

Grey-box: Algunas partes del sistema son modeladas basándose en principios fundamentales, y otras como una caja negra. Algunos de los parámetros del modelo pueden tener una interpretación física; a este tipo de modelos también se les conoce como *Taylor-made*, estimando sólo los parámetros no conocidos. La estructura del modelo es parcialmente conocida de los principios fundamentales (Leyes de Newton) y el resto del sistema se lo construye por datos del mismo.

White-box: La estructura del modelo se obtiene a partir de leyes fundamentales. Los parámetros tienen una interpretación física. La estructura del modelo es basada en las leyes fundamentales (Leyes de Newton) y los parámetros del modelo son estimados de datos medidos.

2.1.3. Tipos de modelos

El artículo [2] nos presenta que existen varias formas de catalogar los modelos matemáticos entre ellos tenemos:

Modelos mentales: Sin formalismo matemático.

Modelos no paramétricos: Se caracterizan mediante gráficos, diagramas o representaciones que describen las propiedades dinámicas mediante un número no finito de parámetros (respuesta al impulso, al escalón, o en frecuencia).

Modelos paramétricos o matemáticos: Describen las relaciones entre las variables del sistema mediante expresiones matemáticas, (ecuaciones diferenciales en sistemas continuos y ecuaciones de diferencias en sistemas discretos). En función del tipo de sistema y de la representación matemática utilizada, los sistemas pueden clasificarse en:

Determinísticos o estocásticos: Se dice que un modelo es determinístico, cuando expresa la relación entre entradas y salidas mediante una ecuación exacta; se estudia la relación entre la entrada y la salida con una parte no conocida. Un modelo es estocástico si posee un cierto grado de incertidumbre. Quedan definidos mediante conceptos probabilísticos o estadísticos.

Dinámicos o estáticos: Un sistema es estático cuando la salida depende únicamente de la entrada en ese instante de tiempo. La función que relaciona las entradas con las salidas, es independiente del tiempo. En un sistema dinámico las salidas varían con el tiempo, el valor actual de la salida en función del tiempo transcurrido desde la aplicación de la entrada. Tienen como objetivo conocer el comportamiento dinámico de un proceso.

Continuos o discretos: Los sistemas continuos trabajan con señales continuas, se formalizan mediante ecuaciones diferenciales. Los sistemas discretos trabajan con señales muestreadas, se describen por medio de ecuaciones en diferencias.

De parámetros concentrados: No se considera la variación en función del espacio.

Lineales o no lineales: Un sistema lineal se define por una función matemática lineal; siendo $y'(t) + a y_2(t) + b y_3(t) = u(t)$, no lineal.

2.1.4. Métodos de identificación

Los artículos [2],[11] nos muestran que existen diversos métodos de identificación, que pueden clasificarse según distintos criterios mostrados a continuación:

2.1.4.1. Dependiendo del tipo de modelo obtenido

Métodos no paramétricos: que permiten obtener modelos no paramétricos del sistema bajo estudio.

Análisis de la respuesta transitoria: consiste en obtener la respuesta del sistema a un impulso o a un escalón.

Análisis de correlación: pertenece al dominio temporal, se obtiene la función de correlación entre las variables de entrada y salida.

Análisis en frecuencia: son utilizadas directamente para estimar la respuesta frecuencial del sistema. Se determina mediante el análisis de Fourier o el análisis espectral, dependiendo la forma de las señales de entrada.

Métodos paramétricos: que permiten obtener modelos paramétricos. Estos métodos requieren la elección de una posible estructura del modelo, de un criterio de ajuste de parámetros, y por último de la estimación de los parámetros que mejor ajustan el modelo a los datos experimentales.

2.1.4.2. Dependiendo de la aplicación

Métodos de identificación off-line (a posteriori): utilizados en aquellas aplicaciones en que no se requiera un ajuste continuado del modelo. En estos casos, se realiza la identificación previa de la planta, considerándose que la validez de los parámetros obtenidos no se verá alterada con el paso del tiempo.

Métodos de identificación on-line (identificación recursiva): en los que los parámetros se van actualizando continuamente a partir de los nuevos datos de entrada-salida obtenidos durante la evolución del proceso. Estos métodos son muy utilizados en sistemas de control adaptativo.

Dependiendo del criterio de ajuste de los parámetros: existen diversos métodos matemáticos para ajustar los parámetros de una estructura a un conjunto de datos de entrada-salida. Algunos de los más utilizados en el campo de la identificación son el método de mínimos cuadrados y el método de las variables instrumentales.

2.1.5. Estructura del modelo

La elección de la estructura del modelo como podemos encontrar en [11],[14] depende de la dinámica y las características del sistema, el uso de un modelo con más libertad o más parámetros no siempre es la mejor elección ya que puede resultar en el modelado de dinámicas inexistentes y características de ruido.

Modelo BJ: La estructura de Box-Jenkins(BJ) proporciona un modelo completo con propiedades de perturbación modeladas separadamente de la dinámica del sistema. el modelo de Box-Jenkins es útil cuando se tiene disturbios que entran tarde al proceso. Por ejemplo, el ruido de medición en la salida es una perturbación tardía en el proceso. La Figura 2.2 muestra el diagrama de bloques del modelo.

Modelo OE: La estructura del modelo Output-Error(OE) describe la dinámica del sistema por separado. No se utilizan parámetros para modelar las características de perturbación. La Figura 2.3 muestra el diagrama de bloques del modelo.

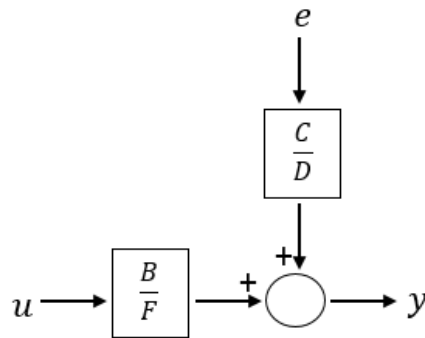


Figura 2.2: Estructura BJ

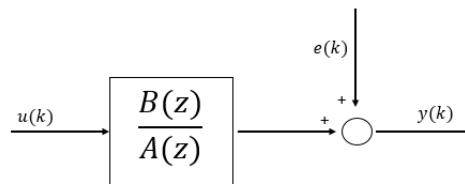


Figura 2.3: Estructura OE

Modelo ARX: Es el modelo más fácil que tiene una señal de estímulo, la estimación del modelo ARX es el más eficiente de los métodos de estimación polinomial, es aconsejable usar cuando el orden del modelo es alto. La desventaja de este modelo son las perturbaciones que son parte del mismo. La Figura 2.4 muestra el diagrama de bloques del modelo.

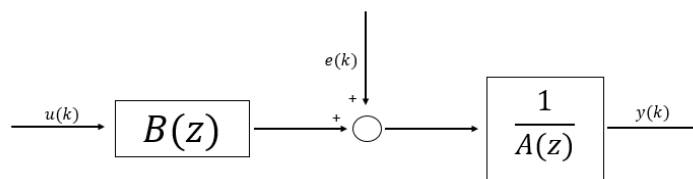


Figura 2.4: Modelo ARX

Modelo ARMAX: La estructura del modelo incluye la dinámica de perturbación, estos modelos son útiles cuando las perturbaciones dominantes entran al inicio del proceso, modela mejor las perturbaciones. La Figura 2.5 muestra el diagrama de bloques del modelo.

En el Cuadro 2.1 presenta los diferentes tipos de modelos, condiciones y estructuras utilizadas.

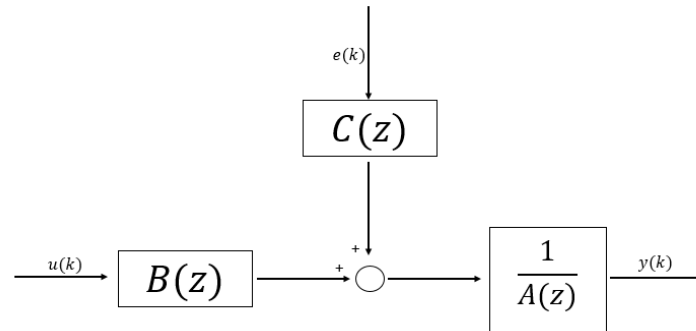


Figura 2.5: Modelo ARMAX

Cuadro 2.1: Diferentes estructuras de modelos

Tipo de modelo	Condición	Estructura resultante
Modelo BJ	$A(q^{-1}) = 1$	$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t)$
Modelo OE	$C(q^{-1}) = D(q^{-1}) = A(q^{-1}) = 1$	$y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + e(t)$
Modelo ARX	$F(q^{-1}) = D(q^{-1}) = C(q^{-1}) = 1$	$A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)$
Modelo ARMAX	$F(q^{-1}) = D(q^{-1}) = 1$	$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t)$

2.1.6. Etapas a seguir para la identificación de un modelo

Los artículos [2],[11] en términos generales, muestran el proceso de identificación que comprende los siguientes pasos:

1. Obtención de datos de entrada-salida

Para ello se debe excitar el sistema mediante la aplicación de una señal de entrada y registrar la evolución de sus entradas y salidas durante un intervalo de tiempo.

2. Tratamiento previo de los datos registrados

Los datos registrados están generalmente acompañados de ruidos indeseados u otro tipo de imperfecciones que puede ser necesario corregir antes de iniciar la identificación del modelo. Se trata, por tanto, de “preparar” los datos para facilitar y mejorar el proceso de identificación.

3. Elección de la estructura del modelo

Si el modelo que se desea obtener es un modelo paramétrico, el primer paso es determinar la estructura deseada para dicho modelo. Este punto se facilita en gran medida si se tiene un cierto conocimiento sobre las leyes físicas que rigen el proceso.

4. Obtención de los parámetros del modelo

A continuación, se procede a la estimación de los parámetros de la estructura que mejor

ajustan la respuesta del modelo a los datos de entrada-salida obtenidos experimentalmente.

5. Validación del modelo

El último paso consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión. Si se llega a la conclusión de que el modelo no es válido, se deben revisar los siguientes aspectos como posibles causas:

- El conjunto de datos de entrada-salida no proporciona suficiente información sobre la dinámica del sistema.
- La estructura escogida no es capaz de proporcionar una buena descripción del modelo.
- El criterio de ajuste de parámetros seleccionado no es el más adecuado.

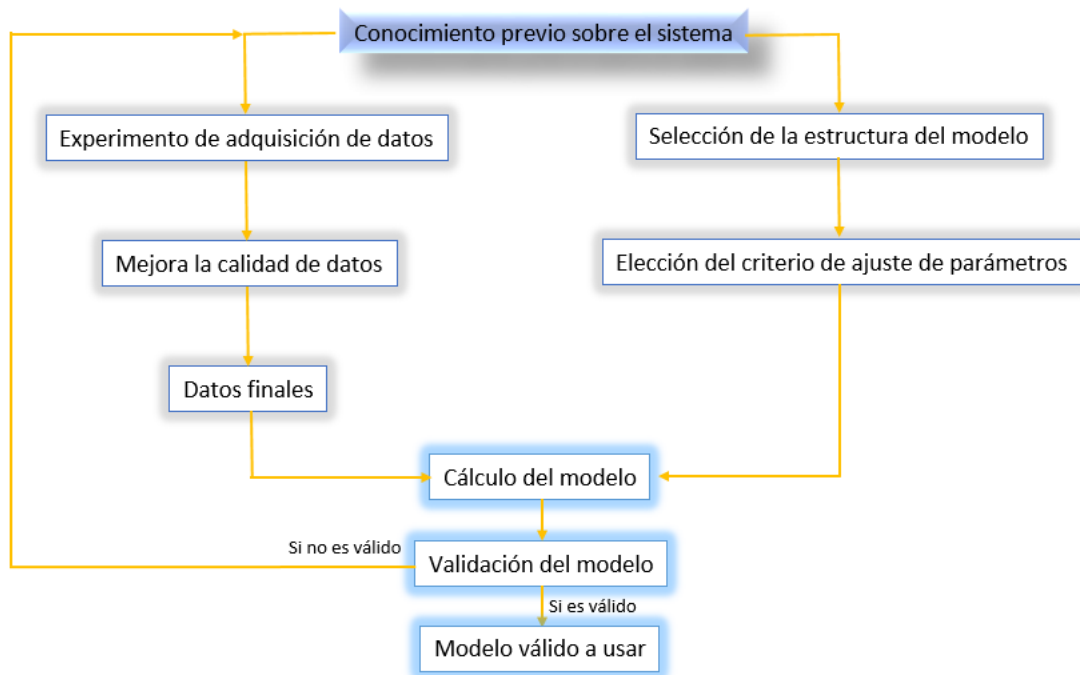


Figura 2.6: Flujograma del proceso de identificación

2.2. Cuadricóptero

2.2.1. Vehículo aéreo no tripulado, UAV

Drone es el término popular por el cual un UAV¹ y un UAS² son conocidos. Etimológicamente la palabra drone proviene del inglés antiguo dron q significa zángano (abeja macho), pero desde el año 1946 se asignó el término a aeronaves sin piloto a bordo [15].

Hoy en día con el avance de las nuevas tecnologías, los drones se han convertido en una herramienta indispensable para ciertos propósitos en el campo de la informática. El origen del desarrollo de estos vehículos pertenece a fines militares, ya que dan la posibilidad de realizar operaciones de alto riesgo, o incluso la de sobrevolar una zona en conflicto para la vigilancia o recogida de información. En la Figura 2.7 presenta un drone utilizado en la Segunda Guerra Mundial.



Figura 2.7: OQ-2 radioplane

Los drones surgieron por primera vez en la primera Guerra Mundial, pero no fue hasta la segunda cuando se emplearon como método de entrenamiento de los cañones antiaéreos. Sin embargo, el auge de los llamados VANT (Vehículos Aéreos no Tripulados) por sus siglas en español, llegó bastante más tarde, a finales del siglo XX. En la guerra de Vietnam se utilizaron drones de reconocimiento y se sabe que anteriormente, durante la guerra del Golfo y la guerra de Bosnia, también se pusieron en marcha ciertos dispositivos con fines bélicos [8].

Existen una gran variedad de UAVs, desde vehículos de grandes dimensiones para altos vuelos y/o grandes distancias de vuelos, hasta pequeñas dimensiones llegando incluso a pocos centímetros. El desarrollo de un UAV se basa en el tipo de acción que va a llevar a cabo, y la distancia que tendrá que recorrer [6].

¹Vehículo aéreo no tripulado, por sus siglas en inglés Unmanned Aerial Vehicle

²Sistema aéreo no tripulado por sus siglas en inglés Unmanned Aircraft System

Si tuviésemos que clasificar a los drones en tres tipos esenciales, éstos serían los siguientes:

- Drones de reconocimiento.
- Drones de combate.
- Drones de investigación y desarrollo.

En este punto es necesario aclarar que vehículo no tripulado, UAV, no tiene un significado autónomo, sino que estará comunicado desde un operador de tierra, sean pilotos, controladores o cualquier otro tipo de operario relacionado con la monitorización de la aeronave. Una vez superado el reto de la creación de vehículos no tripulados, se investigó en otro nivel los llamados UAS, Unmanned Aircraft System. Un UAS se trata de la evolución directa de un UAV. Los UAVs pueden estar controlados remotamente desde una estación de tierra por un operador como se muestra en la Figura 2.8, en cambio, los UAS son autónomos y seguirán una trayectoria ya predefinida, o un vuelo con los recursos de los propios sensores. Existen dos estaciones que pueden manejar información del UAV, la estación de tierra y la estación a bordo del UAV. Dependiendo de cuan autónomo sea el UAV, la estación de tierra realizará más o menos funciones de forma habitual [6].



Figura 2.8: Estación UAV

En las últimas tres décadas, el uso de drones a escala ha llegado a ser cotidiano para un sin número de aplicaciones. Aunque muchas de las tareas que realizan los drones actualmente todavía están ligadas con actividades militares, los drones están formando parte de la vida diaria de las personas. Además, lo más relevante es que hoy en día cualquier persona puede adquirir drones de diferentes tamaños y a precios [15].

2.2.2. Selección de cuadricóptero

El cuadricóptero AR.Drone 2.0 de la compañía francesa Parrot ha sido escogido para el desarrollo de este trabajo basándose en la cantidad de información y su capacidad para el desarrollo de aplicaciones con herramientas libres específicamente en el entorno ROS, gracias a

los controladores específicos para este entorno robótico, además de varias características en las cuales podemos destacar su estructura simple, suficiente sensorica, fácil mantenimiento y su bajo costo lo cual hace que este sea un drone apto para el estudio y aplicaciones prácticas [16].

La información básica del cuadricóptero es tomada de [17] y se describe a continuación:

2.2.2.1. Grabación de video

- Cámara HD (720p 30FPS).
- Objetivo gran angular (diagonal 92°).
- Formato de fotos(JPEG).
- Conexión Wi-Fi.

2.2.2.2. Electrónica

- Procesador 1GHz 32 bits ARM Cortex A8 con DPS video 800MHz TMS320DMC64x.
- Sistema operativo Linux 2.6.3.2.
- RAM DDR2 1GB a 200MHz.
- Giroscopio de 3 ejes, precisión de $2000^{\circ}/seg$.
- Acelerómetro de 3 ejes, precisión de $\pm 50mg$.
- Magnetómetro de 3 ejes, precisión de 6° .
- Sensor de presión de $\pm 10Pa$.
- Sensores ultrasónicos para medir altitud.
- Cámara vertical (QVGA 60 FPS) para medir su velocidad de vuelo.

2.2.2.3. Motorización

- 4 motores sin escobillas de tipo “inrunner”(14.5 vatios y 28500 rpm).
- Rodamiento de bolas en miniatura.
- Engranajes Nylatron.
- Rodamiento de bolas autolubricante de bronce.

2.2.2.4. Estructura del cuadricóptero

Su fabricante ofrece un detallado manual para desarrolladores junto con un kit para su desarrollo en software mostrado en [18], el cual permite conocer el funcionamiento básico del cuadricóptero.

La Figura 2.9 muestra los diferentes movimiento que puede realizar el AR.Drone 2.0 se obtiene cambiando los ángulos de y roll³, pitch⁴ y yaw⁵:

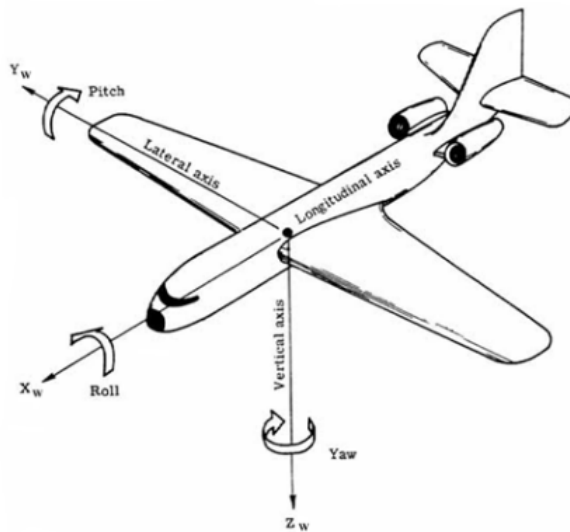


Figura 2.9: Ángulos principales

La estructura mecánica del sistema cuenta con cuatro motores conectados a los extremos en forma de cruz q se unen a la batería y al hardware RF. Un par de motores giran en sentido antihorario mientras que el otro par lo realiza en el sentido de las manecillas del reloj como podemos observar en la Figura 2.10:

- Al variar las velocidades de los rotores derecho e izquierdo produce el movimiento Roll, el cual permite el desplazamiento hacia adelante y atrás como se muestra en la Figura 2.11.
- Si se aplica diferentes velocidades a los rotores delanteros y traseros se produce el movimiento pitch como se muestra en la Figura 2.12.
- Al variar la velocidad de cada par de rotores, la dirección opuesta genera el movimiento yaw, el cual permite girar a la izquierda y derecha como se muestra en la Figura 2.13.

³Giro alrededor del eje X

⁴Giro alrededor del eje Y

⁵Giro alrededor del eje Z

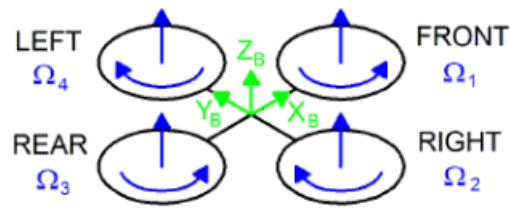


Figura 2.10: Movimiento de hélices

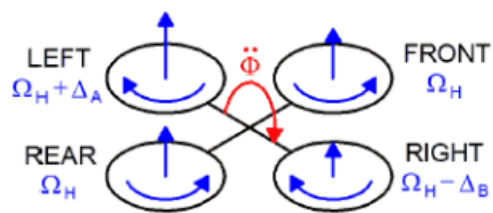


Figura 2.11: Ángulo Roll

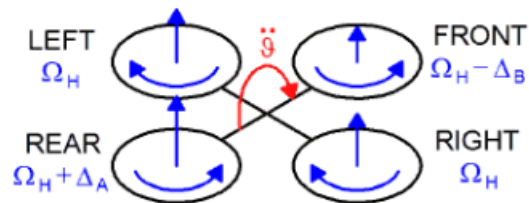


Figura 2.12: Ángulo Pitch

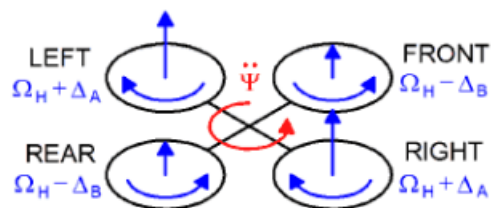


Figura 2.13: Ángulo Yaw

Capítulo 3

Metodología

En este capítulo se realiza una revisión condensada de la identificación y validación del cuadricóptero.

3.1. Introducción

Se eligió la adquisición de datos del cuadricóptero a través del software ROS, de sus siglas en inglés, Robot Operating System, las cuales, nos permiten ver que ROS es un sistema operativo, además se añade que no es solo un sistema operativo, es un meta-sistema operativo de código abierto para el desarrollo en el ámbito de la robótica.

Este sistema provee diferentes servicios, permite introducir librerías y herramientas para ayudar al desarrollo de software en aplicaciones vinculadas al campo de la robótica. Gracias, a su gran variedad de características como, la abstracción de hardware, los controladores de dispositivos, bibliotecas, visualizadores, paso de mensajes, o gestión de paquetes que son algunas de las funciones se esperarían de un sistema operativo. Para la instalación del software se toma como referencia la página oficial de ROS ubicada [19].

Para terminar esta introducción se hablará del Ardrone_autonomy que es un controlador ROS para el cuadricóptero Parrot AR-Drone 1.0 y 2.0. Este controlador se basa en el SDK oficial de AR-Drone versión 2.0.1. Este paquete se desarrolla en Autonomy Lab de la Universidad Simon Fraser por Mani Monajjemi y otros colaboradores. Para mayor información acerca de este controlador y su intalación en software libre referirse a [20].

3.2. Variables del cuadricóptero

3.2.1. Variables manipuladas

Son la cantidad o la condición de la planta que son modificadas con el único fin de influir sobre la variable controlada a través de la dinámica de la planta [1].

Las variables manipuladas del sistema corresponden a los ángulos pitch y roll como se observa en la Figura 3.1

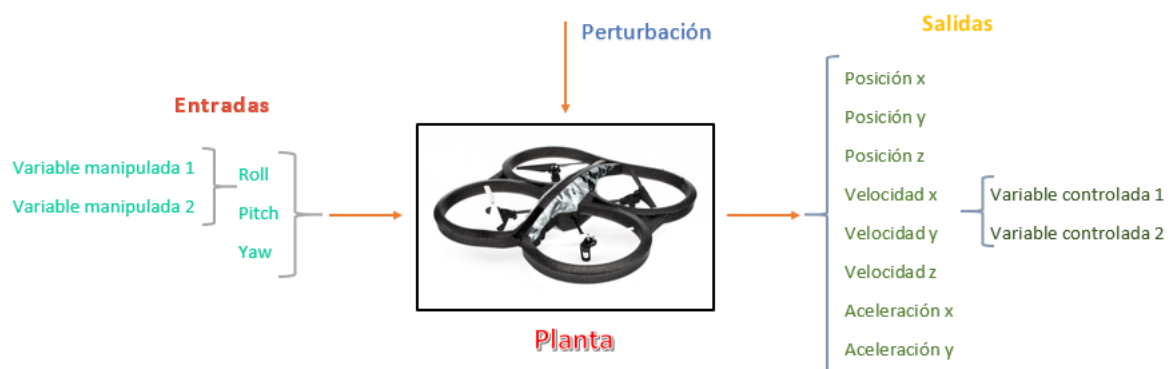


Figura 3.1: Variables que interactúan en el sistema

3.2.2. Variables controladas

Son la cantidad o la condición que se mide y controla, normalmente las variables controladas son consideradas las salidas del sistema [1]., las velocidades (v_x, v_y), se consideran como las variables controladas del sistema, teniendo como referencias las entradas pitch y roll, como se presenta en la Figura 3.1.

3.2.3. Acceso a los datos del cuadricóptero

Para realizar la identificación del cuadricóptero, es necesario conocer los parámetros dinámicos necesarios para los experimentos, accediendo a las variables o datos que posee el cuadricóptero.

En la Figura 3.2 se muestran los principales canales de comunicación, el enlace de video y los datos de los sensores que son enviados del cuadricóptero hacia el computador, el envío de datos

hacia el cuadricóptero es mediante el Wi-Fi¹ incorporado en el dispositivo el pseudocódigo de la comunicación del cuadricóptero se muestra en la Figura 3.3, la recepción de los datos pueden ser visualizados por medio del terminal o consola de ubuntu que se aprecia en la Figura 3.4.

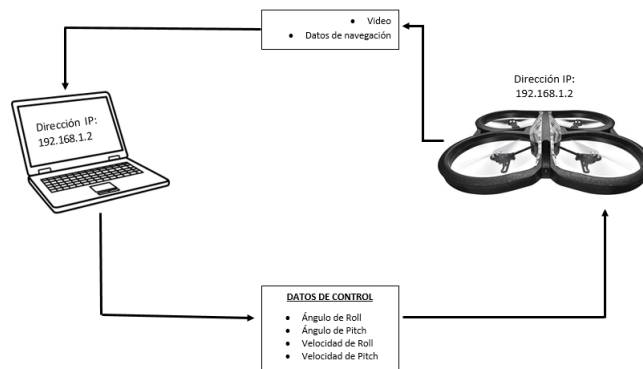


Figura 3.2: Flujo de información entre el computador y el AR.Drone 2.0

El movimiento roll genera el desplazamiento del cuadricóptero en el eje y , mientras el movimiento pitch genera un desplazamiento en el eje x como se muestra en la Figura 3.5, estas variables son adimensionales y pueden variar en un rango entre $[-1,1]$, los cuales corresponden a los límites máximos y mínimos de inclinación establecidos para el desarrollo del experimento de identificación.

La información obtenida del sistema debe contener datos relevantes y ciertas características sobre dicha información, como por ejemplo la elección de los datos idóneos a ser utilizados, el período de muestreo y la cantidad de datos implementados para la identificación y validación del sistema.

Los datos de salida obtenidos en el experimento de vuelo son almacenados en un archivo .txt, mostrados en el apéndice A. De esta manera se accede a las variables o datos que el cuadricóptero proporciona, las variables están detalladas por columnas en un fichero de texto que son incluidos en Matlab para el posterior desarrollo de la identificación.

La información que cada columna proporciona se detalla a continuación:

- Columna 1: Entrada Roll.
- Columna 2: Posición del eje y en metros.
- Columna 3: Velocidad del eje y en metros/segundos.
- Columna 4: Entrada Pitch.
- Columna 5: Posición del eje x en metros.

¹Mecanismo de conexión de dispositivos electrónicos de forma inalámbrica

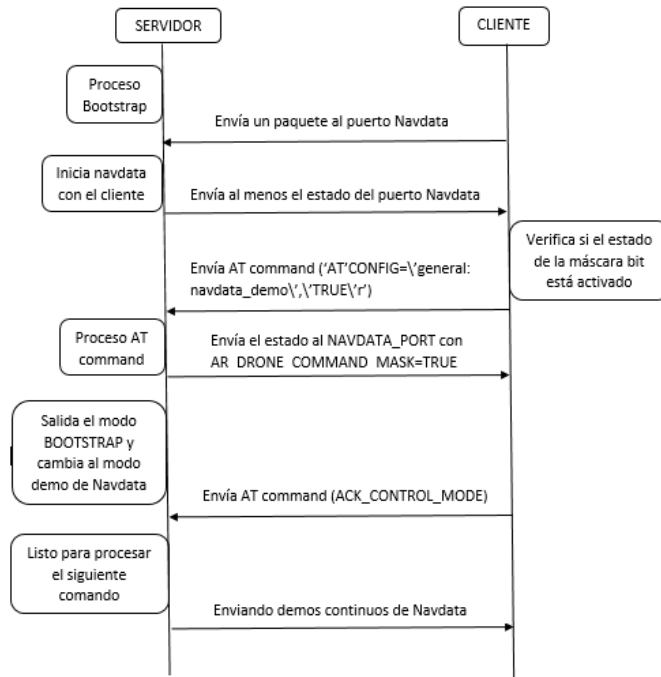


Figura 3.3: Pseudocódigo de comunicación

```

*Navdata.py (-/catkin_ws) - gedit
File Edit View Search Tools Documents Help
*Navdata.py x
z=XT yaw=XT %
(t,navdata.batteryPercent,navdata.vx,navdata.vy,navdata.alt,navdata.r
if __name__ == '__main__':
    rospy.init_node('example_node', anonymous=True)

    # subscribe to navdata (receive from quadrotor)
    rospy.Subscriber("/ardrone/navdata", Navdata, callback)

    # publish commands (send to quadrotor)
    pub_velocity = rospy.Publisher('/cmd_vel', Twist)
    pub_takeoff = rospy.Publisher('/ardrone/takeoff', Empty)
    pub_land = rospy.Publisher('/ardrone/land', Empty)
    pub_reset = rospy.Publisher('/ardrone/reset', Empty)

    print("ready!")
    rospy.sleep(3.0)

    print("takeoff..")
    pub_takeoff.publish(Empty())
    rospy.sleep(10.0)

    while not rospy.is_shutdown():
        # fly forward..
        pub_velocity.publish(Twist(Vector3(0.1,0,0),Vector3(0,0,0)))

        # fly backward..
        pub_velocity.publish(Twist(Vector3(-0.1,0,0),Vector3(0,0,0)))
        rospy.sleep(5.0)

        pub_land.publish(Empty())

ToffLand.py (-/catkin_ws) - gedit
File Edit View Search Tools Documents Help
ToffLand.py x
from std_msgs.msg import Empty
from geometry_msgs.msg import Twist

def takeoff():
    takeoff_pub.publish(Empty())

def land():
    pancho@pancho-Inspiron-5547: ~/catkin_ws
launch http://local... x pancho@pancho-Inspiron-5... x pancho@pancho-Inspiron-5... x
odometry message: time=1498251615.214905 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.040001
odometry message: time=1498251615.216793 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.040001
odometry message: time=1498251615.222791 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.040001
odometry message: time=1498251615.226811 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.040001
odometry message: time=1498251615.229239 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.040001
odometry message: time=1498251615.229266 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.039001
odometry message: time=1498251615.232847 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.038002
odometry message: time=1498251615.232882 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.038002
odometry message: time=1498251615.235843 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.038002
odometry message: time=1498251615.238405 battery=25.000000 vx=0.000000
0000 z=0.000000 yaw=-84.038002
odometry message: time=1498251615.246650 battery=25.000000 vx=0.000000
000000 z=0.000000 yaw=-84.038002
pancho-Inspiron-5547:~/catkin_ws$
pancho-Inspiron-5547:~/catkin_ws$
  
```

Figura 3.4: Envío y recepción de datos



Figura 3.5: Entradas y salidas

- Columna 6: Velocidad del eje x en metros/segundos.

3.3. Planificación de experimentos de vuelo para la adquisición de datos

Se define las variables de interés, y qué valores deben ser manipulados para excitar al sistema durante el experimento de vuelo real, se toma en cuenta que pueden existir señales no deseadas que no deben considerarse como variables manipuladas ya que estas pueden afectar negativamente la salida del sistema, siendo consideradas como perturbaciones para el sistema.

El período de muestreo está directamente relacionado con las constantes de tiempo del sistema, y es de gran importancia para el desarrollo del experimento de identificación. Así, un período de muestreo muy pequeño puede llevar a la adquisición de datos redundantes, que no aportan información relevante del sistema, pero sí ocupan espacio en la memoria del dispositivo, mientras que un período de muestreo demasiado grande provoca grandes dificultades a la hora de identificar el sistema debido a la posible pérdida de información en el momento del experimento. Por medio del software Matlab se consideró un tiempo de muestreo igual a 60 ms con la finalidad que este tiempo no nos permita la pérdida de datos del sistema al igual que la toma de datos redundantes.

Cuanta más información se tenga sobre el sistema, más exacto será el modelo matemático para el cuadricóptero AR.Drone 2.0. Se realizó los experimentos de vuelo en el plano cartesiano $x - y$, la Figura 3.6 ilustra como referencia del eje x positivo la dirección de la cámara de video del cuadricóptero, y de la misma manera el eje y positivo hacia la derecha de la cámara de video del mismo, la altura del cuadricóptero a 1 metro sobre el piso ya que en este eje siempre permanecerá a la misma altura convirtiéndose en una constante del sistema.

3.4. Muestras obtenidas

La estimación del modelo matemático del movimiento del AR.Drone 2.0, relaciona las entradas de los ángulos con las velocidades y posiciones del mismo, se basa en las respuestas de

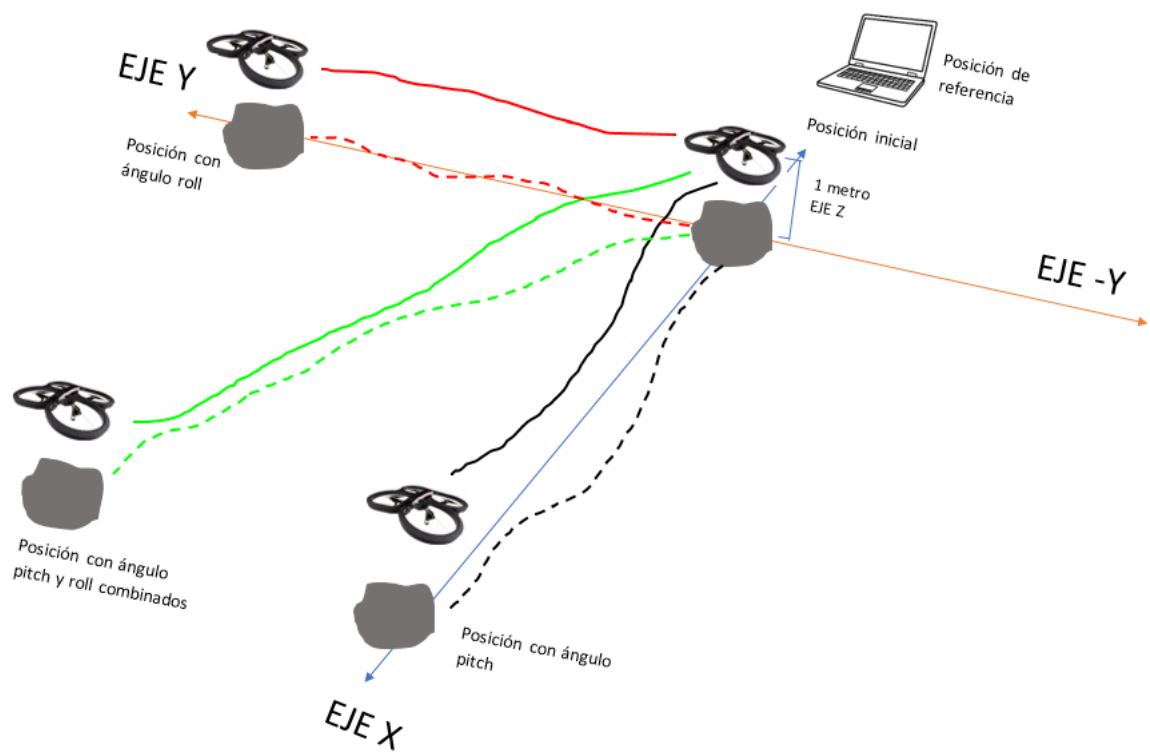


Figura 3.6: Plano de vuelo

vuelo real obtenidas mediante experimentos y proporcionadas por el metasisistema ROS junto con su librería Ardrone_Autonomy, el proceso de identificación del cuadricóptero se lleva a cabo mediante el software Matlab. Para ello, se escoge como variables manipuladas a los ángulos pitch y roll y como variables controladas las velocidades (v_x, v_y), mostradas en la Figura 3.1.

Para la identificación del cuadricóptero se utilizó 12001 muestras de datos, los cuales fueron separados para poder realizar la identificación y de igual manera la validación del sistema, los datos utilizados son mostrados en la la Figura 3.7, siendo los datos de color rojo las variables manipuladas y de color verde y azul las variables controladas, correspondientes a las velocidades en el eje y y en el eje x respectivamnte.

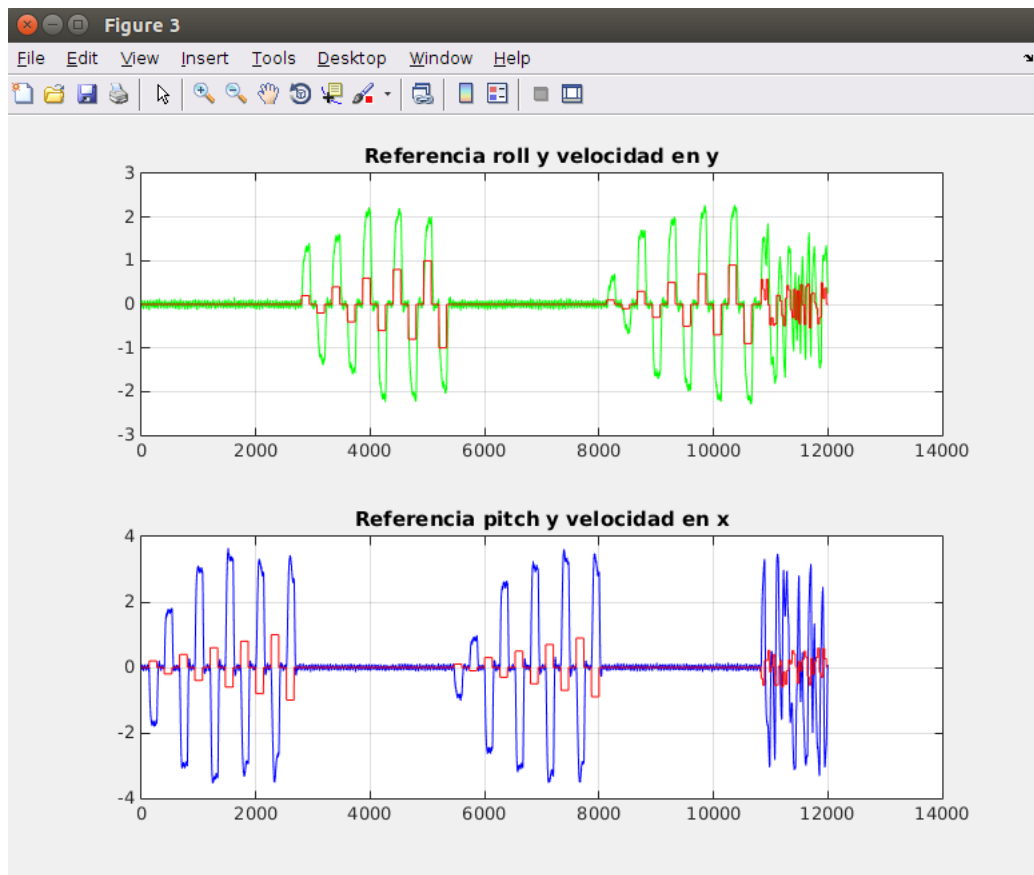


Figura 3.7: Datos del sistema

3.5. Estrategia de identificación

Los cuadricópteros son sistemas MIMO² o multivariantes, y no lineales en los que una entrada afecta a varias salidas y a su vez una salida es afectada por varias entradas. Este tipo de sistemas son más complicados al momento de identificar, por esta razón la estrategia de identificación consiste en separar al sistema en dos subsistemas SISO³ es decir de simple entrada y simple salida para cada uno de los ejes. Los modelos para cada eje de movimiento se definen con las siguientes relaciones, ángulo pitch con la velocidad v_x y el ángulo roll con la velocidad v_y .

La identificación del sistema se la analiza como caja negra y la adquisición de datos se la realizó mediante la librería ARDrone_autonomy y el lenguaje python que permite guardar toda la información del cuadricóptero. Para su posterior desarrollo, se utiliza la herramienta de identificación que Matlab proporciona denominada *ident*, considerando los siguientes criterios:

- El sistema se lo identifica como un proceso lineal en tiempo continuo representado por sus polos y sus ceros.
- El grado de la planta obedece al comportamiento del sistema alterado por los valores de entrada.

seguido por la implementación de un modelo en tiempo discreto ARX, ya que este tipo de modelo es el más eficiente y el que mejor se ajusta cuando el orden del sistema es demasiado alto.

3.6. Modelo y validación para el eje y

Las entradas y salidas del sistema se muestran en la Figura 3.8, en la cual se aprecia las gráficas de la toma de datos para la identificación, como para validar el modelo en el eje y.

Para la identificación del sistema en este eje se consideró 2650 muestras de datos y para su validación 2678 muestras. Para la generación del modelo ARX se hizo uso del Toolbox Ident que el software Matlab proporciona, que cuenta con la herramienta *polynomials models* que es usada como punto de partida para la generación del modelo ARX.

La generación del modelo se lo realiza mediante el botón *order selection*, que genera una interfaz gráfica de selección de la estructura de los modelos, la cual cuenta con tres criterios de selección MDL(best fit minimizes Rissanen) representado por el color verde, AIC(best fit minimizes Akaike) representado por el color azul y BEST FIT(best fit mínimos cuadrados) representados por el color rojo [21]. En la Figura 3.9 se aprecia la estructura generada bajo estos criterios.

²Sistema con varias entradas y salidas por sus siglas en inglés multiple inputs and multiple outputs

³Sistema con una entrada y una salida por sus siglas en inglés single inputs and single outputs

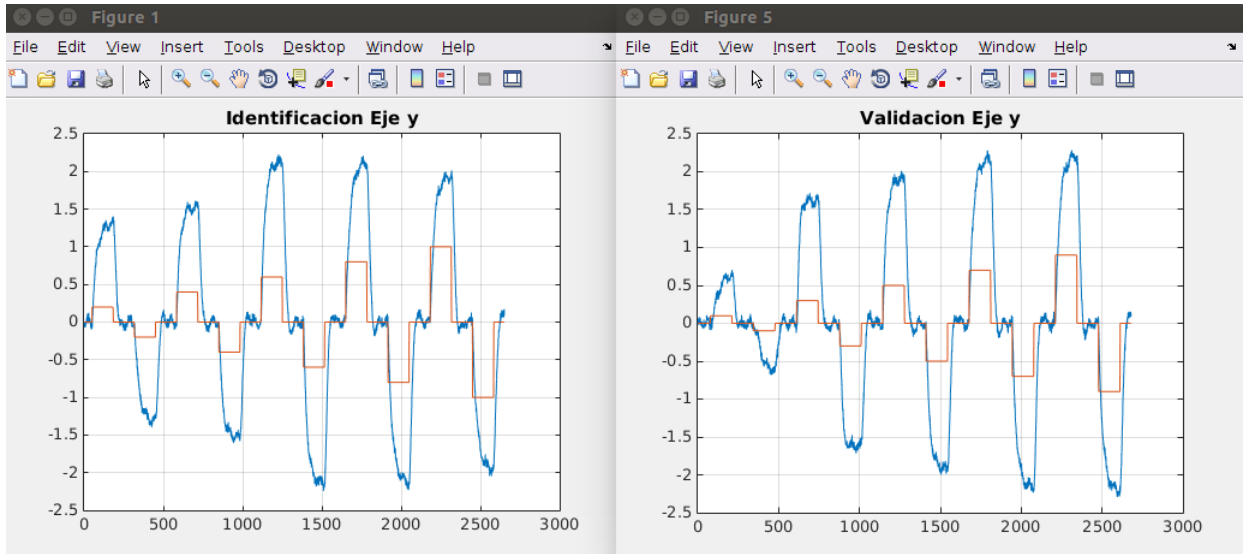


Figura 3.8: Datos para indentificación y validación del sistema

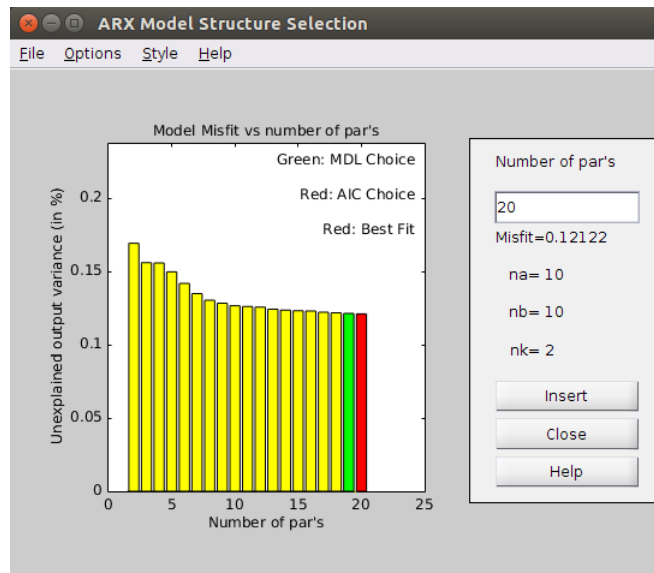


Figura 3.9: Estructura de selección modelo ARX

Matlab permite generar modelos ARX usando sus propias librerías, se debe simular el modelo muchas veces antes de encontrar un modelo que se ajuste de mejor manera a la realidad del sistema por lo cual se tomó modelos ARX de distintos órdenes como muestra la Figura 3.10.

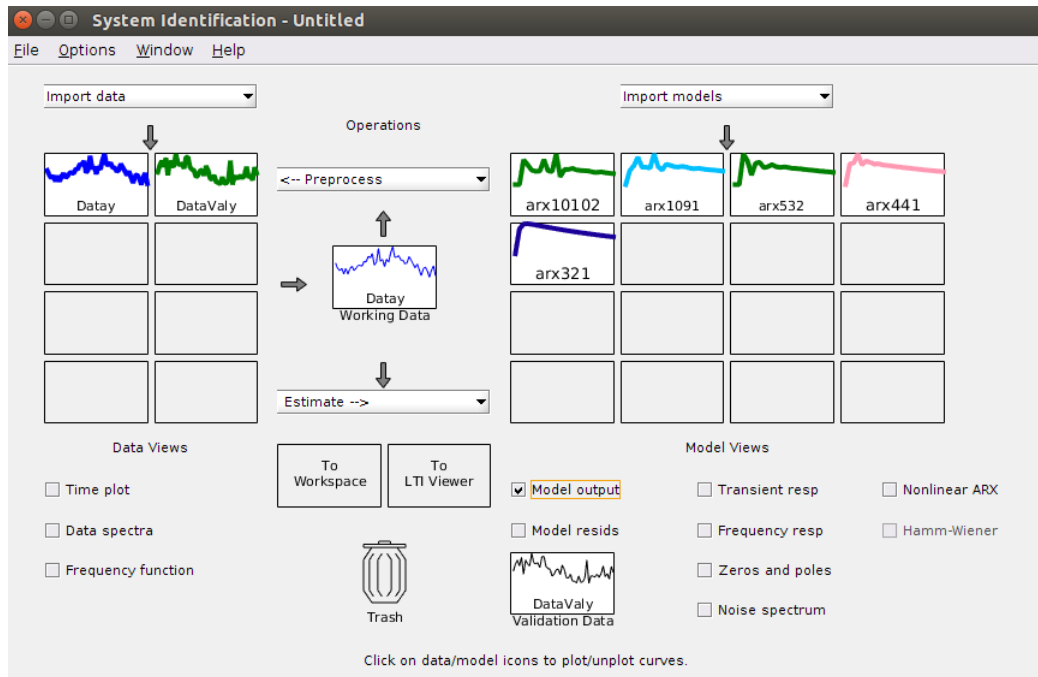


Figura 3.10: Modelos posibles para el eje y

En consecuencia en la Figura 3.11 se aprecia las gráficas pertenecientes al análisis residual y el ajuste a la correlación, las cuales indican que las líneas deben estar lo más cercanas a cero para que el modelo tenga mayor validez, de la misma forma la Figura 3.12 muestra la gráfica con el ajuste que mejor se adaptó al sistema real. El cuadro 3.1 contiene todos los parámetros que deben ser considerados antes de la selección del mejor modelo.

Cuadro 3.1: Parámetros de los modelos en el eje y

Modelo	na	nb	nk	Best Fit	Loss function (MSE)	Final Predictor error (FPE)
ARX10102	10	10	2	94.67	0.001484	0.001503
ARX1091	10	9	1	94.62	0.001485	0.001506
ARX532	5	3	2	93.4	0.001746	0.001756
ARX441	4	4	1	92.54	0.001878	0.00189
ARX321	3	2	1	91.93	0.001964	0.001972

El modelo elegido es ARX10102, además se observó que tiene valores bajos para el error de predicción final (FPE=0.001503) y la función de pérdida (MSE=0.001784). Este modelo es el que mejor se ajusta al sistema con un 94.67% de fidelidad, pero es un modelo con un polinomio

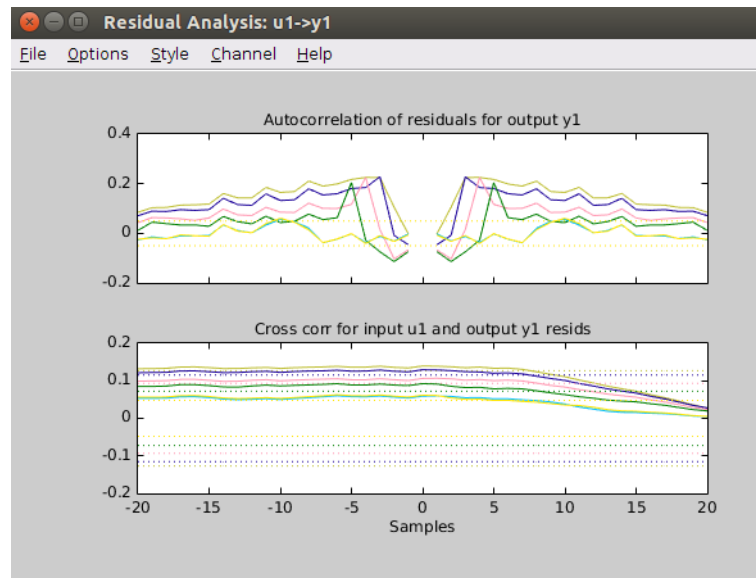


Figura 3.11: Análisis residual

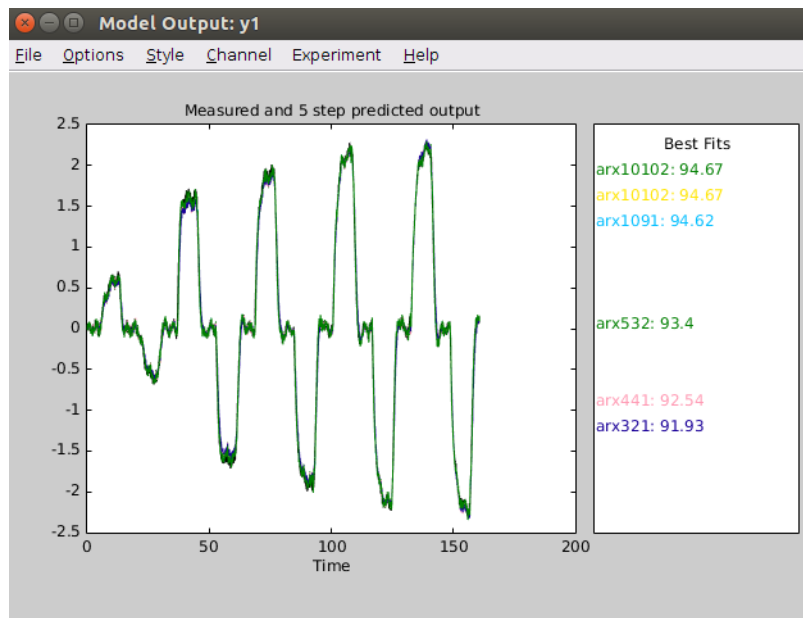


Figura 3.12: Mejores ajustes

con grado diez que es tan difícil de implementar en la vida real. Por esta razón el modelo que mejor se ajusta es el ARX321 porque su ajuste al sistema es aceptable en un 91.93 %, además que el FPE varía en 0.000469 y el MSE varía 0.00048 del modelo que mejor se ajusta al sistema, cabe recalcar que el FPE permite conocer que tanto se acerca del modelo real y el MSE permite conocer que tanto se aleja el modelo del sistema real, validando de esta manera el modelo seleccionado junto con la teoría básica de control que menciona que el número de polos de una función de transferencia deben ser mayor o igual al número de ceros de la ecuación.

El modelo ARX321 en tiempo discreto se muestra en la ecuación 3.1

$$A(z)y(t) = B(z)u(t) + e(t) \quad (3.1)$$

donde:

$$A(z) = 1 - 0,7315z^{-1} - 0,3332z^{-2} + 0,09628z^{-3}$$

$$B(z) = 0,06201z^{-1} + 0,02895z^{-2}$$

Por lo tanto la función de transferencia en tiempo discreto, es la planta identificada de velocidad en el eje y con respecto al ángulo roll se muestra en la ecuación 3.2:

$$P_y = \frac{Vy(z)}{\phi(z)} = \frac{0,06201z^{-1} + 0,02895z^{-2}}{1 - 0,7315z^{-1} - 0,3332z^{-2} + 0,09628z^{-3}} \quad (3.2)$$

donde $Vy(z)$ es la velocidad en el eje y y $\phi(z)$ es el ángulo roll.

3.7. Modelo y validación para el eje x

En la Figura 3.7 se muestra la trama de datos de entradas y salidas utilizadas para identificación, como para validación el modelo en el eje x. Para la identificación del sistema en este eje se consideró 2580 muestras de datos y para su validación 2600 muestras como se observa en la Figura 3.13.

Para la generación del modelo ARX se usó de igual manera el Toolbox Ident de Matlab, observando los criterios que esta herramienta ofrece explicados anteriormente, la Figura 3.14 muestra los mejores modelos que se adaptan al sistema.

De manera que en la Figura 3.15 se aprecia los distintos modelos ARX que mejor se adaptan al sistema real, el Cuadro 3.2 muestra todos los parámetros tomados en cuenta antes de la selección del mejor modelo. En la Figura 3.17 se ilustra las gráficas del análisis residual y el ajuste a la correlación, las cuales indican que las líneas deben estar lo más cercanas a cero para que el modelo tenga mayor validez, al igual que la Figura 3.16 muestra la gráfica con el ajuste que mejor se adaptó al sistema real.

El modelo elegido es ARX8103, además se observó que tiene valores más bajos para el error de predicción final (FPE=0.001983) y la función de pérdida (MSE=0.001957). Este modelo es el que mejor se ajusta al sistema con un 95.55 % de fidelidad, pero es un modelo con un polinomio con grado diez y es tan difícil de implementar en la vida real. Por esta razón el modelo

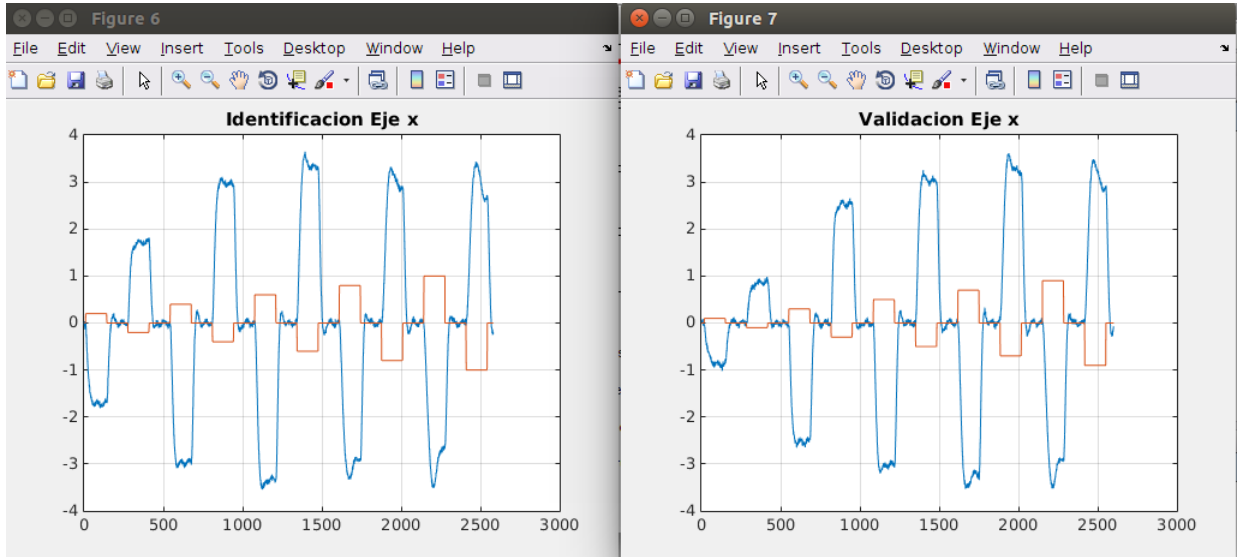


Figura 3.13: Datos de identificación y validación del eje x

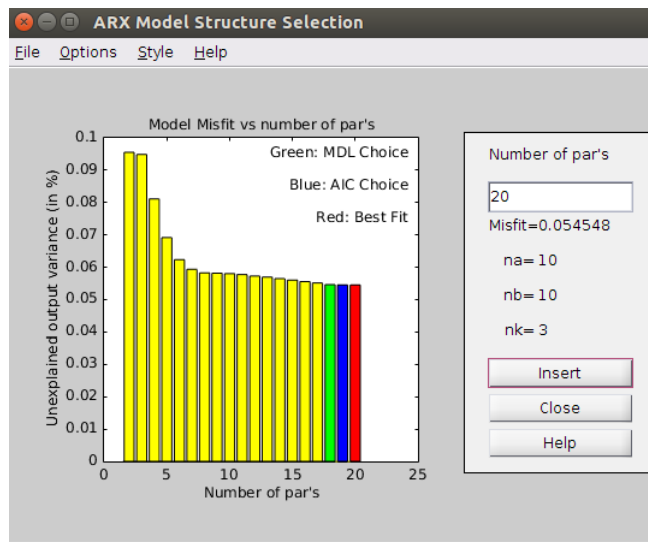


Figura 3.14: Estructura de selección modelo ARX

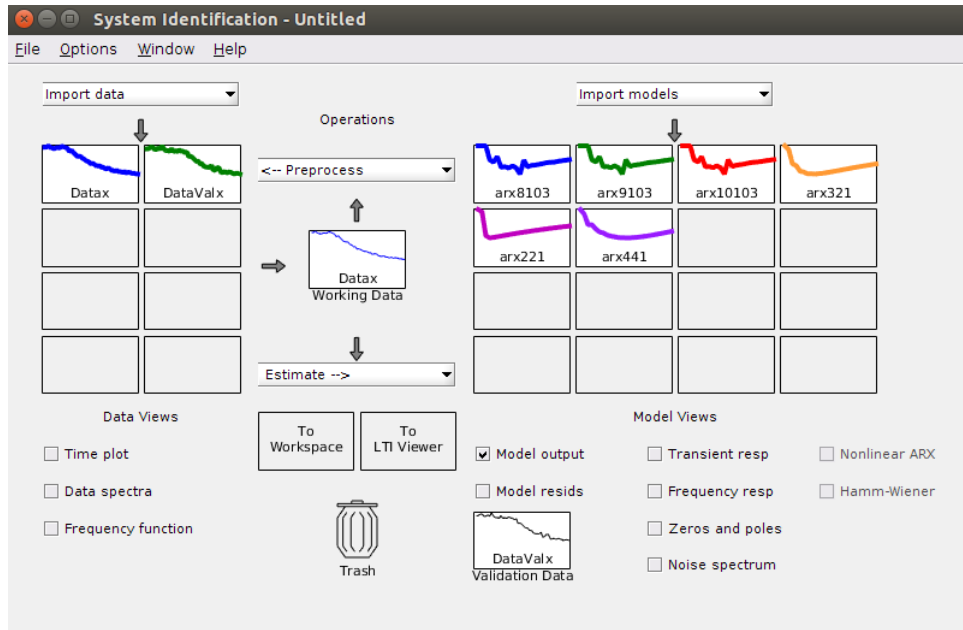


Figura 3.15: Posibles modelos para el eje x

Cuadro 3.2: Parámetros de los modelos en el eje x

Modelo	na	nb	nk	Best Fit	Loss function (MSE)	Final Predictor error (FPE)
ARX8103	8	10	3	95.55	0.001957	0.001983
ARX91031	9	10	3	95.54	0.001956	0.001983
ARX10103	10	10	3	95.55	0.001956	0.001985
ARX321	3	2	1	92.62	0.002949	0.00296
ARX221	2	2	1	90.82	0.003565	0.003577
ARX441	4	4	1	94.11	0.002518	0.002534

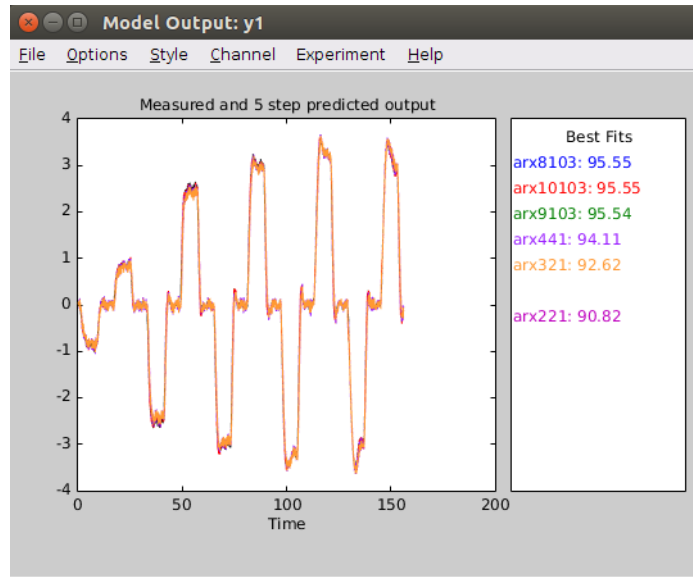


Figura 3.16: Mejores ajustes de modelos

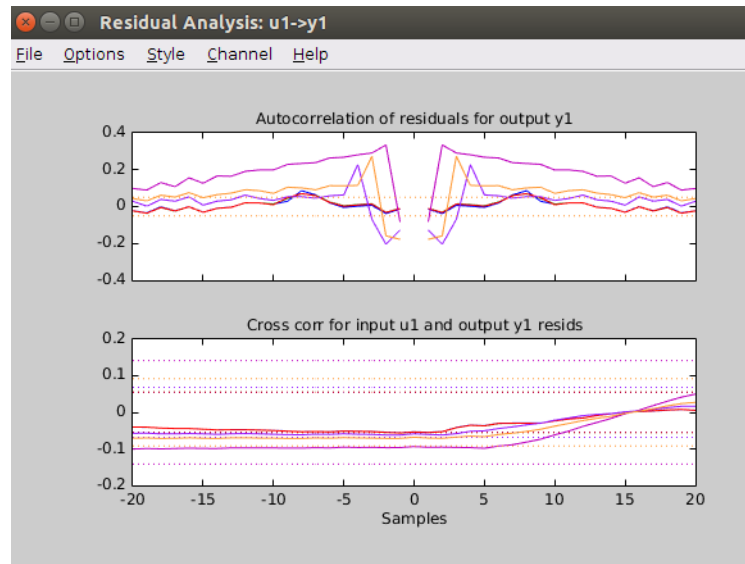


Figura 3.17: Análisis residual

que mejor se ajusta es el ARX321 porque su ajuste es aceptable en un 92.62%, además que el FPE varía en 0.000977 y el MSE varía 0.000992 del modelo que mejor se ajusta al sistema, validando de esta manera el modelo seleccionado junto con la teoría básica de control que nos dice que el número de polos de una función de transferencia deben ser mayor o igual al número de ceros de la ecuación.

El modelo ARX321 en tiempo discreto:

$$A(z)y(t) = B(z)u(t) + e(t) \quad (3.3)$$

donde:

$$A(z) = 1 - 1,041z^{-1} - 0,3472z^{-2} + 0,4044z^{-3}$$

$$B(z) = -0,01415z^{-1} - 0,05767z^{-2}$$

Por lo tanto la función de transferencia en tiempo discreto, es la planta identificada de velocidad en el eje x con respecto al ángulo pitch se muestra en la ecuación 3.4

$$P_x = \frac{Vx(z)}{\theta(z)} = \frac{-0,01415z^{-1} - 0,05767z^{-2}}{1 - 1,091z^{-1} - 0,3472z^{-2} + 0,4044z^{-3}}, \quad (3.4)$$

donde $Vx(z)$ es la velocidad en el eje x y $\theta(z)$ es el ángulo pitch.

Capítulo 4

Conclusiones y recomendaciones

Este capítulo muestra las conclusiones del presente proyecto y manifiesta algunas posibles sugerencias para su aplicación en trabajos futuros.

4.1. Conclusiones

1. La identificación de sistemas tiene por objetivo obtener el modelo matemático de sistemas dinámicos, partiendo de mediciones que usan una serie de etapas o pautas que tienen como fin que el modelo resultante sea representativo y se adapte de mejor manera al sistema identificado.
Estas etapas o pausas deben realizarse en un orden específico ya que su variación puede resultar en comportamientos erróneos del sistema, estas etapas van desde el diseño de la experimentación, adquisición y tratamiento de datos, elección de la estructura del modelo y la validación del sistema.
2. El cuadricóptero AR.Drone 2.0 se eligió para este proyecto basándose en diversos paquetes ya creados para su desarrollo en el entorno ROS, dándonos a entender que este cuadricóptero es idóneo para este tipo de trabajos, especialmente por su paquete *navdata topic* el cual permite el envío de información del cuadricóptero hacia un computador, esta información que es recibida se publica a través de *navdata* en un script de python lo hace posible la lectura de los sensores y su utilización.
3. La adquisición de datos del cuadricóptero, al ser en un entorno cerrado se puede visualizar de mejor manera la variación de los mismos, ya que en este medio no existen perturbaciones como el viento que hacen que los datos varíen de una forma más abrupta pero también existen datos redundantes en el sistema los cuales hay que analizarlos antes de iniciar con el proceso de identificación.
4. El sensor ultrasónico del cuadricóptero presenta mediciones erróneas al despegar en una superficie inclinada dificultando la estabilidad del mismo, se concluye que al despegar de

una superficie plana la elevación no presenta ningún problema y la adquisición de datos se la realiza de una forma más confiable.

5. El rango seleccionado para el movimiento del cuadricóptero es de $[-1;1]$ en los ángulos Pitch y Roll, por lo que se concluye que en este rango de valores el AR.Drone 2.0 presenta un vuelo normal con una velocidad apreciable para cada uno de sus ejes de movimiento, permitiendo capturar los datos necesarios para su identificación.
6. Utilizando la herramienta *ident* que el software Matlab nos proporciona nos resulta sencillo obtener distintos modelos, los resultados obtenidos nos permiten concluir que dichos modelos se aproximan al comportamiento real del cuadricóptero en un 94.67% en el eje y y un 92.62% en el eje x , y valores mínimos de FPE y MSE aseguran esta aproximación al sistema real.

4.2. Recomendaciones

1. Una de las desventajas que el AR.Drone 2.0 presenta es que el tiempo de vuelo está limitado a 12 minutos y la carga de la batería es de 90 minutos por lo que se recomienda tener como mínimo una batería extra con el fin de que el trabajo se realice de forma continua.
2. Las hélices del cuadricóptero disminuyen su durabilidad debido a las constantes pruebas y golpes que sufre al momento de la experimentación de la adquisición de datos, por lo cual para su posterior modelado se recomienda el cambio de hélices y calibrarlas para su óptimo desempeño.
3. Para la adquisición de datos del cuadricóptero, se recomienda tener en cuenta los manuales de la librería ARDrone_Autonomy y su relación con ROS y python para facilitar la comunicación entre el AR.Drone 2.0 y el computador.
4. Se recomienda para salvaguardar las partes mecánicas y electrónicas, especialmente los motores del cuadricóptero, que los futuros vuelos para el modelo del sistema, se realicen en una superficie considerablemente grande y libre de obstáculos.

4.3. Trabajo futuro

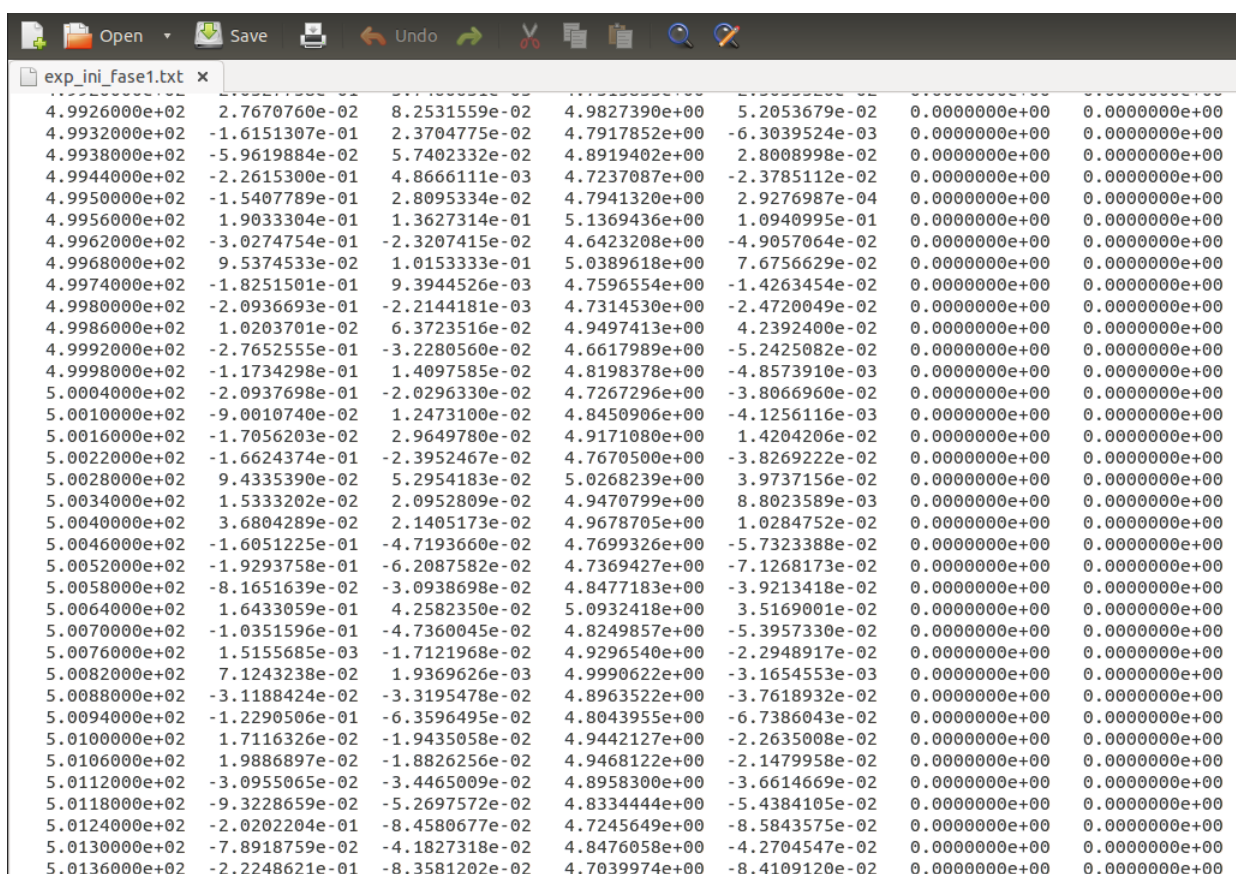
En el presente proyecto se ha abordado la teoría de identificación de sistemas, siendo este el arte de obtener modelos matemáticos que predicen el comportamiento de una planta. El modelo matemático seleccionado mediante diferentes criterios, sirve como base para diferentes tipos de aplicaciones.

Se debe realizar la identificación del sistema en el eje z , además del modelado de la función de transferencia, en el cual ya influye la dinámica del sistema, como por ejemplo los momentos de

inercia del cuadricóptero, su peso, etc, pero se toma en cuenta ciertos parámetros de trabajo. Los datos experimentales que arroja el sistema dependen del lugar geográfico al ser un modelo de caja negra, por lo que la sugerencia principal para el posterior modelado del sistema es que debe realizarse en un mismo medio y bajo las mismas condiciones para que no existan incoherencias al momento de modelar el sistema y así poder aplicar diferentes tipos de control en la planta real.

Apéndice A

Datos del cuadricóptero



Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
4.9926000e+02	2.7670760e-02	8.2531559e-02	4.9827390e+00	5.2053679e-02	0.0000000e+00	0.0000000e+00
4.9932000e+02	-1.6151307e-01	2.3704775e-02	4.7917852e+00	-6.3039524e-03	0.0000000e+00	0.0000000e+00
4.9938000e+02	-5.9619884e-02	5.7402332e-02	4.8919402e+00	2.8008998e-02	0.0000000e+00	0.0000000e+00
4.9944000e+02	-2.2615300e-01	4.8666111e-03	4.7237087e+00	-2.3785112e-02	0.0000000e+00	0.0000000e+00
4.9950000e+02	-1.5407789e-01	2.8095334e-02	4.7941320e+00	2.9276987e-04	0.0000000e+00	0.0000000e+00
4.9956000e+02	1.9033304e-01	1.3627314e-01	5.1369436e+00	1.0940995e-01	0.0000000e+00	0.0000000e+00
4.9962000e+02	-3.0274754e-01	-2.3207415e-02	4.6423208e+00	-4.9057064e-02	0.0000000e+00	0.0000000e+00
4.9968000e+02	9.5374533e-02	1.0153333e-01	5.0389618e+00	7.6756629e-02	0.0000000e+00	0.0000000e+00
4.9974000e+02	-1.8251501e-01	9.3944526e-03	4.7596554e+00	-1.4263454e-02	0.0000000e+00	0.0000000e+00
4.9980000e+02	-2.0936693e-01	-2.2144181e-03	4.7314530e+00	-2.4720049e-02	0.0000000e+00	0.0000000e+00
4.9986000e+02	1.0203701e-02	6.3723516e-02	4.9497413e+00	4.2392400e-02	0.0000000e+00	0.0000000e+00
4.9992000e+02	-2.7652555e-01	-3.2280560e-02	4.6617989e+00	-5.2425082e-02	0.0000000e+00	0.0000000e+00
4.9998000e+02	-1.1734298e-01	1.4097585e-02	4.8198378e+00	-4.8573910e-03	0.0000000e+00	0.0000000e+00
5.0004000e+02	-2.0937698e-01	-2.0296330e-02	4.7267296e+00	-3.8066960e-02	0.0000000e+00	0.0000000e+00
5.0010000e+02	-9.0010740e-02	1.2473100e-02	4.8450906e+00	-4.1256116e-03	0.0000000e+00	0.0000000e+00
5.0016000e+02	-1.7056203e-02	2.9649780e-02	4.9171080e+00	1.4204206e-02	0.0000000e+00	0.0000000e+00
5.0022000e+02	-1.6624374e-01	-2.3952467e-02	4.7670500e+00	-3.8269222e-02	0.0000000e+00	0.0000000e+00
5.0028000e+02	9.4335390e-02	5.2954183e-02	5.0268239e+00	3.9737156e-02	0.0000000e+00	0.0000000e+00
5.0034000e+02	1.5333202e-02	2.0952809e-02	4.9470799e+00	8.8023589e-03	0.0000000e+00	0.0000000e+00
5.0040000e+02	3.6804289e-02	2.1405173e-02	4.9678705e+00	1.0284752e-02	0.0000000e+00	0.0000000e+00
5.0046000e+02	-1.6051225e-01	-4.7193660e-02	4.7699326e+00	-5.7323388e-02	0.0000000e+00	0.0000000e+00
5.0052000e+02	-1.9293758e-01	-6.2087582e-02	4.7369427e+00	-7.1268173e-02	0.0000000e+00	0.0000000e+00
5.0058000e+02	-8.1651639e-02	-3.0938698e-02	4.8477183e+00	-3.9213418e-02	0.0000000e+00	0.0000000e+00
5.0064000e+02	1.6433059e-01	4.2582350e-02	5.0932418e+00	3.5169001e-02	0.0000000e+00	0.0000000e+00
5.0070000e+02	-1.0351596e-01	-4.7360045e-02	4.8249857e+00	-5.3957330e-02	0.0000000e+00	0.0000000e+00
5.0076000e+02	1.5155685e-03	-1.7121968e-02	4.9296540e+00	-2.2948917e-02	0.0000000e+00	0.0000000e+00
5.0082000e+02	7.1243238e-02	1.9369626e-03	4.9990622e+00	-3.1654553e-03	0.0000000e+00	0.0000000e+00
5.0088000e+02	-3.1188424e-02	-3.3195478e-02	4.8963522e+00	-3.7618932e-02	0.0000000e+00	0.0000000e+00
5.0094000e+02	-1.2290506e-01	-6.3596495e-02	4.8043955e+00	-6.7386043e-02	0.0000000e+00	0.0000000e+00
5.0100000e+02	1.7116326e-02	-1.9435058e-02	4.9442127e+00	-2.2635008e-02	0.0000000e+00	0.0000000e+00
5.0106000e+02	1.9886897e-02	-1.8826256e-02	4.9468122e+00	-2.1479958e-02	0.0000000e+00	0.0000000e+00
5.0112000e+02	-3.0955065e-02	-3.4465009e-02	4.8958300e+00	-3.6614669e-02	0.0000000e+00	0.0000000e+00
5.0118000e+02	-9.3228659e-02	-5.2697572e-02	4.8334444e+00	-5.4384105e-02	0.0000000e+00	0.0000000e+00
5.0124000e+02	-2.0202204e-01	-8.4580677e-02	4.7245649e+00	-8.5843575e-02	0.0000000e+00	0.0000000e+00
5.0130000e+02	-7.8918759e-02	-4.1827318e-02	4.8476058e+00	-4.2704547e-02	0.0000000e+00	0.0000000e+00
5.0136000e+02	-2.2248621e-01	-8.3581202e-02	4.7039974e+00	-8.4109120e-02	0.0000000e+00	0.0000000e+00

Apéndice B

Listado de códigos

Este apéndice incluye los programas implementados en la ejecución de la identificación del sistema.

B.1. Programas implementados

B.1.1. Carga de datos a Matlab

Listing B.1: Ingreso de archivo .txt a Matlab

```
load exp_ini_fase1.txt % carga el archivo
dibujaEnsayo(exp_ini_fase1)
t=exp_ini_fase1(:,1);
T = t(4)-t(3);
x=exp_ini_fase1(:,2);%posicion x
vx=exp_ini_fase1(:,3);%velocidad x
y=exp_ini_fase1(:,4);%posicion y
vy=exp_ini_fase1(:,5);%velocidad y
mp=exp_ini_fase1(:,6);%referencia pitch
mr=exp_ini_fase1(:,7);%referencia roll
```

B.1.2. Gráficas de datos en Matlab

Listing B.2: Gráfica de datos en Matlab

```
function dibujaEnsayo(exp)
% dibujaEnsayo(exp)
%
% 'exp' tiene 7 columnas.
% col 1: tiempo en segundos
```

```

% col 2: posicion x en metros
% col 3: velocidad x en metros/segundos
% col 4: posicion y en metros
% col 5: velocidad y en metros/segundos
% col 6: RefPitch entre -1 y 1
% col 7: RefRoll entre -1 y 1
%
figure
subplot(6,1,1)
plot(exp(:,1),exp(:,2))%grafica tiempo y posicion en x
title('Posicion eje x')
subplot(6,1,2)
plot(exp(:,1),exp(:,3))%grafica tiempo y velocidad en x
title('Velocidad en el eje x (vx)')
subplot(6,1,3)
plot(exp(:,1),exp(:,4))%grafica tiempo y posicion en y
title('Posicion eje y')
subplot(6,1,4)
plot(exp(:,1),exp(:,5))%grafica tiempo y velocidad en y
title('Velocidad eje y (vy)')
subplot(6,1,5)
plot(exp(:,1),exp(:,6))%grafica referencia pitch
title('Referencia Pitch')
subplot(6,1,6)
plot(exp(:,1),exp(:,7))%grafica referencia roll
title('Referencia Roll')
xlabel('sec')

figure
plot(exp(:,2),exp(:,4))%grafica  posicionen x & y
title('Trayectoria xy')

```

B.1.3. Identificación

Listing B.3: Proceso de identificación en Matlab

```

%%Drone Identification
clear all
close all
clc

%Reading inputs
load exp_ini_fase1.txt
dibujaEnsayo(exp_ini_fase1)
t=exp_ini_fase1(:,1);
T = t(10)-t(9); % Sampling time
x=exp_ini_fase1(:,2); %posicion en el eje x
vx=exp_ini_fase1(:,3);%velocidad en el eje x
y=exp_ini_fase1(:,4);%posicion en el eje y
vy=exp_ini_fase1(:,5);%velocidad en el eje y
mp=exp_ini_fase1(:,6);%referencia pitch
mr=exp_ini_fase1(:,7);%referencia roll

ident%identificacion

figure(3)%referencia roll y velocidad en y

```

```

subplot(2,1,1)
plot(vy,'g')
hold on
grid on
plot(mr,'r')
title('Referencia roll y velocidad en y')
subplot(2,1,2)%referencia pitch y velocidad en x
plot(vx,'b')
hold on
grid on
plot(mp,'r')
title('Referencia pitch y velocidad en x')

%%Datos eje y

Datos_identy=vy(2751:5401,1)
Datos_rolly=mr(2751:5401,1)
figure()
plot(Datos_identy)
hold on
plot(Datos_rolly)
title('Identificacion Eje y')
grid on
Datay=iddata(Datos_identy,Datos_rolly,T)%sample time
%%Datos validacion y

Datos_valy=vy(8057:10735,1)
Datos_rvaly=mr(8057:10735,1)
figure(5)
plot(Datos_valy)
hold on
plot(Datos_rvaly)
title('Validacion Eje y')
grid on
DataValy=iddata(Datos_valy,Datos_rvaly,T)%sample time

%%Datos eje x

Datos_identx=vx(125:2705,1)
Datos_Pitchx=mp(125:2705,1)
figure(6)
plot(Datos_identx)
hold on
plot(Datos_Pitchx)
title('Eje x')
title('Identificacion Eje x')
grid on
Datax=iddata(Datos_identx,Datos_Pitchx,T)%sample time

%%Datos validacion x

Datos_valx=vx(5450:8050,1)
Datos_pvalx=mp(5450:8050,1)
figure(7)
plot(Datos_valx)
hold on
plot(Datos_pvalx)
title('Validacion Eje x')
grid on
DataValx=iddata(Datos_valx,Datos_pvalx,T)%sample time

```

Bibliografía

- [1] L. Ljung., “System Identification. Theory for the user.”, 1999.
- [2] J. S. Franco and J. R. V. Flecha, “Introducción a la identificación de sistemas (recuperado de: <http://www.tecnicaindustrial.es/tifrontal/a-1408-introduccion-identificacion-si>), 2005, online; accessed 13 April 2017.
- [3] T. Bresciani, “Modelling, Identification and Control of a Quadrotor Helicopter”. *Department of Automatic Control*, Lund University, 2008.
- [4] Y. Sun, “MODELING, IDENTIFICATION AND CONTROL OF A QUAD-ROTOR DRONE USING LOW-RESOLUTION SENSING”. *University of Illinois at Urbana-Champaign*, 2012.
- [5] A. A. S. Vásquez, “Desarrollo de un sistema de SLAM para el robot AR.Drone en entorno ROS”, *UNIVERSIDAD DE ALCALÁ*, 2015.
- [6] J. C. Fernández, “DISEÑO DE UN SISTEMA DE CONTROL PARA UN CUADRICÓPTERO”. *España*, 2013.
- [7] G. Arencibia-Carballo, “Es posible el uso de drones en la investigación científica y el monitoreo ambiental?”, 2016.
- [8] C. M. Poveda, “Aplicaciones sobre ARDrone”. *España*, 2013.
- [9] I. C. A. Organization, “Unmanned Aircraft Systems (UAS)”, 2011.
- [10] C. V. G. FRANKLIN and O. P. K. COLEMAN, “Diseño e implementación de un sistema de control de posición para el cuadricóptero AR.DRONE 2.0 de Parrot”. *Sangolquí*, 2015.
- [11] E. L. Guillén, “Identificación de sistemas. Aplicación al modelado de un motor de continua”. *España*.
- [12] C. Kunusch, “Identificación de sistemas dinámicos”. *Argentina*, 2003.
- [13] L. Ljung, “Perpectives on System Identification”. *Suecia*.
- [14] A. Bemporad, “System identification”. *Italia*, 2011.

- [15] L. M. V. FONSECA, “Desarrollo de algoritmos para el seguimiento de trayectorias de un cuadrotor utilizando técnicas modernas de control con álgebra lineal. *Quito*, 2015.
- [16] I. J. Hvizdo and prof. Ing. Peter Sinák CSc., “Control library for AR.Drone 2.0”. *Slovakia*, 2015.
- [17] Parrot(recuperado de: <https://www.parrot.com/es/drones/parrot-ardrone-20-elite-%C3%A9dition#parrotardrone-20-elite-edition>, online; accessed 13 April 2017.
- [18] S. Piskorski, N. Brulez, P. Eline, and F. DHaeyer, “AR.Drone Developer Guide”, 2012.
- [19] ROS, ROS.org(recuperado de: <http://wiki.ros.org/indigo/Installation/Ubuntu.>, online; accessed 13 April 2017.
- [20] Ardone_autonomy(recuperado de: <http://ardrone-autonomy.readthedocs.io/en/latest/>, online; accessed 13 April 2017.
- [21] L. Ljung, “System Identification Toolbox-User’s Guide”, 2017.